Méthodes d'apprentissage statistique pour le scoring

Marine Depecker

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de docteur

de l'Ecole Nationale Supérieure des Télécommunications

Spécialité : Signal et Images

Marine DEPECKER

Méthodes d'apprentissage statistique pour le scoring.

Soutenue le 10 Décembre 2010 devant le jury composé de

Alexandre Tsybakov	Président
Trevor Hastie	Rapporteur
Alain Rakotomamonjy	Rapporteur
Nicolas Vayatis	Examinateur
Stéphan Clémençon	Directeur de thèse
François Roueff	Co-directeur de thèse
Yves Tourbier	Encadrant Renault
Antione Saint-Marcoux	Encadrant Renault

Remerciements

Mes premiers remerciements vont à mes directeurs de thèse François Roueff, qui m'a accueillie au laboratoire et m'a mis le pied à l'étrier et Stéphan Clémençon, qui m'a encadrée et guidée tout au long de ces trois ans. François, nous n'avons eu que peu d'occasions de travailler ensemble, mais ta présence et ta gentillesse ont été un soutien quotidien. Stéphan, je n'aurais pas pu accomplir tout ce travail si tu n'avais pas été là pour me guider et me (re-)motiver si souvent. Tu m'as fait découvrir le monde de la recherche, initiée aux joies du Machine Learning et convaincue de poursuivre dans cette voie ! Je te remercie de m'avoir fait confiance, de m'avoir accompagnée ainsi pendant trois ans et pour toutes ces discussions et séances de travail, autour d'un café, à Paris ou à Vancouver. J'ai beaucoup appris à tes côtés et j'espère que nous pourrons continuer à travailler ensemble à l'avenir.

Je remercie du fond du coeur Antoine Saint-Marcoux, qui m'a encadrée chez Renault, pour mon stage de fin d'études d'abord, pour ma thèse ensuite. Merci pour tout ce que tu m'as apporté, pour ta disponibilité, ton soutien, ta gentillesse et ton amitié. Merci pour tous ces fous rires, pour toutes ces séances de travail qui se sont souvent terminées par des fous rires, pour tous ces cafés et toutes les discussions qui les ont accompagnés. Je n'aurais pas pu rêver meilleur encadrant que toi, ce fut un plaisir et un honneur d'être ton "padawan" et j'espère que de nombreux autres doctorants pourront profiter de tes nombreux conseils et de ta joie de vivre ! Je remercie aussi chaleureusement l'ensemble du groupe Calcul et Optimisation (DREAM-team...) de Renault. J'ai passé de très bons moments à vos côtés et beaucoup appris à votre contact. Cela a été un réel plaisir de travailler avec vous, de partager vos boquettes...et les cris du voisinage ! Je dois vous avouer que même nos réunions hebdomadaires me manquent...

Je remercie l'ensemble des membres du jury, pour avoir accepté de braver le froid et la neige pour participer à ma soutenance et pour l'intérêt porté à mes travaux de recherche. Je remercie tout particulièrement Alexandre Tsybakov, professeur à l'Université Paris 6, d'avoir accepté de présider ce jury et je tiens à exprimer ma reconnaissance à Trevor Hastie, professeur à l'Université de Stanford, et Alain Rakotomamonjy, professeur à l'Université, pour m'avoir fait l'honneur d'être rapporteurs de ces travaux.

Comme l'a dit un célèbre scribe il y a bien longtemps, "si je devais résumer ma vie, aujourd'hui, avec vous, je dirais que c'est d'abord des rencontres" ([Chabat *et al.* 2002]). Des rencontres, j'en ai fait beaucoup pendant ces trois ans, toutes très enrichissantes. Je remercie notamment Nicolas Vayatis pour sa gentillesse, son soutien, sa patience et sa grande pédagogie. Merci pour ce temps que tu m'as consacré, malgré un emploi du temps de ministre ! C'est une chance et un plaisir de pouvoir travailler avec toi. Un grand merci aussi à toute l'équipe STA du département TSI de Télécom ParisTech pour leur accueil chaleureux et la bonne humeur qui règne dans les couloirs. Une mention spéciale à l'ensemble des doctorants de Télécom, pour cette belle ambiance et tous les bons moments passés ensemble, et à l'équipe Panna Cotta, pour votre accueil toujours chaleureux... et pour m'avoir fait un découvrir ce sympathique petit restaurant italien !

Mes derniers remerciements vont à ma famille, ma belle-famille et à mes proches, qui m'ont soutenue et supportée pendant ces trois ans...et qui le font depuis bien plus longtemps !
Agnès, colocataire téméraire, tu as vu passer le blues de la première année, merci pour ton amitié, ton soutien...pour tout ce riz et surtout pour tous ces bons moments ! Pascal,

merci de m'avoir accueillie et fait découvrir Paris, merci aussi pour toutes ces conversations où nous avons (tenté) de refaire le monde...heureusement, nous n'en avons pas encore terminé ! Antoine, JF, Sarah, Nico, plus que des collègues, vous êtes devenus des amis et votre soutien a été précieux !

Mes plus grands mercis sont pour mes parents, qui supportent mes doutes et partagent mes joies depuis toujours. Merci pour votre amour et votre confiance sans faille. Je ne serais pas là où j'en suis aujourd'hui sans vous et sans votre soutien inconditionnel, merci du fond du coeur, pour tout...y compris pour votre patiente (double) relecture (croisée) de ce manuscrit !

Et puis parce qu'il n'y en avait pas assez dans cette histoire, Nicolas, je te remercie du fond du coeur, pour avoir supporté les hauts et les bas, pour avoir accepté de partager ta vie avec un zombie pendant 4 mois, pour m'avoir nourrie pendant 4 mois... et surtout pour être toujours là, après tout ça ! Merci d'avoir, toi aussi, relu scrupuleusement l'intégralité de ce manuscrit. Merci enfin pour tout ce bonheur que tu m'apportes quotidiennement. La vie est plus belle depuis que tu es là !

Enfin, je n'oublie pas Titi, qui a bercé ces longues heures de rédaction de son doux ronronnement et qui est venue inlassablement me rappeler par un doux miaulement qu'il y a un temps pour tout dans la vie, un temps pour travailler et un temps pour faire une pause et aller remplir la gamelle de croquettes. D'ailleurs, il est temps...

Je dédie ce travail à mes parents
et à la mémoire de ma tante Edwige

Résumé

Ces travaux de thèse portent sur l'étude théorique et le développement algorithmique d'une méthode non-paramétrique pour l'apprentissage supervisé de règles d'ordonnancement à partir de données étiquetées de façon binaire. Cette méthode de scoring, appelée TreeRank, dont les fondements théoriques ont été introduits par Clémençon et Vayatis (2007), généralise la notion d'arbre de décision au problème de l'ordonnancement. En scindant de manière récursive et adaptative l'espace $\mathcal{X}$ des observations de sorte à maximiser le critère ASC (pour Aire Sous la Courbe COR), l'algorithme TreeRank produit des règles de score constantes par morceaux, représentées graphiquement par des arbres binaires et orientés, dits *arbres d'ordonnancement*.

Les travaux de recherche présentés dans ce manuscrit s'organisent autour de trois axes. Dans un premier temps, nous introduisons une procédure de partitionnement, appelée LeafRank, permettant de scinder l'espace $\mathcal{X}$, à chaque itération de l'algorithme TreeRank, selon des règles adaptatives et complexes. La possibilité de choisir la règle de partitionnement selon le problème considéré confère une flexibilité importante à cette méthode d'apprentissage. L'étape d'optimisation de l'algorithme TreeRank pouvant être formulée comme un problème de classification binaire pondéré, nous proposons deux règles de partitionnement distinctes, basées respectivement sur la mise en oeuvre d'une version pondérée de l'algorithme de classification CART (Breiman *et al.*, 1984) ou de Machines à Vecteurs Supports (SVM, Vapnik, 1992).

Dans un deuxième temps, nous proposons de mettre en oeuvre une procédure de sélection automatique de modèle afin de lutter contre le phénomène de *sur-apprentissage*. Pour cela, nous nous inspirons de la procédure d'*élagage* introduite par Breiman *et al.* (1984) pour l'algorithme de classification CART. Adaptée aux *arbres d'ordonnancement*, celle-ci consiste à maximiser l'ASC empirique pénalisée linéairement par le cardinal de la partition induite sur l'espace $\mathcal{X}$ et à sélectionner le meilleur modèle par validation croisée. Nous proposons aussi une procédure alternative fondée sur la maximisation de l'ASC structurelle, par analogie avec l'approche de minimisation du risque structurel proposée par Vapnik (1982). Nous introduisons ainsi deux pénalités, pour les cas où la procédure LeafRank repose sur la définition a priori d'une partition dyadique et sur la mise en oeuvre d'une version pondérée de l'algorithme CART.

Enfin, nous abordons un troisième axe de recherche relatif à l'instabilité des *arbres d'ordonnancement*, inhérente à leur mode de construction. Afin de produire des règles de scores plus robustes, nous proposons d'adapter deux procédures d'agrégation de règles de prédiction ré-échantillonnées, initialement développées par Breiman pour les arbres de décision dans le cadre de la classification et de la régression : le *bagging* (1996) et les forêts aléatoires (*Random Forests*, 2001). Transposées au problème de l'ordonnancement, ces deux procédures nous amènent à considérer le problème de l'agrégation de *pré-ordres*, que nous étudions du point de vue de l'approche métrique.

Les deux derniers chapitres de ce manuscrit sont consacrés à l'étude empirique de la méthode d'ordonnancement développée pendant cette thèse. Nous proposons notamment une étude comparative entre différentes configurations de l'algorithme et quelques méthodes de l'état de l'art. Nous présentons ensuite l'application de la méthode TreeRank à une problématique industrielle : l'*objectivation des prestations* d'un véhicule automobile. Enfin, nous concluons ce manuscrit par un chapitre dédié au problème du test de l'*homogénéité* de deux populations, dans lequel nous proposons une heuristique en deux étapes fondée sur l'algorithme TreeRank et permettant de généraliser les tests de rangs au cas multi-dimensionnel.

Abstract

Bipartite ranking is a statistical issue consisting in sorting objects lying in a multidimensional feature space, randomly associated with binary labels, so that positive instances appear on top of the list with highest probability. This research work is centered on the brand-new ranking method proposed by Clémençon and Vayatis (2007), called TreeRank, specifically tailored for learning scoring functions by recursively maximizing the Area Under the ROC Curve (AUC). This tree-induction approach is based on a top-down recursive partitioning strategy leading to a ranking tree summarized by a rooted, binary, left-right oriented tree graph.

The contribution of this research work to the TreeRank learning method aims at improving its performances by addressing the three following issues : the choice of a suitable partitioning rule to grow an accurate ranking tree, the definition of a model selection procedure leading to the *best* ranking tree for prediction and the adaptation of re-sampling and aggregating procedures to output smoother ranking rules and fight against the instability inherent to the hierarchical structure of this tree-based learning procedure.
Thus, we first focus on the optimization step of the TreeRank algorithm, which consists in splitting into two siblings each cell of the partition induced by the ranking tree on the feature space $\mathcal{X}$, so that the AUC is maximized. To solve this optimization problem in a flexible manner, we introduce a partition-based procedure, called LeafRank, involving complex and adaptive splitting rules. Each iteration of the TreeRank procedure boiling down to a weighted binary classification problem with data-dependent cost, we propose two LeafRank heuristics respectively based on a weighted version of the classification CART algorithm (Breiman *et al.*, 1984) and on a weighted SVM classifier (Vapnik, 1992).
We then tackle the classical issue of model selection. We propose two penalization-based procedures to avoid both under-fitting and over-fitting phenomena and provide the best ranking rule for prediction. In particular, we adapt the bottom-up pruning scheme introduced by Breiman *et al.* (1984) for the tree-induction CART algorithm, which consists in optimizing a linearly penalized version of the AUC criterion. We also propose a second pruning procedure based on AUC structural optimization. In particular, we introduce non-linear penalties for two different partitioning rules.
The third research topic tackled by this thesis work deals with the question of instability : as any tree-induction method, TreeRank is very sensitive to small changes in the learning data which may provide very different ranking rules. Therefore, we propose to adapt two re-sampling and aggregating procedures introduced by Breiman in the classification and regression contexts to reduce the instability and increase the performances of such tree-induction procedures : bagging (1996) and random forests (2001). As the majority voting scheme used in the classification context to aggregate prediction rules has been shown to yield non-optimal aggregated ranking rules (Condorcet, 1785), we consider a metric approach based on the computation of a *median* prediction rule.

Last but not least, intensive experiments have been carried out on both toy examples and real data sets. In particular, we compare the performances of several versions of the TreeRank algorithm among them and with existing ranking methods. We also present the results output by the TreeRank algorithm on industrial *objectivization data*. Finally, the last chapter of this manuscript is dedicated to *homogeneity testing* in a multidimensional setting. More precisely, we introduce a two-stage testing procedure solving the *two-sample problem* based on the TreeRank algorithm and on one-dimensional rank tests.

Table des matières

Introduction

Les travaux de thèse présentés dans ce manuscrit portent sur le développement d'une méthode non-paramétrique pour l'apprentissage supervisé de règles d'ordonnancement à partir de données étiquetées de façon binaire.

On suppose que l'on observe une collection d'individus représentatifs d'une population, caractérisés par un ensemble d'*entrées* (*variables* ou *prédicteurs*) et par une *étiquette* (*label* ou *classe*) binaire, de la forme « bons-mauvais » par exemple. Lorsque l'on dispose de telles données, l'objectif le plus courant est d'apprendre un modèle permettant de prédire l'étiquette (inconnue) d'une nouvelle observation de la population. Ce problème d'apprentissage supervisé est bien connu dans la littérature : il s'agit de la classification binaire. Cependant, il existe aussi de nombreuses applications pour lesquelles l'enjeu principal n'est pas de prédire les labels binaires mais plutôt de définir une relation d'ordre sur la population. On parlera dans ce cas d'un problème de *scoring* ou d'*ordonnancement binaire*.

Cette problématique apparaît de manière récurrente dans des domaines très variés. On la rencontre par exemple en finance où, pour évaluer le risque de non-remboursement de crédits, les sociétés bancaires étudient un panel de clients, pour lesquels elles disposent d'informations concernant leur mode de vie (catégorie socio-professionnelle, situation maritale, revenus, etc.) et d'un label binaire indiquant si les crédits qu'ils ont contractés ont été remboursés. De même, la mise au point de règles de décision pour l'aide au diagnostic médical repose sur l'étude de patients pour lesquels on dispose d'une description des symptômes ressentis et d'une étiquette binaire indiquant s'ils sont « sains » ou « atteints ». Un autre exemple d'application réside dans la mise au point de moteurs de recherche capables de hiérarchiser une collection de documents selon leur pertinence par rapport à la requête formulée par un utilisateur.

Ainsi, le champ d'application des méthodes d'ordonnancement est très vaste et c'est encore un tout autre contexte qui a motivé le lancement de ces travaux de recherche par le constructeur automobile Renault, celui de l'*objectivation* des prestations relatives à l'*agrément de conduite* et plus particulièrement du *brio*, qui désigne la sensation d'accélération procurée par un véhicule, dont l'étude a été au centre de ces travaux de thèse. Ainsi, la méthode d'ordonnancement que nous allons présenter dans ce manuscrit a été mise en oeuvre afin d'identifier les caractéristiques physiques des véhicules expliquant le brio *ressenti* par le client, fourni sous la forme d'une cotation binaire de type « bons-mauvais », et de construire un indice permettant de quantifier le niveau de prestation atteint et de comparer les différents véhicules du marché.

Dans le cadre particulier de l'ordonnancement binaire, où l'étiquette observée est donc de nature binaire, l'objectif est d'arriver à ordonner une population de telle sorte que

les « meilleurs » individus se trouvent en tête de classement. Intuitivement, on perçoit qu'une « bonne » [1] solution consisterait à classer les individus selon leur probabilité d'être « bons », ou du moins « pertinents » selon l'étiquette considérée. Le problème de scoring revient alors à estimer l'ordre induit sur l'espace d'entrée $\mathcal{X} \subset \mathbb{R}^q$, $q \geq 1$, par la probabilité à posteriori définie par

$$\forall\, x \in \mathcal{X},\ \eta(x) = \mathbb{P}\{Y = +1 \mid X = x\},$$

où $X \in \mathcal{X}$, représente un ensemble de prédicteurs permettant de modéliser le label binaire $Y \in \{-1, +1\}$.

Plusieurs méthodes ont été proposées dans la littérature pour produire des règles d'ordonnancement à partir de données étiquetées de façon binaire. D'une manière générale, on peut distinguer deux grandes approches. D'une part, on trouve des méthodes statistiques classiques, qui visent à estimer la probabilité à posteriori η, puis utilisent cet estimateur pour ordonner les observations de l'espace d'entrée $\mathcal{X}$. Parmi ces méthodes on peut citer par exemple l'*analyse discriminante linéaire* (LDA pour *Linear Discriminant Analysis*, [Fisher 1936]) ou la *régression logistique* (voir par exemple [Hastie & Tibshirani 1990], [Friedman *et al.* 1998] ou [Hastie *et al.* 2001]), toutes deux basées sur la maximisation du rapport de vraisemblance d'un modèle de type *logit*. Un autre exemple est la *régression logistique à noyau* (KLR pour *Kernel Logistic Regression*, voir par exemple [Zhu & Hastie 2001], [Jaakkola & Haussler 1999] et [Keerthi *et al.* 2002]), qui procède à l'estimation *non paramétrique* de la probabilité à posteriori en résolvant un problème d'optimisation convexe de type SVM (pour *Support Vector Machines*, voir par exemple [Schölkopf & Smola 2002]).

D'autre part, une seconde approche est fondée sur l'optimisation d'un critère évaluant la *performance* ou *a contrario* le *risque* de la règle d'ordonnancement produite, au sens de l'approche ERM (pour *Empirical Risk Minimization*, [Vapnik & Chervonenkis 1974]). Parmi les nombreuses méthodes proposées, on peut citer les méthodes RankBoost (proposée dans [Freund *et al.* 2003]) et AdaRank (proposée dans [Xu & Li 2007]), fondées sur une procédure de type *boosting* ([Freund & Schapire 1999]), les méthodes RankSVM (proposée dans [Rakotomamonjy 2004] et [Joachims 2002b]) et RankRLS (proposée dans [Pahikkala *et al.* 2007]), basées sur la mise en oeuvre de SVM pour deux fonctions de coût différentes (respectivement la *hinge loss* et les *moindres carrés*) ou encore les méthodes RankNet (proposée dans [Burges *et al.* 2005]) et LambdaRank (proposée dans [Burges *et al.* 2006]), reposant sur l'utilisation de réseaux de neurones. Toutes ces méthodes présentent un point commun : elles reposent en réalité sur la comparaison des paires d'observations dans l'esprit de l'approche proposée dans [Cohen *et al.* 1999]. Ainsi, selon la méthode considérée, la performance de la règle d'ordonnancement produite est évaluée par des critères comme l'ASC, l'Aire Sous la Courbe COR (pour Caractéristique de la Courbe de Réception, [Egan 1975]), le τ de Kendall ([Kemeny 1959], [Kendall 1945]) ou encore la statistique de Wilcoxon ([Wilcoxon 1945]), s'exprimant en fonction du nombre de *paires discordantes*, *i.e.* du nombre de paires d'observations pour lesquelles l'ordonnancement induit par la règle de prédiction n'est pas cohérent avec les étiquettes binaires observées.

Ces deux approches présentent naturellement des avantages et des inconvénients. S'il semble naturel d'estimer directement la probabilité à posteriori, induisant l'ordonnancement que l'on souhaite retrouver, cette approche présente certaines limites. Tout d'abord,

1. au sens d'un critère que l'on précisera ultérieurement

dans les trois méthodes que nous venons de citer, la *représentation* de la fonction η est imposée par l'utilisation d'un modèle de type *logit*, qui pourrait ne pas correspondre aux données étudiées. Par ailleurs, les méthodes *paramétriques* comme la LDA ou la régression logistique sont confrontées à ce que l'on appelle le *fléau de la dimension*, l'estimation de la probabilité η devenant délicate dans un espace $\mathcal{X}$ de grande dimension. Sur ce point précis, les méthodes fondées sur la minimisation d'un *risque* d'ordonnancement permettront d'obtenir des règles de prédiction plus performantes. Cependant, les procédures de type ERM que nous avons citées ci-dessus présentent elles aussi un inconvénient majeur qui réside dans la nature *globale* des critères de performance optimisés. En effet, si ceux-ci permettent de trouver un « bon » ordonnancement de manière globale sur l'ensemble des observations, ils ne garantissent nullement d'obtenir le classement le plus pertinent des *meilleures* observations de l'échantillon. Or, il existe de nombreuses applications, comme la recherche de documents sur Internet par exemple, dans lesquelles on ne s'intéresse qu'au début du classement proposé. Aussi, d'autres critères ont été proposés dans la littérature afin d'évaluer la performance *locale* d'une règle d'ordonnancement, comme par exemple les critères d'ASC *partielles* ou *tronquées*, introduits respectivement dans [Dodd & Pepe 2003] et [Clémençon & Vayatis 2007], ou encore le critère *p-norm push* proposé dans [Rudin 2006].

Dans ce manuscrit, nous proposons une nouvelle méthode de scoring, appelée TreeRank, qui réunit les avantages des deux approches précédemment décrites. En effet, celle-ci produit une règle d'ordonnancement, qui s'exprime comme une transformée strictement croissante de la probabilité à posteriori, en optimisant le critère de performance ASC de manière *récursive*. La méthode TreeRank permet ainsi de retrouver l'ordre induit sur l'espace d'entrée $\mathcal{X}$ par la fonction η, sans pour autant être confrontée au problème de son estimation *directe* dans un espace de grande dimension.

Cette méthode de scoring repose sur le *partitionnement récursif* de l'espace d'entrée et sur la maximisation, à chaque itération, de l'ASC calculée *localement* sur un sous-ensemble de $\mathcal{X}$. En ce sens, cette approche permet d'étendre la notion d'*arbre de décision* (voir par exemple [Breiman *et al.* 1984], [Quinlan 1986] ou le Chapitre 9 de [Hastie *et al.* 2001]) à la problématique d'ordonnancement binaire. Nous nous plaçons ainsi dans la continuité des nombreuses contributions proposées dans la littérature, reposant d'une part sur l'utilisation d'arbres de classification, connus sous le nom de *Probabilistic Estimation Trees* (PET), pour estimer la probabilité à posteriori comme dans ([Provost & Domingos 2002]) et d'autre part, sur le choix d'un critère de partitionnement adapté au problème de l'ordonnancement, comme dans [Ferri *et al.* 2002], où l'auteur considère la construction d'arbres de décision fondés sur l'optimisation du critère ASC. La méthode de scoring proposée dans ce manuscrit produit donc, par construction, des règles d'ordonnancement constantes par morceaux et représentées graphiquement par un *arbre binaire orienté*, que nous appellerons *arbre d'ordonnancement*. Les *feuilles* de cet arbre représentent les cellules de la partition induite sur l'espace d'entrée $\mathcal{X}$ et sont ordonnées selon l'orientation « gauche-droite » de l'arbre, ce qui permet ainsi de *visualiser* la règle d'ordonnancement produite (cf Figure 1 ci-dessous).

Plan du manuscrit

Les travaux présentés dans ce manuscrit ont été réalisés dans le cadre d'une thèse CIFRE co-dirigée par le laboratoire LTCI de Télécom ParisTech et la Direction de la Re-

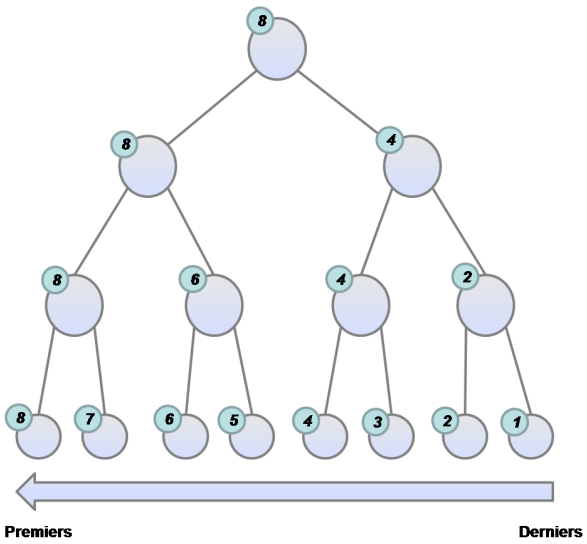


FIGURE 1 – Arbre binaire d'ordonnancement présentant les *scores* attribués à chaque sous-ensemble de $\mathcal{X}$ par la règle de prédiction.

cherche de l'entreprise Renault SA et plus particulièrement, la Direction des Technologies Automobiles Avancées (DTAA). Ces recherches ont contribué à améliorer la méthode de scoring TREERANK, introduite dans [Clémençon & Vayatis 2009d]. D'une part, nous introduisons une procédure, nommée LEAFRANK, pour l'optimisation *locale* du critère ASC à chaque itération de TREERANK, permettant de contrôler la *flexibilité* de l'algorithme. D'autre part, nous proposons deux procédures de *sélection de modèle* permettant de définir de manière automatique la taille *optimale* d'un arbre d'ordonnancement (au sens d'un critère que nous préciserons par la suite). Enfin, nous adaptons deux procédures de *ré-échantillonnage* et de *randomisation*, initialement proposées dans le contexte de la classification et de la régression, afin de réduire l'*instabilité* des règles de prédiction produites, inhérente à la structure hiérarchique de cette méthode de partitionnement. Ce manuscrit est composé de six chapitres dont nous résumons le contenu ci-après.

Le Chapitre 1 est constitué de deux parties, consacrées respectivement à l'introduction des notions *de base*, nécessaires à la description de ces travaux de recherche, et à la présentation de la méthode de scoring TREERANK proposée dans [Clémençon & Vayatis 2008c] et [Clémençon & Vayatis 2009d]. Nous commençons donc par formaliser la problématique d'ordonnancement binaire, avant de définir les deux critères de performance sur lesquels est fondée cette méthode : la courbe Caractéristique de l'Opérateur de Réception (COR) et l'Aire Sous la Courbe COR (ASC). Nous introduisons ensuite la méthode TREERANK, que nous présentons comme une procédure d'approximation de la courbe *optimale*[2] COR^* par une fonction affine par morceaux. Enfin, nous décrivons une première heuristique, fondée sur la scission de l'espace d'entrée $\mathcal{X}$ perpendiculairement à ses axes, dans le même esprit que l'algorithme de classification CART ([Breiman *et al.* 1984]), et nous évaluons ses performances sur des jeux de données simulées.

Les trois chapitres suivants sont consacrés à la description des améliorations apportées

2. au sens de la distance entre courbes COR induite par la norme $\mathcal{L}_\infty$

à la méthode TreeRank au cours de ces travaux de thèse. Dans le Chapitre 2, nous nous focalisons sur le problème de l'optimisation de l'ASC calculée localement sur un sous-ensemble $C \subset \mathcal{X}$, résolu à chaque itération de l'algorithme TreeRank. Nous proposons de résoudre cette étape au moyen d'une procédure, appelée LeafRank, permettant de contrôler la flexibilité de la méthode d'ordonnancement. Nous considérons deux approches différentes. La première consiste à *partitionner* le sous-ensemble C en *cellules élémentaires*, puis de fusionner ces dernières de façon à définir deux sous-ensembles de C, non vides, disjoints et suffisamment *complexes* pour pouvoir s'adapter aux données étudiées. La seconde approche repose quant à elle sur l'interprétation de chaque itération de l'algorithme TreeRank comme un problème de classification binaire pondérée. De ce point de vue, l'optimisation de l'ASC peut donc être résolue au moyen de n'importe quel algorithme de classification. Nous proposons alors deux nouvelles versions de l'algorithme TreeRank, fondées respectivement sur l'implémentation récursive d'une version pondérée de l'algorithme CART ([Breiman *et al.* 1984]) et sur la mise en oeuvre de Machines à Vecteurs Supports ([Schölkopf & Smola 2002], [Taylor & Cristianini 2000]). Enfin, nous concluons ce chapitre par une étude expérimentale afin d'illustrer et de comparer les performances de ces deux heuristiques.

Dans le Chapitre 3, nous considérons le problème de la sélection automatique de la taille *optimale* (au sens du critère ASC) d'un arbre d'ordonnancement. Une fois encore, nous envisageons deux démarches, fondées sur l'optimisation du critère ASC pénalisée par un terme permettant d'évaluer la complexité des modèles comparés. Tout d'abord, nous proposons d'adapter la procédure d'*élagage* proposée dans [Breiman *et al.* 1984] dans le contexte de la classification et de la régression. Cette procédure repose sur l'optimisation de l'ASC pénalisée *linéairement* par le cardinal de la partition engendrée par le modèle sur l'espace d'entrée $\mathcal{X}$. Cette approche présente l'avantage d'être facile à mettre en oeuvre en pratique, mais malheureusement, nous ne disposons pas de résultats théoriques permettant d'étayer le choix de la pénalité ou d'établir la *consistance* des règles de prédiction sélectionnées. Aussi, nous considérons une deuxième approche, fondée cette fois sur l'optimisation de l'ASC *structurelle* ([Vapnik 1982]), qui consiste à optimiser le critère ASC pénalisée par une fonction *non-linéaire* de la complexité du modèle. Nous proposons ainsi deux pénalités (pour deux configurations spécifiques de l'heuristique TreeRank) indépendantes de la distribution des données, reposant sur l'évaluation de la complexité des modèles au moyen de la dimension de Vapnik-Chervonenkis ([Vapnik 1982]). Nous établissons notamment une *inégalité oracle* et la *consistance* des règles d'ordonnancement sélectionnées via ces deux pénalités.

Nous consacrons le Chapitre 4 à l'adaptation de deux procédures de *ré-échantillonnage* et de *randomisation* afin de *stabiliser* et d'améliorer les performances de la méthode TreeRank : le *bagging* ([Breiman 1996b]) et les *forêts aléatoires* ([Breiman 2001]). Dans le contexte de la classification, ces deux procédures visent à construire une collection d'arbres et à les *agréger* selon un principe de *vote à la majorité*, afin de définir une règle de prédiction *moyennée*. Cependant, dans le cas spécifique de l'agrégation d'arbres d'ordonnancement, la règle de prédiction définie par le *vote majoritaire* n'est pas nécessairement *optimale*. Nous considérons donc une approche *métrique*, afin d'établir un consensus parmi les règles de prédiction définies par la collection d'arbres, en minimisant une certaine *distance* entre les ordonnancements produits. Cette approche nous permet ainsi d'adapter les procédures de *bagging* et des *forêts aléatoires* à la problématique d'ordonnancement. Nous concluons ce chapitre par une étude empirique, dans laquelle nous évaluons l'impact de ces deux

procédures sur la *stabilité* et les performances de l'algorithme de scoring TreeRank.

Les deux derniers chapitres de ce manuscrit sont consacrés à des exemples d'applications de la méthode TreeRank. Le Chapitre 5 est constitué de deux parties indépendantes. Dans la première, nous proposons deux séries d'expériences réalisées sur 3 jeux de données simulées et 13 jeux de données réelles, provenant de la banque de données de l'UCI (http ://archive.ics.uci.edu/ml/). Nous comparons, dans un premier temps, huit versions de l'heuristique TreeRank, mettant en oeuvre un algorithme CART ou des classifieurs SVM et éventuellement une procédure de *ré-échantillonnage* voire de *randomisation*. Puis, dans un deuxième temps, nous comparons la méthode de scoring TreeRank avec trois de ses concurrentes : les méthodes RankBoost, RankSVM et RankRLS, respectivement fondées sur une procédure de type *boosting* et sur la mise oeuvre d'heuristiques de type SVM. Enfin, la deuxième partie de ce chapitre est consacrée à l'application industrielle de ces travaux de thèse. Nous y décrivons la problématique d'objectivation de manière générale, avant de présenter l'étude réalisée pour l'objectivation de la prestation *brio*. Nous en profitons pour comparer les performances de la méthode TreeRank avec une approche *plug-in*, basée sur la résolution d'un problème de type LASSO.

Enfin, nous abordons dans le dernier chapitre, le problème du test de l'homogénéité de deux populations en grande dimension. Afin d'étendre le champ d'action des tests de rangs ([Hajek & Sidak 1967]) au cas multi-dimensionnel, nous envisageons d'exploiter l'approche du scoring. En particulier, nous proposons une procédure, fondée sur une étape préalable d'optimisation de l'ASC, permettant de tester l'homogénéité de deux populations définies en grande dimension via le test de la somme des rangs de Wilcoxon. Nous concluons ce dernier chapitre par une étude empirique succinte, dans laquelle nous comparons trois procédures de test sur divers jeux de données simulées.

Liste des publications

Ces travaux de thèse ont fait l'objet de diverses publications et participations à des conférences, listées ci-dessous.

- S. Clémençon, M. Depecker and N. Vayatis, *Adaptive Partitioning Schemes for Bipartite Ranking : How to Grow and Prune a Ranking Tree*, Journal of Machine Learning (Accepted for publication), 2010.

- S. Clémençon, M. Depecker and N. Vayatis, *Bagging Ranking Trees*, Proceedings of IEEE-ICMLA'09, 2009.

- S. Clémençon, M. Depecker and N. Vayatis, ASC *optimization and the two-sample problem*, Proceedings of NIPS'09, 2009.

- S. Clémençon, M. Depecker and A. Saint-Marcoux, *Services Objectivization : a Ranking Approach*, Proceedings of SEDM AICIT'10, 2010.

- N. Baskiotis, S. Clémençon, M. Depecker and N. Vayatis, *TreeRank : a Statistical Software for Bipartite Ranking*, Demonstration session of AISTAT'10, 2010.

– S. Clémençon, M. Depecker et N. Vayatis, *Données avec label binaire : avancées récentes dans le domaine de l'apprentissage statistique d'ordonnancements*, Actes de la conférence CAP'10, 2010.

– S. Clémençon, M. Depecker et N. Vayatis, *Avancées récentes dans le domaine de l'apprentissage d'ordonnancements*, Revue d'Intelligence Artificielle (RIA) (publication acceptée), 2010, (version longue de l'article présenté à la conférence CAP).

D'autres publications sont actuellement en cours de préparation :

– N. Baskiotis, S. Clémençon, M. Depecker and N. Vayatis, *R-implementation of the TreeRank algorithm*, Journal of Machine Learning Research (Accepted for publication).

– S. Clémençon, M. Depecker and N. Vayatis, *An Empirical Comparison of Learning Algorithms for Nonparametric Scoring. The* TreeRank *Algorithm and Other Methods*, Data Mining and Knowledge Discovery (DMKD).

Par ailleurs, nous avons implémenté une version de la méthode TreeRank sous le logiciel Matlab (code propriété Renault) et un package a été développé par N. Baskiotis sous le logiciel libre R. Cette version, ainsi qu'une documentation explicative, sont disponibles à l'adresse : http ://cran.r-project.org/web/packages/TreeRank/index.html.

Première partie

Apprentissage d'Arbres Binaires d'Ordonnancement

Chapitre 1

Méthode TreeRank : Optimisation récursive de la courbe COR

L'objectif de ce chapitre est de présenter la méthode de scoring TreeRank, proposée dans [Clémençon & Vayatis 2008c] et [Clémençon & Vayatis 2009d]. Cette méthode étend la notion d'*arbre de décision* à la problématique d'ordonnancement binaire, les fonctions de score produites pouvant être représentées graphiquement par un arbre binaire orienté, que l'on appellera *arbre de scoring* ou encore *arbre d'ordonnancement*.

Dans cette approche, le problème de l'apprentissage d'une fonction de score repose sur l'optimisation d'un critère empirique, dans l'esprit des méthodes de type ERM (*Empirical Risk Minimization*). Mais contrairement à la plupart des méthodes de l'état de l'art, qui consistent à minimiser une mesure scalaire du *risque* empirique d'ordonnancement, la procédure TreeRank vise à produire des fonctions de score de courbes COR (pour Caractéristique de l'Opérateur de Réception) optimales. Dans [Clémençon & Vayatis 2008c] et [Clémençon & Vayatis 2009d], ce problème est abordé du point de vue de la théorie de l'approximation : l'heuristique TreeRank est introduite comme une version empirique d'une procédure d'approximation récursive et adaptative de la courbe optimale COR* par une fonction affine par morceaux.

La première partie de ce chapitre est consacrée à la formalisation de la problématique d'ordonnancement binaire, que nous considérons ici comme un problème de scoring. Nous y définissons notamment la notion de fonction de score, avant d'introduire deux critères dédiés à l'évaluation de la performance de règles d'ordonnancement : la courbe Caractéristique de l'Opérateur de Réception (COR) et l'Aire Sous la Courbe COR (ASC). Afin d'introduire la méthode TreeRank, nous décrivons, dans un premier temps, le principe de l'approximation de la courbe optimale COR* par une fonction affine par morceaux, que nous relions au problème de l'optimisation récursive du critère ASC. Dans un deuxième temps, nous présentons le schéma d'approximation arborescent sous-jacent à l'algorithme TreeRank, dont nous détaillons ensuite les différentes étapes. Enfin, nous fournissons des résultats théoriques relatifs à la convergence des fonctions de score estimées au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$ définies sur l'espace des courbes COR et présentons une rapide étude empirique afin d'illustrer les performances de cette approche. Nous ne reproduisons pas les preuves des résultats théoriques énoncés dans ce chapitre, qui peuvent être trouvées principalement dans les contributions [Clémençon & Vayatis 2008c]

et [Clémençon & Vayatis 2009d] ainsi que dans les différentes références indiquées en tête de chaque énoncé.

1.1 Un problème de scoring

Tout au long de ce manuscrit, nous allons observer des populations d'individus -ou des collections d'objets- caractérisés à la fois par un ensemble de *variables -entrées*, *prédicteurs-* et par une *étiquette -label*, *classe- binaire*, *i.e.* prenant exactement deux valeurs, par exemple « $+1$ » ou « -1 » ou encore « bon » ou « mauvais ». On parlera de données *étiquetées de façon binaire*. Lorsque l'on dispose de telles données, l'objectif le plus courant est de prédire l'étiquette d'une nouvelle *observation*, *i.e.* d'un nouvel individu. Ce problème d'apprentissage supervisé est bien connu dans la littérature : il s'agit de la classification binaire. Cependant, il existe aussi de nombreuses applications pour lesquelles l'enjeu principal n'est pas de prédire les labels pour de nouvelles observations, mais plutôt de définir une *relation d'ordre* sur la population, afin de hiérarchiser les individus qui la composent. Nous désignerons cette problématique sous le terme d'*ordonnancement binaire*.

Comme nous l'avons indiqué dans le chapitre d'introduction, on peut envisager différentes approches pour résoudre un tel problème. De nombreuses méthodes reposent, par exemple, sur l'analyse et la comparaison des *paires* d'observations et produisent des *ordonnancements* en agrégeant les *préférences* ainsi observées. C'est une tout autre approche que nous avons retenue dans nos travaux, la problématique d'ordonnancement étant considérée ici comme un problème de *scoring*. L'objectif de cette partie est de formaliser le problème d'ordonnancement binaire vu sous cet angle.

1.1.1 Problématique d'ordonnancement binaire

Bien que les deux problématiques d'ordonnancement et de classification binaire diffèrent par leurs objectifs, elles sont toutes deux définies dans le même cadre probabiliste, que l'on peut formuler comme suit : soit un couple de variables aléatoires

$$(X, Y) \in \mathcal{X} \times \{-1, +1\},$$

où $X \in \mathcal{X} \subset \mathbb{R}^d$, $d \geq 1$, représente le vecteur des prédicteurs permettant de modéliser l'étiquette binaire Y. La distribution $\mathcal{P}_{X,Y}$ du couple (X, Y) est entièrement caractérisée par les distributions conditionnelles $G(dx)$ et $H(dx)$ de la variable aléatoire X sachant les événements $\{Y = +1\}$ et $\{Y = -1\}$ respectivement ou, de manière équivalente, par le couple (μ, η), où $\mu(dx)$ désigne la loi marginale de X et, $\forall\, x \in \mathcal{X}$:

$$\eta(x) = \mathbb{P}\{Y = +1 \mid X = x\}, \tag{1.1}$$

est la probabilité à posteriori, correspondant, à une transformation affine près, à la fonction de régression. Soit $p = \mathbb{P}\{Y = +1\}$ le taux théorique d'étiquettes positives.

Afin de se placer dans un cadre suffisamment complexe, où les observations des distributions $G(dx)$ et $H(dx)$ sont non séparables, on suppose que les distributions $G(dx)$ et $H(dx)$ sont continues et satisfont les deux hypothèses suivantes :

– $(\mathbf{A_1})$: $G(dx)$ et $H(dx)$ sont *équivalentes*, au sens où chacune est absolument continue par rapport à l'autre. De plus, le rapport de vraisemblance $(dG/dH)(X)$ est borné, ou en d'autres termes, le supremum de la variable aléatoire $\eta(X)$ est strictement inférieur à 1.

– $(\mathbf{A_2})$: la distribution de la variable aléatoire $\eta(X)$ est absolument continue par rapport à la mesure de Lebesgue.

En pratique, le problème d'ordonnancement binaire consiste à apprendre un modèle, à partir d'un échantillon $\mathcal{D}_n = \{(X_i, Y_i), 1 \leq i \leq n\}$ de copies *i.i.d.* (indépendantes et identiquement distribuées) du couple (X, Y), afin d'ordonner un ensemble $\mathbf{X}_m$ de nouvelles observations $\{X_1^{'}, \cdots, X_m^{'}\}$, $m \in \mathbb{N}^*$, de la variable aléatoire X, pour lesquelles l'étiquette Y n'est pas observée. En d'autres termes, à partir de l'observation de $\mathcal{D}_n$, on souhaite induire une relation d'ordre sur l'espace d'entrée $\mathcal{X}$, faisant apparaître les observations positives en tête du classement avec une forte probabilité et, dans le même temps, les observations négatives en queue de classement, voir la Figure 1.1 ci-dessous :

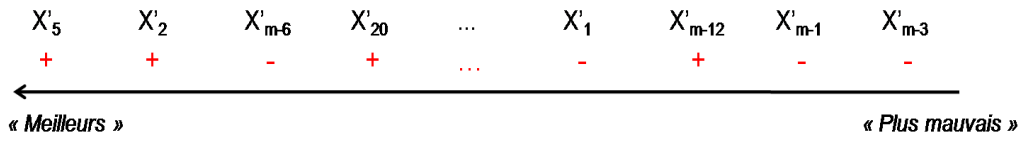


FIGURE 1.1 – Relation d'ordre définie sur les objets de $\mathcal{X}$.

Intuitivement, on perçoit que l'ordre induit sur $\mathcal{X}$ par la probabilité à posteriori η est *optimal*[1]. En effet, la probabilité η permet d'ordonner les instances de $\mathcal{X}$ en positionnant en tête de classement les instances ayant la plus forte probabilité d'être *positives* et en queue de classement, les instances les plus probablement *négatives*. De ce point de vue, le problème de scoring revient à estimer l'ordre induit sur $\mathcal{X}$ par la fonction de régression, ou de manière équivalente par la probabilité η.

1.1.2 Fonction de score

L'approche la plus classiquement utilisée pour induire une relation d'ordre sur un espace $\mathcal{X}$ de grande dimension consiste à *projeter* les observations de cet espace sur la droite des réels par le biais d'une *fonction de score*, puis à utiliser l'ordre naturel sur $\mathbb{R}$ pour les ordonner.

Définition 1 *(Fonction de score)*
Une fonction de score $s : \mathcal{X} \to \mathbb{R}$ est une fonction borélienne, qui attribue une valeur réelle, appelée score, à chaque observation de $\mathcal{X}$. On notera $\mathcal{S}$ l'ensemble des fonctions de score définies sur l'espace $\mathcal{X}$.

Une fonction de score $s : \mathcal{X} \to \mathbb{R}$ définit un *pré-ordre* sur l'espace $\mathcal{X}$, noté $\preccurlyeq_s$, *i.e.* une relation d'ordre non nécessairement réflexive. Dans la suite, on parlera aussi de *règle de score* ou de *règle d'ordonnancement*.

1. au sens d'un critère que nous définirons ultérieurement.

Définition 2 *(Pré-ordre sur $\mathcal{X}$)*
Soit $s \in \mathcal{S}$ une fonction de score, la règle de score $\preccurlyeq_s$ sur $\mathcal{X}$ est un pré-ordre, i.e. une relation binaire sur $\mathcal{X}$ satisfaisant les deux propriétés suivantes :

- *(***Totalité***)* $\forall\,(x_1, x_2) \in \mathcal{X}^2$*, on a soit* $x_1 \preccurlyeq_s x_2$*, soit* $x_2 \preccurlyeq_s x_1$*.*
- *(***Transitivité***)* $\forall\,(x_1, x_2, x_3) \in \mathcal{X}^3$*, si* $x_1 \preccurlyeq_s x_2$ *et* $x_2 \preccurlyeq_s x_3$*, alors* $x_1 \preccurlyeq_s x_3$*.*

Avec ces notations, une observation $x_1 \in \mathcal{X}$ sera dite *meilleure* qu'une observation $x_2 \in \mathcal{X}$ selon s, si et seulement si $s(x_2) \leq s(x_1)$ et on notera $x_2 \preccurlyeq_s x_1$. D'après la Définition 2, le pré-ordre $\preccurlyeq_s$ n'est pas nécessairement *anti-symétrique*, à moins qu'il ne soit stipulé de manière explicite qu'une fonction $s \in \mathcal{S}$ ne peut attribuer un même score à deux observations distinctes de $\mathcal{X}$. Ceci signifie que $\forall\,(x_1, x_2) \in \mathcal{X}^2$, l'on n'a pas forcément l'égalité $x_1 = x_2$ si l'on observe à la fois $x_1 \preccurlyeq_s x_2$ et $x_2 \preccurlyeq_s x_1$.

Dans la suite, on qualifiera d'*ex-aequo* des observations $(x_1, x_2) \in \mathcal{X}^2$ ayant le même score, *i.e.* vérifiant simultanément les deux relations précédentes : $x_1 \preccurlyeq_s x_2$ et $x_2 \preccurlyeq_s x_1$. On les notera alors $x_1 \asymp x_2$ et dans le cas particulier où seule la relation $x_1 \preccurlyeq_s x_2$ est vérifiée, on pourra noter $x_2 \succ x_1$.

1.1.3 Fonction de score optimale

Notre objectif étant d'estimer l'ordre induit sur $\mathcal{X}$ par la probabilité η, une fonction de score $s^* \in \mathcal{S}$ sera dite *optimale* si elle induit sur $\mathcal{X}$ le même ordre que η, *i.e.* si

$$\forall\ x,\ x' \in \mathcal{X},\ s^*(x) - s^*(x') > 0 \Rightarrow \eta(x) - \eta(x') > 0.$$

On peut aisément en déduire que l'ensemble $\mathcal{S}^*$ des fonctions de score *optimales*, solutions du problème d'ordonnancement binaire, est l'ensemble des transformées strictement croissantes de la probabilité à posteriori :

$$\mathcal{S}^* = \{s^* = T \circ \eta \mid T : [0,1] \to \mathbb{R},\ \text{strictement croissante}\}\,. \tag{1.2}$$

La Proposition 1 suivante permet de définir plus précisément ce problème d'estimation, en établissant un lien explicite entre une fonction de score *optimale* $s^* \in \mathcal{S}^*$ bornée et la distribution $\mathcal{P}_{X,Y}$ des données, par le biais de la probabilité η.

Proposition 1 *(Fonction de score optimale [Clémençon & Vayatis 2009d])*
Une fonction de score bornée s^* est optimale si et seulement si il existe une fonction w, positive et intégrable, et une variable aléatoire continue $V \in]0,1[$ telles que :

$$\forall x \in \mathcal{X}\ ,\quad s^*(x) = \inf_{x \in \mathcal{X}} s^*(x) + \mathbb{E}\left[w(V) \cdot \mathbb{I}\{\eta(x) > V\}\right]. \tag{1.3}$$

Dans le cas particulier où l'on considère la probabilité à posteriori η, l'identité devient :

$$\forall x \in \mathcal{X}\ ,\quad \eta(x) = \mathbb{E}\left[w(U) \cdot \mathbb{I}\{\eta(x) > U\}\right], \tag{1.4}$$

où U est une variable aléatoire uniforme sur $]0,1[$ et la fonction w est l'indicatrice du support de la variable aléatoire $\eta(X)$.

L'Identité (1.3) montre qu'une fonction de score optimale $s^* \in \mathcal{S}^*$ peut être entièrement caractérisée par une collection $\mathcal{C}^* = \{ \{x \in \mathcal{X} \mid \eta(x) > u\}, u \in]0,1[\}$, qui constitue ce que l'on appelle les *ensembles de niveaux* de la probabilité à posteriori η. Pour une valeur de u fixée dans $]0,1[$, $C = \{x \in \mathcal{X} \mid \eta(x) > u\}$ est le sous-ensemble de $\mathcal{X}$ contenant les observations dont la probabilité d'être *positives* est supérieure au niveau u.

Une solution naturelle pour définir une relation d'ordre sur l'espace d'entrée $\mathcal{X}$, consiste à estimer directement la probabilité à posteriori, c'est le principe des méthodes de type *plug-in*. Cependant, la Proposition 1 montre qu'il n'est pas nécessaire de résoudre ce problème d'estimation, d'autant plus délicat que la dimension de l'espace $\mathcal{X}$ est grande. En effet, d'après le résultat énoncé ci-dessus, pour retrouver l'ordre induit sur $\mathcal{X}$ par la probabilité η, il *suffit* d'estimer la collection $\mathcal{C}^*$ de ses ensembles de niveaux.

Les travaux que nous présentons dans ce manuscrit abordent ce problème selon le principe de l'approche ERM (*Empirical Risk Minimization*), *i.e.* l'apprentissage d'une règle de score définie sur l'espace $\mathcal{X}$ repose sur l'optimisation d'un critère empirique. On parlera de M-estimation d'une fonction de score $s^* \in \mathcal{S}^*$ optimale. Nous introduisons, dans la Partie 1.2 suivante, les deux critères d'évaluation de la performance d'une règle de score que nous avons utilisés dans nos travaux.

1.2 Mesures de performance d'une règle de score

Divers critères ont été proposés dans la littérature pour évaluer la performance d'une règle de score. Les critères les plus populaires sont de nature fonctionnelle, comme la courbe Caractéristique de l'Opérateur de Réception (COR) ou la courbe Rappel-Précision (RP). Ces dernières permettent de comparer visuellement les performances de différentes relations d'ordre. Cependant, l'optimisation de ces critères fonctionnels par le biais d'algorithmes d'apprentissage est une tâche difficile. C'est pourquoi, en pratique, on leur préfère des critères de nature scalaire, comme par exemple, l'Aire Sous la Courbe COR (ASC) ou encore la précision moyenne (MAP pour *Mean Average Precision*) ou le critère NDCG (*Normalized Discounted Cumulative Gain*), deux critères introduits dans le contexte de la recherche automatique de documents (*Information Retrieval*) (voir [Yates & Ribeiro-Neto 1999] et [Järvelin & Kekäläinen 2000]).

Dans ces travaux de thèse, nous nous sommes focalisés sur deux critères : l'un fonctionnel, la courbe COR et l'autre scalaire, l'ASC. Nous les définissons ici et présentons leurs principales propriétés.

1.2.1 Courbe COR

La courbe Caractéristique de l'Opérateur de Réception (COR) a été utilisée pour la première fois pendant la seconde guerre mondiale pour la détection d'anomalies dans les signaux sonars (*Signal Detection Theory*, [Egan 1975]). Depuis, cet outil est devenu très populaire, notamment dans le domaine bio-médical ([Green & Swets 1974], [Swets 1979], [Hanley & McNeil 1982], [Obuchowski 2003]) ou encore en météorologie ([Mason 1982], [Harvey *et al.* 1992]). Introduit plus récemment dans le contexte de l'apprentissage automatique, il est utilisé notamment pour évaluer et comparer les performances de di-

vers algorithmes d'apprentissage ([Spackman 1989], [Bradley 1997], [Bradley *et al.* 1994]). L'introduction de la notion de courbe COR que nous proposons ci-dessous est largement inspirée de [Fawcett 2006], nous renvoyons à la publication d'origine et aux références qui y sont citées pour une description plus détaillée.

1.2.1.1 Mesures de performance d'un classifieur binaire

La notion de courbe COR a été initialement introduite dans le contexte de la classification binaire. Elle repose sur des critères classiquement utilisés pour évaluer la performance de classifieurs binaires, que nous rappelons ici. Soit C un classifieur discret de la forme $C(X) = 2 \cdot \mathbb{I}\{f_C(X) \geq z\} - 1$, pour tout $X \in \mathcal{X}$, où la fonction f_C est à valeurs dans $\mathbb{R}$ et $z \in \mathbb{R}$ est le seuil de discrimination de C. A une observation $X \in \mathcal{X}$, ce classifieur associe une étiquette binaire $\widehat{Y} = C(X)$ à valeurs dans $\{-1, +1\}$. On peut identifier 4 configurations différentes, que l'on résume dans la *matrice de confusion* du classifieur C, schématisée sur la Figure 1.2 :

a) C associe une étiquette *positive* à l'instance X *positive*, X est un *vrai positif*,
b) C associe une étiquette *positive* à l'instance X *négative*, X est un *faux positif*,
c) C associe une étiquette *négative* à l'instance X *négative*, X est un *vrai négatif*,
d) C associe une étiquette *négative* à l'instance X *positive*, X est un *faux négatif*.

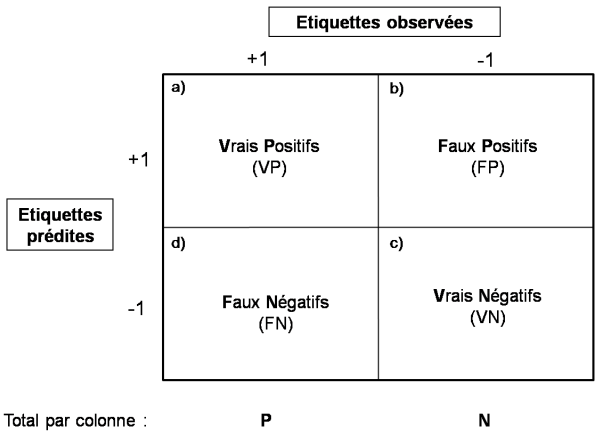


Figure 1.2 – Matrice de confusion d'un classifieur binaire C.

De cette matrice de confusion, on peut extraire divers critères pour mesurer la performance du classifieur C. On définit notamment :

– *le taux de vrais positifs*, encore appelé *rappel* ou *sensibilité* : il s'agit du taux d'étiquettes positives correctement prédites par C

$$\mathrm{TVP_C} = \mathbb{P}\{\mathrm{C(X)} = +1 \mid \mathrm{Y} = +1\} \approx \frac{\mathrm{VP}}{\mathrm{P}}, \tag{1.5}$$

– *le taux de faux positifs* : il s'agit du taux d'étiquettes négatives mal classées par C

$$\mathrm{TFP_C} = \mathbb{P}\{\mathrm{C(X)} = +1 \mid \mathrm{Y} = -1\} \approx \frac{\mathrm{FP}}{\mathrm{N}}. \tag{1.6}$$

Le plan défini par le taux de faux positifs (TFP) en abscisses, et le taux de vrais positifs (TVP) en ordonnées, permet de visualiser la performance de classifieurs binaires. Dans ce plan, le classifieur C définit un unique point, dont les coordonnées $(\mathrm{TFP_C}, \mathrm{TVP_C})$ correspondent à ses taux de faux positifs et vrais positifs respectivement.

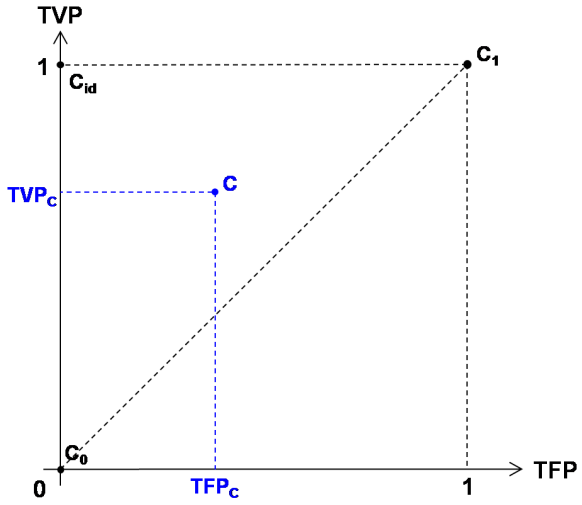


FIGURE 1.3 – Plan $(\mathrm{TFP}, \mathrm{TVP})$.

Sur la Figure 1.3, nous avons représenté 3 points extrêmes du plan, qui correspondent à 3 classifieurs remarquables. Le point C_0, de coordonnées $(0; 0)$, représente un classifieur *ne prédisant aucune étiquette positive*. Par construction, ses taux de faux et vrais positifs sont nuls. Le point C_1, de coordonnées $(1; 1)$, correspond à l'extrême inverse, *i.e.* à un classifieur *ne prédisant que des étiquettes positives*. Le taux de vrais positifs associé à C_1 est donc maximal, mais son taux de faux positifs l'est également. Enfin, le point C_{id}, de coordonnées $(0; 1)$, correspond au classifieur *optimal* dans le cas *séparable*, *i.e.* quand les instances positives et négatives peuvent être parfaitement séparées en deux classes. En effet, ce classifieur présente simultanément un taux de vrais positifs maximal et un taux de faux positifs nul.

La diagonale principale du plan $(\mathrm{TVP}, \mathrm{TFP})$ caractérise une autre collection de classifieurs remarquables : tout point situé sur ce segment correspond à un classifieur *prédisant une classe de façon totalement aléatoire*. Considérons par exemple un classifieur prédisant aléatoirement 50% d'instances positives sur l'échantillon observé. En probabilité, celui-ci estimera correctement la moitié des instances positives et la moitié des instances négatives. Son taux de vrais positifs sera donc égal à 0.50, de même que son taux de faux positifs.

Remarque 1 *(Espace en-dessous de la diagonale)*
Un classifieur se situant en-dessous de la diagonale principale du plan $(\mathrm{TFP}, \mathrm{TVP})$ *n'apporte aucune information sur les données. Sa performance est en-deçà de celle d'un classifieur aléatoire, qui serait représenté par un point de la diagonale. Comme cela est indiqué dans [Fawcett 2006], il suffit de prendre la négation de ce classifieur, i.e. d'inverser la prédiction sur chaque instance, pour le transporter au-dessus de la diagonale.*

L'environnement graphique défini par le plan $(\mathrm{TFP}, \mathrm{TVP})$ permet de comparer aisément la performance d'un ensemble de classifieurs. Considérons, par exemple, les points

représentés sur la Figure 1.4 ci-dessous. Si l'on fixe le taux de faux positifs à la valeur $\mathrm{TFP_A}$, on voit directement sur le graphe que le classifieur A est plus performant que le classifieur C, au sens où son taux de vrais positifs est plus élevé. De manière analogue, si l'on fixe cette fois-ci le taux de vrais positifs à la valeur $\mathrm{TVP_A}$, le graphe nous permet de conclure que le classifieur A est aussi plus performant que le classifieur B étant donné que son taux de faux positifs est plus faible.

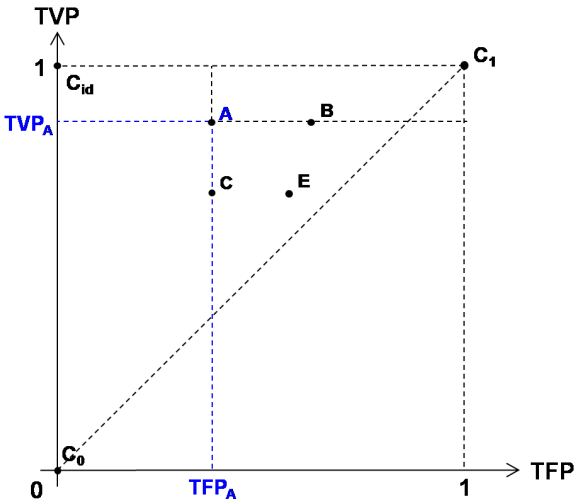


Figure 1.4 – Comparaison de classifieurs dans le plan $(\mathrm{TFP}, \mathrm{TVP})$.

En réalité, plus les coordonnées d'un classifieur seront proches du point C_{id}, plus celui-ci sera performant, au sens de l'erreur -ou du coût- de classification. En effet, si l'on considère les deux classifieurs A et E représentés sur la Figure 1.4, le classifieur A a un taux de vrais positifs plus élevé que celui du classifieur E et dans le même temps, son taux de faux positifs est plus faible. Ainsi, l'erreur de classification associée au classifieur A, que l'on peut évaluer en sommant le nombre de faux positifs avec le nombre de faux négatifs, est plus faible que l'erreur associée à E. (Notons cependant que la réciproque de ce résultat est fausse.)

1.2.1.2 Courbe COR d'une collection de classifieurs binaires

Dans le contexte de la classification binaire, la notion de *courbe* COR prend tout son sens dans le cas particulier où le coût d'erreur de classification associé à chacune des classes est inconnu (voir notamment [Provost & Fawcett 1997], [Barreno *et al.* 2007], [Bach *et al.* 2005] et [Bach *et al.* 2006]). En effet, dans de nombreuses applications, on ne souhaite pas seulement contrôler le nombre -ou le taux- d'étiquettes mal prédites, *i.e.* l'erreur de classification, mais plutôt le coût global $\mathbf{C}_C$ associé au classifieur C, qui correspond au nombre -taux- de faux positifs FP et de faux négatifs FN pondérés respectivement par les coûts $\mathbf{c}_{\mathrm{FP}}$ et $\mathbf{c}_{\mathrm{FN}}$, associés à l'erreur de classification d'une observation respectivement négative et positive :

$$\mathbf{C}_C \approx \mathrm{FP} \cdot \mathbf{c}_{\mathrm{FP}} + \mathrm{FN} \cdot \mathbf{c}_{\mathrm{FN}}.$$

Dans la plupart des cas cependant, les coûts $\mathbf{c}_{\mathrm{FP}}$ et $\mathbf{c}_{\mathrm{FN}}$ sont inconnus. Une solution consiste alors à optimiser le taux de faux positifs pour un taux de vrais positifs fixé et

vice versa : c'est la méthode de Neyman-Pearson ([Neyman & Pearson 1933], voir aussi [Bradley 1997]). Afin de sélectionner la solution la plus adaptée aux contraintes du problème, la procédure la plus simple consiste à représenter les taux de vrais et faux positifs d'un classifieur binaire C tout en faisant varier son seuil de discrimination. On obtient alors la courbe COR de la collection $\mathcal{C}_C$ de classifieurs définis à partir de C.

Définition 3 *(Courbe* COR *d'une collection de classifieurs binaires)*
Soit un classifieur binaire de la forme $C_z(x) = 2 \cdot \mathbb{I}\{f_C(x) \geq z\} - 1$*, pour tout* $x \in \mathcal{X}$*, où la fonction* f_C *est à valeurs dans* $\mathbb{R}$ *et* $z \in \mathbb{R}$ *est un seuil fixé a priori. Soit* $\mathcal{C}_C$ *la collection de classifieurs binaires obtenue en faisant varier le seuil de discrimination* z*, telle que*

$$\forall\, x \in \mathcal{X},\, \mathcal{C}_C = \{C_z(x) = 2 \cdot \mathbb{I}\{f_C(x) \geq z\} - 1,\ z \in \mathbb{R}\}.$$

La courbe COR *associée à la collection* $\mathcal{C}_C$ *est la courbe paramétrée, qui à tout* $z \in \mathbb{R}$ *associe un couple* $(\mathrm{TFP}_{\mathrm{C_z}}, \mathrm{TVP}_{\mathrm{C_z}}) \in [0,1]^2$*.*

1.2.1.3 Courbe COR associée à une fonction de score

Maintenant que la notion de courbe COR a été définie dans le contexte de la classification binaire, sa transposition au problème du scoring repose sur le fait que toute fonction de score $s \in \mathcal{S}$ définit une collection $\mathcal{C}_s$ de classifieurs binaires, paramétrée par un seuil $z \in \mathbb{R}$, et de la forme suivante :

$$\mathcal{C}_s = \{C_z(x) = 2 \cdot \mathbb{I}\{s(x) \geq z\} - 1,\ z \in \mathbb{R}\}.$$

Comme nous venons de le voir, chaque classifieur $C_z \in \mathcal{C}_s$ définit un unique point dans le plan $(\mathrm{TFP}, \mathrm{TVP})$, dont les coordonnées correspondent à ses taux de faux et vrais positifs

$$\begin{aligned} \mathrm{TFP}_{\mathrm{C_z}} &= \mathbb{P}\{C_z(X) = +1 \mid Y = -1\} = \mathbb{P}\{s(X) > z \mid Y = -1\}, \\ \mathrm{TVP}_{\mathrm{C_z}} &= \mathbb{P}\{C_z(X) = +1 \mid Y = +1\} = \mathbb{P}\{s(X) > z \mid Y = +1\}, \end{aligned}$$

et l'ensemble des points définis par la collection $\mathcal{E}_s$ constitue la courbe COR associée à la fonction de score s.

Définition 4 *(Courbe* COR *associée à une fonction de score)*
Soient α_s *et* β_s *les taux théoriques de faux et vrais positifs associés à une fonction de score* $s \in \mathcal{S}$*.* $\forall\ z \in \mathbb{R}$*, on a*

$$\begin{aligned} \alpha_s(z) &= \mathbb{P}\{s(X) > z \mid Y = -1\}, \\ \beta_s(z) &= \mathbb{P}\{s(X) > z \mid Y = +1\}. \end{aligned}$$

Avec ces notations, la courbe COR *associée à la fonction* s *est la courbe paramétrée suivante :*

$$\begin{aligned} \mathrm{COR(s)} \ :\ \mathbb{R} &\to [0,1]^2 \\ z &\to (\alpha_s(z), \beta_s(z)). \end{aligned} \tag{1.7}$$

De plus, notons G_s et H_s les fonctions de répartition conditionnelles de la variable aléatoire $s(X)$ sachant les événements $\{Y = +1\}$ et $\{Y = -1\}$ respectivement, que l'on suppose continues et satisfaisant l'hypothèse $(\mathbf{A_1})$. Les taux théoriques de vrais et faux positifs de la fonction s coïncident respectivement avec les fonctions de répartition conditionnelles résiduelles $\bar{G}_s$ et $\bar{H}_s$. $\forall\, z \in \mathbb{R}$:

$$\alpha_s(z) = \mathbb{P}\{s(X) > z \mid Y = -1\} = \bar{H}_s(z) = 1 - H_s(z), \tag{1.8}$$

$$\beta_s(z) = \mathbb{P}\{s(X) > z \mid Y = +1\} = \bar{G}_s(z) = 1 - G_s(z). \tag{1.9}$$

Avec ces nouvelles notations, on peut assimiler la courbe COR associée à s, à une fonction du taux de faux positifs $\alpha \in [0, 1]$. Soit $\mathcal{Q}_s(\alpha)$ le quantile d'ordre $(1-\alpha)$ de la distribution conditionnelle $H_s(dx)$, pour tout $\alpha \in [0, 1]$, on a :

$$\begin{aligned} \mathrm{COR}(\mathrm{s}, \alpha) &= \bar{G}_s \circ \bar{H}_s^{-1}(\alpha) \\ &= \bar{G}_s \circ Q_s(\alpha), \\ &= 1 - G_s \circ H_s^{-1}(1-\alpha), \end{aligned} \tag{1.10}$$

où $\bar{H}_s^{-1}$ est l'inverse généralisée de la fonction $\bar{H}_s$ *càdlàg* (continue à droite et limitée à gauche) :

$$\bar{H}_s^{-1}(\alpha) = \inf\{t \in \mathbb{R} \mid \bar{H}_s(t) \geq \alpha\} = \mathcal{Q}_s(\alpha). \tag{1.11}$$

Remarque 2 *(Conventions* COR*)*
Lorsque les distributions conditionnelles $G_s(dx)$ et $H_s(dx)$ présentent des discontinuités, les points de la courbe COR *correspondant à des sauts dans les distributions sont reliés par continuité. Dans la littérature, deux conventions ont été considérées :*

- *(Conv. 1) : les points de* COR *sont reliés par des segments constants,*
- *(Conv. 2) : les points de* COR *sont reliés par des segments linéaires.*

Dans le cadre de nos travaux, nous avons opté pour la deuxième convention. Aussi, lorsque l'on considèrera des distributions conditionnelles discrètes, la courbe COR *associée sera affine par morceaux.*

Depuis son introduction dans le domaine du traitement du signal, la courbe COR est devenue un outil très prisé dans de nombreux domaines. Si cette popularité réside en grande partie dans la représentation visuelle fournie par ce critère, elle repose aussi sur ses bonnes propriétés. Nous en rappelons quelques unes dans la Proposition 2 suivante (voir notamment [van Trees 1968] et [Fawcett 2006] pour plus de détails et de références).

Proposition 2 *(Propriétés de la courbe* COR *([van Trees 1968])*
Pour toute distribution $\mathcal{P}_{X,Y}$ et toute fonction de score $s \in \mathcal{S}$, en utilisant la formulation (1.7), la courbe COR(s, .) *associée à s satisfait les propriétés suivantes :*

- *(***Valeurs limites***) :* $\mathrm{COR}(\mathrm{s}, 0) = 0$ *et* $\mathrm{COR}(\mathrm{s}, 1) = 1$.
- *(***Invariance***) : pour toute fonction $T : \mathbb{R} \to \mathbb{R}$ strictement croissante et pour tout $\alpha \in (0, 1)$, on a :* $\mathrm{COR}(\mathrm{T} \circ \mathrm{s}, \alpha) = \mathrm{COR}(\mathrm{s}, \alpha)$.

- *(***Concavité***) : si le rapport de vraisemblance dG_s/dH_s est monotone alors la courbe* $\mathrm{COR}(s,.)$ *est concave.*
- *(***Linéarité***) : si le rapport de vraisemblance dG_s/dH_s est constant sur une partie du domaine de définition de la fonction de score s alors la courbe* $\mathrm{COR}(s,.)$ *est linéaire sur l'intervalle correspondant.*
- *(***Différentiabilité***) : supposons que la distribution μ de X soit continue, alors la courbe* $\mathrm{COR}(s,.)$ *est différentiable si et seulement si la distribution conditionnelle de $s(X)$ sachant Y est continue.*

La visualisation d'une collection de courbes $\{\mathrm{COR}(s,.)\}_{s\in\mathcal{S}}$ permet de comparer les performances des fonctions de score associées. Par analogie avec le cadre de la classification binaire, on peut identifier aisément la courbe $\mathrm{COR}_{\mathrm{id}}$, optimale dans le cas séparable, comme étant la ligne brisée en deux segments reliant les points $(0;0)$, $(0;1)$ et $(1;1)$ dans le plan $(\mathrm{TFP},\mathrm{TVP})$ (voir la Figure 1.5). Une courbe COR sera d'autant *meilleure* qu'elle sera *proche* de $\mathrm{COR}_{\mathrm{id}}$, au sens d'une norme définie a priori (cf Partie 1.3). Ainsi, les courbes COR induisent une relation d'ordre sur l'espace $\mathcal{S}$ des fonctions de score. Si l'on considère deux fonctions de score quelconques s_1 et s_2 de $\mathcal{S}$, la fonction s_1 sera plus performante que s_2 si et seulement si

$$\forall\ \alpha \in]0,1[\ ,\ \mathrm{COR}(s_1,\alpha) \geq \mathrm{COR}(s_2,\alpha). \tag{1.12}$$

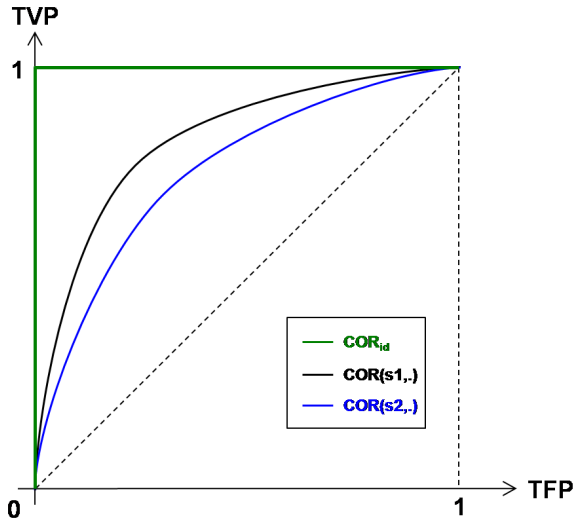


FIGURE 1.5 – Courbes COR associées à des fonctions de score.

Soulignons cependant, que la relation d'ordre induite sur $\mathcal{S}$ par les courbes COR est seulement *partielle*. En effet, selon la relation (1.12), la fonction de score s_1 est plus performante que s_2 si et seulement si sa courbe $\mathrm{COR}(s_1,.)$ est *uniformément supérieure* à $\mathrm{COR}(s_2,.)$, *i.e.* en tout point $\alpha \in [0,1]$. Par contre, ce critère ne permet pas de discriminer les performances de deux fonctions de score s_1 et s_2 de même ASC, mais dont les courbes COR s'entrecroisent, comme sur la Figure 1.6.

1.2.1.4 Courbe optimale COR^*

L'objectif de ce paragraphe est de justifier l'utilisation de la courbe COR comme critère d'évaluation de la performance de fonctions de score dans le cadre du problème d'ordonnan-

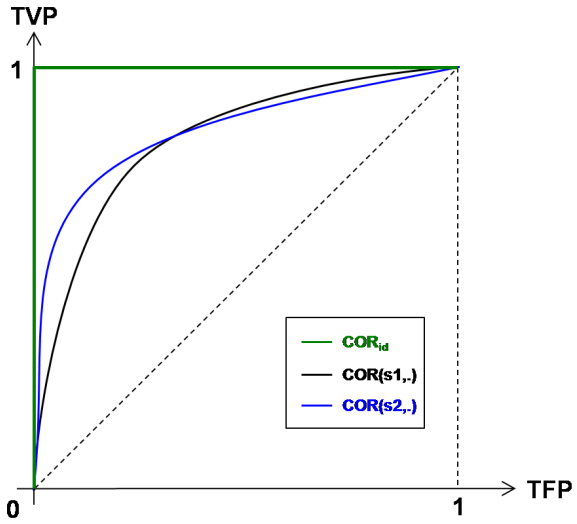


Figure 1.6 – Courbes COR *entrecroisées*.

cement binaire. Pour cela, nous allons montrer que la courbe COR associée à la probabilité à posteriori η est *optimale*, au sens où elle domine uniformément toutes les autres. Dans la suite, on utilisera la notation $\mathrm{COR}^* = \mathrm{COR}(\eta, .)$. Le résultat énoncé dans la Proposition 3 repose sur la considération de la courbe COR du point de vue de la théorie des tests statistiques.

Soit $s \in \mathcal{S}$ une fonction de score continue, de distributions conditionnelles $G_s(dx)$ et $H_s(dx)$ satisfaisant l'hypothèse $(\mathbf{A_1})$. La courbe COR(s, .) associée à s peut être interprétée comme la courbe de *puissance* du test d'hypothèse

$$\mathcal{H}_0 \ : \ X \sim H(dx) \ \text{ versus } \ \mathcal{H}_1 \ : \ X \sim G(dx), \tag{1.13}$$

fondé sur la statistique de test $s(X)$ ou en d'autres termes, si l'on considère le couple de variables aléatoires $(s(X), Y) \in \mathcal{X} \times \{-1, 1\}$, du test de l'hypothèse $\mathcal{H}_0 : Y = -1$ contre l'*alternative* $\mathcal{H}_1 : Y = +1$.
Dans la théorie des tests statistiques, on définit de manière classique deux types d'erreurs :

- l'*erreur de type I* ou *erreur de première espèce* est la probabilité de rejeter *à tort* l'hypothèse nulle $\mathcal{H}_0$,

- l'*erreur de type II* ou *erreur de deuxième espèce* est la probabilité d'accepter *à tort* l'hypothèse nulle $\mathcal{H}_0$.

Idéalement, on souhaite pouvoir minimiser simultanément les erreurs de première et de deuxième espèce. Ce n'est malheureusement pas réalisable en pratique, où l'on ne dispose que d'un échantillon fini d'observations pour calculer la statistique de test. Il faut établir un compromis. Traditionnellement, on contrôle l'erreur de type I en fixant un seuil de tolérance, que l'on appelle le *niveau* du test. On peut alors évaluer la performance du test en calculant sa *puissance*, égale à 1 *moins* l'erreur de deuxième espèce.

Dans le cas du test (6.2) basé sur la statistique $s(X)$, l'erreur de type I correspond à

la prédiction d'un *faux positif* et l'erreur de type II est associée à la prédiction d'un *faux négatif*. Le risque de première espèce est donc mesuré par le taux de faux positifs α_s. Quant au taux de vrais positifs β_s, il correspond à la puissance du test. La courbe COR(s, .) fait donc correspondre à chaque *niveau* $\alpha_s \in [0,1]$ du test, sa *puissance* $\beta_s \in [0,1]$. Or, d'après le lemme de Neyman-Pearson ([Neyman & Pearson 1933]), on sait que le test basé sur le rapport de vraisemblance

$$\Phi^*(x) = \frac{dG}{dH}(x) = \frac{1-p}{p} \cdot \frac{\eta(x)}{1-\eta(x)}, \tag{1.14}$$

est *uniformément le plus puissant* parmi les tests sans biais, ce qui implique que sa courbe COR(Φ^*, .) associée domine uniformément toutes les autres. Comme de plus la fonction

$$u \mapsto \frac{1-p}{p} \cdot \frac{u}{1-u}$$

est strictement croissante sur l'intervalle $]0,1[$, le test fondé sur la statistique Φ^* est *équivalent* au test fondé sur la probabilité à posteriori η, au sens où pour un niveau α fixé, ces deux tests ont la même puissance. Les courbes COR(Φ^*, .) et COR* sont donc confondues. Enfin, d'après la Proposition 2, la courbe COR est invariante par transformée strictement croissante. On en déduit donc que la courbe COR* associée à la probabilité à posteriori η est associée de manière équivalente à toute fonction de score optimale $s^* \in \mathcal{S}^*$. On résume ces résultats dans la Proposition 3 ci-dessous.

Proposition 3 *(Courbe optimale* COR**)*
Soit COR* *la courbe* COR *associée à la fonction de régression et l'ensemble des fonctions de score optimales*

$$\mathcal{S}^* = \{s^* = T \circ \eta \mid T : [0,1] \mapsto \mathbb{R},\ \textit{strictement croissante}\}.$$

On a :

$$\forall\ s^* \in \mathcal{S}^*,\ \forall\ \alpha \in]0,1[\,,\ \mathrm{COR}(\mathrm{s}^*, \alpha) = \mathrm{COR}^*(\alpha). \tag{1.15}$$

De plus, toute fonction $s^* \in \mathcal{S}^*$ induit sur $\mathcal{X}$ une relation d'ordre optimale au sens de la courbe COR*. En effet, pour toute fonction de score $s \in \mathcal{S}$, on a :*

$$\forall\ \alpha \in]0,1[\,,\ \mathrm{COR}^*(\alpha) \geq \mathrm{COR}(\mathrm{s}, \alpha). \tag{1.16}$$

De plus, d'après la Proposition 2, on sait que, sous certaines conditions, la courbe COR* est différentiable. Nous explicitons dans la Proposition 4 suivante, les dérivées premières et secondes de la courbe optimale COR*, dont nous aurons besoin dans la suite pour justifier certains résultats théoriques.

Proposition 4 *(Dérivabilité de la courbe* COR* *[Clémençon & Vayatis 2009d])*
Soit $\mathcal{Q}^*(\alpha)$, $\alpha \in [0,1]$, le quantile d'ordre $(1-\alpha)$ de la distribution $H^*(dx)$. On suppose que G^* et H^* sont différentiables, que

$$\mathcal{Q}^*(0) = \lim_{\alpha \to 0} \mathcal{Q}^*(\alpha) < 1,$$

et qu'il existe une constante $c > 0$ telle que $H^{*\prime}(u) \geq c$ pour tout u appartenant au support de $H^{*\prime}(X)$. Sous ces hypothèses, la courbe COR* *optimale est deux fois dérivable sur* $[0,1]$,

de dérivées bornées telles que $\forall \alpha \in [0,1]$,

$$\frac{d}{d\alpha}\mathrm{COR}^*(\alpha) = \frac{1-p}{p} \cdot \frac{\mathcal{Q}^*(\alpha)}{1-\mathcal{Q}^*(\alpha)} , \tag{1.17}$$

$$\frac{d^2}{d\alpha^2}\mathrm{COR}^*(\alpha) = \frac{1-p}{p} \cdot \frac{\mathcal{Q}^{*\prime}(\alpha)}{(1-\mathcal{Q}^*(\alpha))^2} , \tag{1.18}$$

où $\mathcal{Q}^{*\prime}(\alpha) = -1/H^{*\prime}(\mathcal{Q}^*(\alpha))$.

Le problème de l'estimation statistique de ce critère fonctionnel a fait l'objet de nombreux travaux de recherches. En particulier, l'estimation de ce critère d'un point de vue paramétrique et non-paramétrique est étudiée dans [Hsieh & Turnbull 1996]. On peut aussi citer les contributions de [Macskassy & Provost 2004] et [Horvath *et al.* 2008], qui abordent la question de la construction de bandes de confiance pour la courbe COR, celles de [Macskassy *et al.* 2005] et [Bertail *et al.* 2008], qui proposent une procédure d'estimation *bootstrap* ou encore les travaux présentés dans [Mozer *et al.* 2001], qui considèrent le problème de l'estimation d'un point spécifique de la courbe COR.

1.2.1.5 Courbes COR et RP

Outre la courbe COR, il existe un autre critère fonctionnel permettant d'évaluer et de comparer les performances de règles de score définies sur un espace $\mathcal{X}$: la courbe Rappel-Précision (RP), qui associe le *rappel* -taux de vrais positifs- d'un classifieur binaire à sa précision Prec_C. En se basant sur la matrice de confusion du classifieur C, donnée dans la Figure 1.2, on définit la précision comme suit :

$$\mathrm{Prec}_C = \mathbb{P}\{Y = +1 \mid C(X) = +1\} \approx \frac{\mathrm{VP}}{\mathrm{VP}+\mathrm{FP}}, \tag{1.19}$$

$$= \frac{p \cdot \mathrm{TVP}_C}{p \cdot \mathrm{TVP}_C + (1-\mathrm{p}) \cdot \mathrm{TFP}_C}, \text{ où } p = \mathbb{P}\{Y = +1\}. \tag{1.20}$$

On préfère utiliser ce critère quand la représentation des deux classes est très déséquilibrée. En effet, en considérant la précision plutôt que le taux de faux positifs, la courbe RP tient compte de la présence massive d'observations négatives dans l'échantillon. Ce critère permet ainsi de visualiser des différences entre des règles de score que l'on ne pourrait voir en considérant leurs courbes COR.

Ces deux critères fonctionnels sont clairement liés, puisque l'on peut exprimer la précision en fonction du taux de faux positifs (cf (1.19)). En particulier, dans [Davis & Goadrich 2006], les auteurs soulignent qu'il existe une correspondance *points par points* entre ces deux courbes, au sens où les points qu'elles contiennent sont définis par les mêmes matrices de confusion. Cette *bijection* implique notamment qu'une courbe COR^* domine uniformément toutes les autres si et seulement si sa courbe RP^* correspondante domine uniformément toutes les courbes RP. Notons que le problème de l'estimation statistique de la courbe RP a été abordé récemment dans [Clémençon & Vayatis 2009b].

Malgré leurs caractéristiques, qui feraient de ces courbes des critères parfaitement adaptés pour l'apprentissage de fonctions de score, celles-ci sont surtout utilisées à posteriori pour valider les performances de règles d'ordonnancement estimées par ailleurs. En effet, leur nature fonctionnelle les rend difficiles à optimiser directement et, en pratique,

la construction de fonctions de score repose le plus souvent sur l'optimisation de critères scalaires, comme l'Aire Sous la Courbe COR (ASC) introduit ci-dessous.

1.2.2 Aire sous la courbe COR

Le critère scalaire défini par l'aire sou la courbe COR, que l'on notera ASC, est un des critères scalaires les plus utilisés dans le cadre de la problématique d'ordonnancement binaire.

Définition 5 *(Aire sous la courbe* COR*)*
Soit $s \in \mathcal{S}$ une fonction de score définie sur $\mathcal{X}$ et COR(s, .) *sa courbe* COR*. Le critère* ASC(s) *associé à la fonction s est donné par*

$$\mathrm{ASC(s)} = \int_{\alpha=0}^{1} \mathrm{COR(s}, \alpha)\mathrm{d}\alpha. \tag{1.21}$$

Une des particularités de ce critère scalaire est qu'il admet une interprétation probabiliste (voir [Hanley & McNeil 1982]). En effet, l'ASC(s) associée à une fonction de score $s \in \mathcal{S}$ s'exprime comme la probabilité qu'une paire d'observations $(X, X') \in \mathcal{X}^2$ soit correctement ordonnée par s. La Proposition 5 suivante donne la formulation probabiliste de l'ASC d'une fonction $s \in \mathcal{S}$ pour la convention *Conv. 2* retenue pour la courbe COR (cf Remarque 2).

Proposition 5 *(Inteprétation probabiliste de l'*ASC *[Clémençon et al. 2008])*
Soit une fonction de score $s \in \mathcal{S}$ et deux couples i.i.d. (X, Y) et (X', Y') de $\mathcal{X} \times \{-1, +1\}$. L'aire sous la courbe COR(s, .) *peut s'écrire sous la forme :*

$$\begin{aligned} \mathrm{ASC(s)} &= \mathbb{P}\{s(X) > s(X') \mid (Y, Y') = (+1, -1)\} \\ &+ \frac{1}{2} \cdot \mathbb{P}\{s(X) = s(X') \mid (Y, Y') = (+1, -1)\}. \end{aligned} \tag{1.22}$$

Remarque 3 *(Convention* COR*)*
*La convention Conv. 1, définie dans la Remarque 2, est plus souvent utilisée en pratique. Elle implique une formulation différente de l'*ASC *associée à une fonction de score $s \in \mathcal{S}$, qui est alors donnée par*

$$\mathrm{ASC(s)} = \mathbb{P}\{\mathrm{s(X)} \geq \mathrm{s(X')} \mid \mathrm{(Y, Y')} = (+1, -1)\}. \tag{1.23}$$

1.2.2.1 Aire sous la courbe COR empirique

Considérons un échantillon $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$ de $n \geq 1$ copies *i.i.d.* du couple de variables aléatoires $(X, Y) \in \mathcal{X} \times \{-1, +1\}$. Il découle naturellement de la Proposition 5, qu'une contrepartie empirique de l'ASC(s) associée à une fonction de score $s \in \mathcal{S}$, est donnée par le taux de *paires concordantes* selon la fonction s, calculé sur l'échantillon $\mathcal{D}_n$, que l'on note

$$\widehat{\mathrm{ASC}}(s) = \frac{1}{n_+ n_-} \sum_{i\,:\,Y_i=+1} \sum_{j\,:\,Y_j=-1} \left[\mathbb{I}\{s(X_i) > s(X_j)\} + \frac{1}{2}\mathbb{I}\{s(X_i) = s(X_j)\} \right], \tag{1.24}$$

où, $n_+ = \sum_{i\,:\,Y_i=+1} \mathbb{I}\{Y_i = +1\}$ et $n_- = n - n_+$. De plus, il est intéressant de remarquer que la quantité (3.1) correspond exactement à la statistique de test de Wilcoxon-Mann-Whitney associée à la variable aléatoire $s(X)$, pour $s \in \mathcal{S}$ et $x \in \mathcal{X}$ (cf Chapitre 6). On

rappelle en effet que, pour une variable aléatoire réelle $Z \in \mathbb{R}$ et deux échantillons $\mathcal{D}_n^+$ et $\mathcal{D}_n^-$, constitués respectivement de n_+ et n_- copies *i.i.d.* de Z, tels qu'à tout $Z_i^+ \in \mathcal{D}_n^+$ est associé un label positif et à tout $Z_j^- \in \mathcal{D}_n^-$ est associé un label négatif, la statistique de test de Mann-Whitney s'écrit

$$\widehat{U}_{n_+,n_-} = \frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \left[\mathbb{I}\{Z_i^+ > Z_j^-\} + \frac{1}{2} \cdot \mathbb{I}\{Z_i^+ = Z_j^-\} \right].$$

Une autre propriété intéressante de ce critère, introduite dans [Clémençon *et al.* 2005b], réside dans le fait qu'il peut s'écrire sous la forme d'une U-statistique, dont nous rappelons la définition ci-dessous.

Définition 6 *(U-statistique)*
Soient $(X_1, \cdots, X_n)$, n copies i.i.d. de la variable aléatoire $X \in \mathcal{X}$. La statistique U_n définie, pour tout $i \in \{1, \cdots, n\}$ et $j \in \{1, \cdots, n\}$, par

$$U_n = \frac{1}{n(n-1)} \sum_{i \neq j} q(X_i, X_j),$$

avec $q : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ un noyau symétrique, est une U-statistique d'ordre 2.

En effet, en partant de la formulation probabiliste (1.22) de l'ASC(s), associée à une fonction de score $s \in \mathcal{S}$, on peut écrire pour tout (X, Y) et (X', Y') de $\mathcal{X} \times \{-1, +1\}$:

$$\begin{aligned} \text{ASC(s)} &= 1 - \frac{1}{2p(1-p)} \mathbb{P}\{(s(X) - s(X^{'})) \cdot (Y - Y^{'}) < 0\} \\ &- \frac{1}{2} \mathbb{P}\{s(X) = s(X^{'} \mid (Y, Y^{'}) = (+1, -1)\}, \\ &= 1 - \frac{\mathcal{L}(s)}{2p(1-p)} - \frac{1}{2} \mathbb{P}\{s(X) = s(X^{'} \mid (Y, Y^{'}) = (+1, -1)\}. \end{aligned}$$

La quantité (abusivement notée) $\mathcal{L}(s)$ peut être interprétée comme le *risque d'ordonnancement*

$$\mathcal{L}(r) = \mathbb{P}\{Z \cdot r(X, X') < 0\}, \tag{1.25}$$

associé à un classifieur $r(X, X') = 2 \cdot \mathbb{I}\{s(X) \geq s(X')\} - 1 + \mathbb{I}\{s(X) = s(X')\}$ prédisant l'étiquette

$$Z = \frac{Y' - Y}{2} \in \{-1, 0, +1\}, \tag{1.26}$$

à toute paire (X, X') d'observations de l'espace $\mathcal{X}$. Or, la U-statistique d'ordre 2, donnée par

$$\mathcal{L}_n(s) = \frac{1}{n(n-1)} \sum_{i \neq j} \mathbb{I}\{s(X_i) - s(X_j)(Y_i - Y_j) < 0\}, \tag{1.27}$$

$$\tag{1.28}$$

pour tout (X_i, Y_i) et (X_j, Y_j) de $\mathcal{X} \times \{1, +1\}$, est un estimateur empirique sans biais naturel du risque $\mathcal{L}(s)$. Par analogie, on peut donc définir un estimateur de l'ASC(s) à partir de

la U-statistique suivante :

$$U_n(s) = \frac{1}{n(n-1)} \sum_{i \neq j} \mathbb{I}\{s(X_i) - s(X_j)(Y_i - Y_j) > 0\}, \tag{1.29}$$

$$+ \frac{1}{2n(n-1)} \sum_{i \neq j} \mathbb{I}\{s(X_i) = s(X_j), Y_i \neq Y_j\}. \tag{1.30}$$

Cette nouvelle formulation de l'ASC empirique est particulièrement intéressante. En effet, les U-statistiques ont été largement étudiées dans la littérature et les nombreux résultats obtenus, que l'on peut trouver dans [Serfling 1980] et [Peña & Giné 1999] par exemple, facilitent grandement l'étude théorique de la M-estimation du critère ASC. En particulier, la théorie des U-processus permet de contrôler la déviation entre l'ASC* optimale et l'ASC(s) associée à une fonction de score $s \in \mathcal{S}$. Ainsi, des résultats théoriques de convergence, sur l'ensemble $\mathcal{S}$ au sens de l'ASC, ont pu être établis dans [Clémençon *et al.* 2005b] et [Clémençon *et al.* 2008]. De la même façon, les résultats théoriques de convergence présentés dans les différents chapitres de ce manuscrit, reposent sur cette formulation du critère ASC.

Remarque 4 *(Classification binaire sur des paires)*
*L'expression (1.25), donnant l'*ASC *d'une fonction de score $s \in \mathcal{S}$ en fonction du risque d'ordonnancement $\mathcal{L}(s)$, permet d'interpréter l'ordonnancement binaire comme un problème de classification sur les paires d'observations de $\mathcal{X}$ (voir [Clémençon et al. 2005a]). En effet, il apparaît clairement que minimiser le risque $\mathcal{L}(r)$ équivaut à maximiser l'*ASC *associée à la fonction s. Or, l'étiquette Z, définie par (1.26), induit un relation d'ordre sur les données de $\mathcal{X}$, au sens où une observation X est mieux classée qu'une observation X' si et seulement si $Z > 0$.*

1.2.2.2 Aire sous la courbe optimale COR*

Il apparaît clairement que la classe $\mathcal{S}^*$ de fonctions de score optimales correspond à l'ensemble des fonctions de score d'ASC maximale, étant donné que leur courbe COR* associée domine toutes les autres par rapport à la norme $||.||_\infty$. On notera :

$$\forall s \in \mathcal{S}^*, \ \ \mathrm{ASC}^* = \mathrm{ASC(s)}.$$

La Proposition 6 permet d'établir un lien direct entre l'ASC* optimale et la dispersion de la probabilité à posteriori η.

Proposition 6 *(*ASC *optimale [Clémençon et al. 2008])*
*Supposons que la distribution de $\eta(X)$ soit continue, l'*ASC *maximale dépend de la dispersion de $\eta(X)$:*

$$\mathrm{ASC}^* = \frac{1}{2} + \frac{\mathbb{E}[|\eta(\mathrm{X}) - \eta(\mathrm{X}')|]}{4\mathrm{p}(1-\mathrm{p})}, \tag{1.31}$$

où X' est une copie i.i.d. de la variable aléatoire X et la quantité $\mathbb{E}[|\eta(X) - \eta(X')|]$ est l'écart moyen de Gini (Gini Mean Difference) de $\eta(X)$.

Cette proposition montre que plus la difficulté du problème augmente, *i.e.* plus la probabilité η est *concentrée*, plus l'ASC* optimale se rapproche de la valeur $1/2$. En d'autres termes, plus la courbe COR* optimale est proche de la diagonale principale du plan $(\mathrm{TFP}, \mathrm{TVP})$, plus il est difficile d'ordonner les observations de $\mathcal{X}$.

Remarque 5 *(Ordonnancement versus classification)*
Il est possible d'établir un parallèle avec un résultat analogue bien connu dans le contexte de la classification binaire. Considérons le classifieur de Bayès g^*, prédisant $g^*(X) = +1$ si $\eta(X) > 1/2$ et $g^*(X) = -1$ sinon, optimal au sens où il minimise le risque $\mathcal{L}(g) = \mathbb{P}\{g(X) \neq Y\}$ sur l'ensemble $\mathcal{G}$ des classifieurs binaires définis sur $\mathbb{R}$: $\mathcal{L}(g^*) = \mathcal{L}^* = \min_{g \in \mathcal{G}} \mathcal{L}(g)$. Pour tout couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$, on peut écrire que :

$$\mathcal{L}^* = \mathbb{E}\left[|2\eta(X) - 1|\right]. \tag{1.32}$$

*La comparaison de ce résultat avec l'Identité (1.31) met en évidence la différence entre les problématiques d'ordonnancement et de classification binaire. En effet, l'Egalité (1.32) montre que la difficulté du problème de la classification binaire réside dans la concentration de la probabilité à posteriori autour de la valeur $1/2$, alors que dans le cadre de l'ordonnancement binaire, la difficulté du problème dépend de la concentration de η en tout point de $\mathcal{X}$. Les problèmes de la minimisation du risque d'un classifieur binaire $\mathcal{L}(g)$ et de la maximisation de l'*ASC(s) *associée à une fonction de score $s \in \mathcal{S}$ ne sont donc pas équivalents. En particulier, il a été montré dans [Cortes & Mohri 2004] et [Yan et al. 2003], que l'utilisation du critère* ASC *conduisait à des règles de score plus performantes que celles obtenues par minimisation du critère traditionnel* MSE *(Mean Squared Error).*

1.2.2.3 Limites du critère ASC

Contrairement à la courbe COR (cf Partie 1.2.1.3), l'ASC définit un ordre total sur l'ensemble $\mathcal{S}$ des fonctions de score. Cependant, il est tout à fait possible d'avoir deux courbes $\mathrm{COR}(s_1)$ et $\mathrm{COR}(s_2)$ différentes mais de même ASC, telles que la fonction de score s_1 soit plus performante que s_2 pour le classement des meilleures observations (cf Figure 1.6). Or, dans de nombreuses applications, comme la mise au point des moteurs de recherche sur internet, il est préférable de se concentrer sur les observations les plus probablement positives et dans ce cas précis, le critère ASC ne garantit pas de sélectionner la fonction de score la plus pertinente.

Une solution à ce problème est de se focaliser sur le sous-ensemble contenant les meilleures observations. Une approche naïve consisterait à optimiser l'ASC restreinte à ce sous-ensemble de données. Cependant, les considérations géométriques présentées dans la contribution [Clémençon & Vayatis 2007], montrent que ce critère n'est pas pertinent. C'est pourquoi, les auteurs de cette contribution proposent de considérer un autre critère d'ASC tronqué, garantissant de sélectionner la fonction de score la plus performante sur une portion fixée des meilleures observations de l'échantillon. Nous reviendrons plus en détail sur ce critère au Chapitre 5.

Une autre possibilité consiste à optimiser des critères comme la précision moyenne MAP (voir [Yue & Finley 2007]) ou la mesure NDCG (voir [Järvelin & Kekäläinen 2000]), introduits dans le contexte de la recherche automatique de documents (*Information Retrieval*),

qui permettent eux-aussi de se concentrer sur les meilleures observations. Enfin, on peut également citer la contribution de [Rudin 2006], proposant la méthode *P-Norm Push*, dans laquelle l'apprentissage d'une règle de score repose sur l'optimisation d'un critère empirique convexe, privilégiant l'ordonnancement des meilleures observations de l'échantillon.

Néanmoins, le critère de l'ASC a fait l'objet de nombreux travaux et reste un critère de référence. Comme nous l'avons déjà cité, la question de l'apprentissage statistique de ce critère a notamment été abordé dans [Clémençon *et al.* 2005b] et [Clémençon *et al.* 2008]. Citons aussi les contributions de [Agarwal *et al.* 2005] et [Usunier & ans P. Gallinari 2005], introduisant différents types de bornes pour ce critère, ou encore les travaux présentés dans [Cortes & Mohri 2005], relatifs à la construction d'intervalles de confiance pour l'ASC. Enfin, comme nous allons le voir dans la Partie 1.3 suivante, de nombreux travaux traitent de la question de l'optimisation de ce critère du point de vue de l'apprentissage automatique, et c'est sans doute la raison pour laquelle ce critère est si populaire.

1.3 M-estimation d'une fonction de score

Estimer l'ordre induit sur l'espace d'entrée $\mathcal{X}$ par la fonction de régression consiste à trouver une fonction de score $s \in \mathcal{S}$, dont la courbe COR(s, .) soit la plus *proche* possible de la courbe optimale COR*. En notant $d(\mathrm{COR}_1, \mathrm{COR}_2)$ une mesure de la distance entre deux courbes COR, le problème de la M-estimation d'une fonction de score se formalise comme suit :

$$\forall\, x \in \mathcal{X},\ \hat{s}(x) = \underset{s \in \mathcal{S}}{\arg\min}\ d(\mathrm{COR}^*, \mathrm{COR}(\mathrm{s}, .)). \tag{1.33}$$

Naturellement, on peut considérer diverses métriques sur l'espace des courbes COR. En pratique, la plupart des méthodes d'apprentissage pour l'ordonnancement binaire reposent sur l'optimisation de l'ASC empirique ou, de manière équivalente, du risque empirique d'ordonnancement défini par (1.25) (voir par exemple [Freund *et al.* 2003], [Joachims 2002b], [Cortes & Mohri 2004], [Rakotomamonjy 2004] ou encore [Yan *et al.* 2003]). Or ce critère est lié à une mesure de la distance entre courbes COR, au sens de la norme $\mathcal{L}_1$:

$$d_1(s^*, s) = \int_0^1 (\mathrm{COR}^*(\alpha) - \mathrm{COR}(\mathrm{s}, \alpha))\mathrm{d}\alpha. \tag{1.34}$$

Avec l'utilisation de cette métrique, le problème de scoring devient :

$$\begin{aligned} s^* \ &\in\ \underset{s \in \mathcal{S}}{\arg\min}\ d_1(s^*, s) \\ &\in\ \underset{s \in \mathcal{S}}{\arg\min}\ \{\int_0^1 (\mathrm{COR}^*(\alpha) - \mathrm{COR}(\mathrm{s}, \alpha))\mathrm{d}\alpha\} \\ &\in\ \underset{s \in \mathcal{S}}{\arg\min}\ \{\mathrm{ASC}^* - \mathrm{ASC}(\mathrm{s})\} \\ &\in\ \underset{s \in \mathcal{S}}{\arg\max}\ \{\mathrm{ASC}(\mathrm{s})\}. \end{aligned} \tag{1.35}$$

Comme nous l'avons indiqué dans la Partie 1.2.2.3, l'optimisation de ce critère permet d'identifier une fonction de score d'ASC empirique maximale parmi d'autres mais ne garantit pas qu'il s'agisse de la *meilleure*, au sens de l'ordonnancement des meilleures observations. Pour cette raison et étant donné la nature de l'ordre induit sur $\mathcal{S}$ par les courbes

COR, caractérisé par la Relation (1.16) (Proposition 3), il nous semble plus pertinent de raisonner en termes de norme supérieure sur l'ensemble des courbes COR. En effet, seule cette approche permet de discriminer des fonctions de score de courbes COR différentes mais de même ASC. En outre, la convergence en norme $\mathcal{L}_\infty$ garantit la convergence en norme $\mathcal{L}_1$.

Soient deux fonctions de score $s \in \mathcal{S}$ et $s^* \in \mathcal{S}^*$, la norme $\mathcal{L}_\infty$ sur l'espace des courbes COR permet de définir la distance

$$d_\infty(s^*, s) = \sup_{\alpha \in]0,1[} (\mathrm{COR}^*(\alpha) - \mathrm{COR}(\mathrm{s}, \alpha)). \tag{1.36}$$

Avec ces notations, le problème d'ordonnancement binaire devient :

$$s^* = \arg\min_{s \in \mathcal{S}} \; d_\infty(s^*, s) = \arg\min_{s \in \mathcal{S}} \quad \{ \sup_{\alpha \in (0,1)} (\mathrm{COR}^*(\alpha) - \mathrm{COR}(\mathrm{s}, \alpha))\}. \tag{1.37}$$

Malheureusement, contrairement au problème défini par (1.35), cette formulation n'admet pas de contrepartie empirique. Toutefois, le résultat présenté ci-dessous confirme la bonne adéquation de ce critère avec notre problématique, en établissant un lien entre le problème d'ordonnancement binaire défini par (1.37) et l'approximation point par point de la courbe optimale COR*.

Proposition 7 *([Clémençon & Vayatis 2009d]) Soit $s \in \mathcal{S}$ une fonction de score quelconque. On note R^*_α et $R_{s,\alpha}$ les ensembles de niveaux de la fonction de régression et de s définis comme suit :*

$$R^*_\alpha \;\; = \{x \in \mathcal{X} \mid \eta(x) > \mathcal{Q}^*(\alpha)\} \;\; et \;\; R_{s,\alpha} \;\; = \{x \in \mathcal{X} \mid s(x) > \mathcal{Q}_s(\alpha)\},$$

où $\mathcal{Q}^*(\alpha)$ et $\mathcal{Q}_s(\alpha)$ sont les quantiles d'ordre $(1-\alpha)$, $\alpha \in]0,1[$, des distributions conditionnelles des variables aléatoires $\eta(X)$ et $s(X)$ respectivement, sachant l'événement $\{Y = -1\}$. Si l'on suppose que :

(i) les distributions conditionnelles $G_s(dx)$ et $H_s(dx)$ sont continues au point $\mathcal{Q}_s(\alpha)$,

(ii) les distributions conditionnelles $G^*(dx)$ et $H^*(dx)$ sont continues au point $\mathcal{Q}^*(\alpha)$,

(iii) $\mathcal{Q}^*(\alpha) < 1$,

alors pour toute fonction de score $s \in \mathcal{S}$ on a :

$$\begin{aligned} \mathrm{COR}^*(\alpha) - \mathrm{COR}_\mathrm{s}(\alpha) \;\; &= \;\; \frac{\mathbb{E}\left[|\eta(X) - \mathcal{Q}^*(\alpha)| \cdot \mathbb{I}\{X \in R^*_\alpha \Delta R_{s,\alpha}\}\right]}{p(1 - \mathcal{Q}^*(\alpha))} \\ &+ \;\; \frac{1-p}{p} \frac{\mathcal{Q}^*(\alpha)}{1 - \mathcal{Q}^*(\alpha)} \left(\alpha - 1 + H_s(\mathcal{Q}_s(\alpha))\right), \end{aligned} \tag{1.38}$$

où Δ représente la différence symétrique entre ensembles :

$$R^*_\alpha \Delta R_{s,\alpha} = \mathbb{I}\{X \in R^*_\alpha\} - \mathbb{I}\{X \in R_{s,\alpha}\}.$$

Remarque 6 *Une conséquence de la Proposition 7 est que la statistique uni-dimensionnelle η suffit à caractériser la courbe optimale* COR*. *En d'autres termes, la courbe* COR *n'est pas altérée par la projection des observations de l'espace $\mathcal{X}$ sur le segment $[0,1]$ par le biais de la fonction de régression (cf Corollaire 5 dans [Clémençon & Vayatis 2009d]).*

Rappelons que, d'après la représentation (1.3) d'une fonction de score optimale donnée dans la Proposition 1, le problème d'ordonnancement binaire revient à estimer la collection $\mathcal{C}^* = \{\{x \in \mathcal{X} \mid \eta(x) \geq u\},\ u \in]0,1[\}$ des ensembles de niveaux de la fonction de régression. Or, l'Identité (1.38) montre que la différence point par point entre une courbe COR(s), associée à une fonction de score s, et la courbe COR* peut-être vue comme l'erreur d'estimation de l'ensemble de niveau R^*_α par $R_{s,\alpha}$. Il apparaît donc clairement qu'estimer la courbe COR* optimale revient à estimer la collection $\mathcal{C}^*$ des ensembles de niveaux de η.

L'estimation de ce critère fonctionnel est un problème difficile. Une manière classique d'aborder ce problème est de le discrétiser en considérant la question de l'estimation de la courbe COR du point de vue de la théorie de l'approximation. C'est sur cette approche que reposent les travaux proposés dans [Clémençon & Vayatis 2008c] et [Clémençon & Vayatis 2009d]. En effet, l'algorithme TreeRank y est présenté comme une version empirique d'une procédure d'approximation de la courbe optimale COR* par une fonction affine par morceaux. De plus, les auteurs parviennent à établir un lien entre cette approche et l'optimisation récursive de l'ASC empirique, calculée localement sur des sous-ensembles de $\mathcal{X}$ choisis de manière adaptative. Ainsi, en procédant à la maximisation récursive de l'ASC empirique, la méthode de scoring TreeRank considère le problème de la M-estimation de fonctions de score optimales au sens de l'ASC et de la courbe COR. La Partie 1.4 suivante est consacrée à la description du principe de cette procédure d'approximation de la courbe COR* et à la présentation du schéma d'approximation sous-jacent à l'heuristique TreeRank proposée dans [Clémençon & Vayatis 2008c] et [Clémençon & Vayatis 2009d].

1.4 Approximation de la courbe COR* par une fonction affine par morceaux

Le premier objectif de cette partie est d'introduire le principe de l'approximation de la courbe COR* par une fonction affine par morceaux. Pour cela, nous décrivons une première procédure, dans laquelle un nouveau point de la courbe COR* est estimé à chaque itération. Puis, dans un deuxième temps, nous détaillons le schéma d'approximation arborescent, sous-jacent à l'algorithme TreeRank, qui permet à chaque itération d'intercaler un nouveau point dans chacun des segments de l'estimateur affine par morceaux de la courbe COR*.

Mais avant cela, il nous faut introduire une nouvelle classe de fonctions, intrinsèquement liées à cette procédure d'approximation : les fonctions de score constantes par morceaux. Rappelons en effet que l'estimation de la courbe COR* optimale par une fonction affine par morceaux permet d'estimer une collection finie

$$\mathcal{C}^*_N = \{\{x \in \mathcal{X} \mid \eta(x) > u_k\},\ u_k \in]0,1[\}_{1 \leq k \leq N}$$

constituée de N ensembles de niveaux de la fonction de régression. Or, cette collection définit une partition $\mathcal{P}$ de l'espace $\mathcal{X}$ finie et ordonnée, constituée de N cellules disjointes de la forme $\{x \in \mathcal{X} \mid u_{k+1} < \eta(x) \leq u_k\}$, où pour tout $k \in \{1, \cdots, N\}$, $u_k \in]0,1[$ et $u_0 = 0$ par convention, à laquelle on peut naturellement associer une fonction de score constante par morceaux, attribuant un score différent à chaque cellule de $\mathcal{P}$. Aussi, on

verra que les procédures introduites dans cette partie permettent de considérer simultanément les problèmes de l'approximation de la courbe COR* optimale et de la construction de fonctions de score constantes par morceaux *quasi-optimales*, *i.e.* d'ASC et de courbe COR asymptotiquement optimales.

1.4.1 Fonction de score constante par morceaux

Une fonction de score $s_N : \mathcal{X} \to \mathbb{R}$ constante par morceaux, prenant N valeurs distinctes, définit une partition disjointe $\mathcal{P}$ sur l'espace $\mathcal{X}$, constituée de N cellules non vides $(C_j)_{1\leq j\leq N}$ telles que $\forall\, j,\, C_j \subset \mathcal{X}$, $\cup_j C_j = \mathcal{X}$ et $C_i \cap C_j = \emptyset$ pour tout $i \neq j$. Pour tout $j \in \{1, \cdots, N\}$, toutes les observations de $\mathcal{X}$ contenues dans une même cellule C_j ont le même score et s_N attribue un score différent à chacune des cellules de $\mathcal{P}$, permettant ainsi de les ordonner et d'induire un pré-ordre sur $\mathcal{X}$.

En particulier, une fonction de score constante par morceaux quasi-optimale induit un pré-ordre sur l'espace d'entrée $\mathcal{X}$ par le biais d'une partition $\mathcal{P}^*$ finie, disjointe et ordonnée. En effet en remplaçant, en première approximation, la notion d'espérance par sa contrepartie empirique dans la représentation (1.3) d'une fonction de score optimale donnée dans la Proposition 1, on obtient une fonction constante par morceaux s_N^*, qui s'exprime sous la forme d'une somme finie d'indicatrices du type $\mathbb{I}\{\eta(x) > V_j\}$, pour $j \in \{1, \cdots, N\}$, où les $(V_j)_j$ sont les observations d'une variable aléatoire $V \in]0, 1[$ continue. Ainsi, la fonction s_N^* définit une partition $\mathcal{P}^*$ disjointe et ordonnée sur $\mathcal{X}$, constituée de cellules de la forme $R_j^* = \{x \in \mathcal{X} \mid v_{j+1} < \eta(x) \leq v_j\}$, où pour tout $j \in \{1, \cdots, N\}$, $v_j \in]0, 1[$.

1.4.1.1 $(\mathcal{P}, \sigma)$-représentation

Soit $s_N : \mathcal{X} \to \mathbb{R}$ une fonction de score constante par morceaux prenant $N \geq 1$ valeurs distinctes, $\lambda_1, \cdots, \lambda_N$. Le pré-ordre $\preceq_{s_N}$ induit par s_N sur l'espace $\mathcal{X}$ est entièrement caractérisé par la donnée d'une partition finie $\mathcal{P}$ de $\mathcal{X}$, constituée de N sous-ensembles mesurables non-vides et disjoints $C_1, \ldots, C_N$ de $\mathcal{X}$, et d'une permutation σ de $\{1, \ldots, N\}$, du groupe symétrique $\mathfrak{S}_N$. Avec ces notations, on définit la $(\mathcal{P}, \sigma)$-*représentation* de la fonction s_N comme suit :

$$\forall\, x \in \mathcal{X},\, s_N(x) = \sum_{j=1}^{N} \lambda_j \cdot \mathbb{I}\{x \in C_{\sigma(j)}\}. \tag{1.39}$$

Réciproquement, une partition $\mathcal{P} = \{C_1, \ldots, C_N\}$, constituée de $\#\mathcal{P} = N$ cellules non vides et associée à une permutation $\sigma \in \mathfrak{S}_N$ définit une fonction de score constante par morceaux prenant N valeurs distinctes, que l'on peut écrire sous la forme :

$$\forall\, x \in \mathcal{X},\, s_N(x) = \sum_{j=1}^{N} (N - j + 1) \cdot \mathbb{I}\{x \in C_{\sigma(j)}\}.$$

On notera $\mathcal{S}_{\mathcal{P}}$ l'ensemble des fonctions de score constantes par morceaux associées à un couple $(\mathcal{P}, \sigma)$, où $\mathcal{P}$ est une partition de $\mathcal{X}$ de cardinal $\#\mathcal{P}$ et $\sigma \in \mathfrak{S}_{\#\mathcal{P}}$ une permutation de $\{1, \cdots, \#\mathcal{P}\}$.

D'après la Proposition 2, la courbe COR associée à une fonction de score s_N constante par morceaux, prenant N valeurs distinctes, est affine par morceaux et constituée de N

segments. En posant les notations suivantes pour les taux théoriques de faux et vrais positifs d'un sous-ensemble $C \subset \mathcal{X}$ mesurable :

$$\alpha(C) = \mathbb{P}\{X \in C \mid Y = -1\}, \tag{1.40}$$
$$\beta(C) = \mathbb{P}\{X \in C \mid Y = +1\}, \tag{1.41}$$

la proposition suivante définit de manière explicite la courbe $\mathrm{COR}(\mathrm{s_N})$ et l'$\mathrm{ASC}_{\mathrm{s_N}}$ associée à une fonction de score constante par morceaux s_N admettant une $(\mathcal{P}, \sigma)$-représentation.

Proposition 8 *(Courbe* COR *et* ASC *d'une fonction de score constante par morceaux [Clémençon et al. 2010])*
Soit s_N une fonction de score constante par morceaux admettant la $(\mathcal{P}, \sigma)$-représentation suivante :

$$\forall\, x \in \mathcal{X},\ s_N(x) = \sum_{j=1}^{N} \lambda_j \cdot \mathbb{I}\{x \in C_{\sigma(j)}\}.$$

La courbe $\mathrm{COR}(\mathrm{s_N})$ *associée à s_N est la ligne brisée reliant les noeuds $(\alpha_j(s_N), \beta_j(s_N))$, avec $0 \leq j \leq N\}$, où $\forall j \in \{1, \ldots, N\}$,*

$$\alpha_j(s_N) = \sum_{l=1}^{j} \alpha(C_{\sigma(l)}) \text{ et } \beta_j(s_N) = \sum_{l=1}^{j} \beta(C_{\sigma(l)}),$$

avec $\alpha_0(s_N) = \beta_0(s_N) = 0$ par convention.
De plus, l'$\mathrm{ASC}(\mathrm{s_N})$ *associée à s_N est donnée par :*

$$\mathrm{ASC}(\mathrm{s_N}) = \frac{1}{2} \sum_{j=0}^{N-1} \left(\alpha_{j+1}(s) - \alpha_j(s)\right) \cdot \left(\beta_{j+1}(s) + \beta_j(s)\right). \tag{1.42}$$

1.4.1.2 *D*- et *I*-représentations

Dans la présentation de la procédure d'approximation de la courbe COR* et de l'algorithme TreeRank, nous aurons recours à deux formes de représentations alternatives d'une fonction de score constante par morceaux, que nous introduisons ici. Premièrement, une fonction de score constante par morceaux s_N à valeurs dans $\{a_1, \ldots, a_N\}$ admet une *D-représentation* (pour *représentation Disjointe*), donnée par :

$$\forall\, x \in \mathcal{X}, \quad s_N(x) = \sum_{j=1}^{N} a_j\ \mathbb{I}\{x \in C_j\}\ , \tag{1.43}$$

où $(a_j)_{j \geq 1}$ est une séquence décroissante, définissant un ordre sur les cellules d'une partition $\mathcal{C}_N = (C_j)_{1 \leq j \leq N}$ de $\mathcal{X}$. Notons que la courbe COR associée à une fonction s_N admettant une D-représentation ne dépend pas des valeurs de la séquence $(a_j)_{j \geq 1}$, mais seulement de leur ordonnancement.

On peut aussi définir une deuxième représentation, à partir d'une *séquence croissante de sous-ensembles* de $\mathcal{X}$, *i.e.* d'une classe finie de sous-ensembles $\mathcal{R}_N = (R_j)_{1 \leq j \leq N}$ telle que $\bigcup_j R_j = \mathcal{X}$ et $R_i \subset R_j$ pour tout $i < j$. On a donc en particulier $R_N = \mathcal{X}$. Avec ces

notations, la *I-représentation* d'une fonction de score constante par morceaux s_N à valeurs dans $\{1, \ldots, N\}$ est donnée par :

$$\forall x \in \mathcal{X} \ , \quad s_N(x) = \sum_{j=1}^{N} \mathbb{I}\{x \in R_j\} \ , \tag{1.44}$$

pour une séquence croissante $\mathcal{R}_N = (R_j)_{1 \leq j \leq N}$ de sous-ensembles de $\mathcal{X}$.

On peut établir un lien évident entre ces deux représentations. En effet, supposons que s_N prenne ses valeurs dans $\{1, \ldots, N\}$ et considérons la séquence $\mathcal{R}_N$ provenant de la I-représentation de s_N. On peut obtenir facilement sa D-représentation en prenant $C_1 = R_1$ et :

$$\forall i > 1, \ C_i = R_i \setminus R_{i-1} \quad \text{et} \quad \forall j \ , \ a_j = N - j + 1.$$

Considérer la I-représentation de s_N permet notamment de simplifier la définition de sa courbe $\mathrm{COR}(\mathrm{s_N})$. En effet, en considérant la I-représentation de la fonction s_N définie par (1.44), la courbe $\mathrm{COR}(\mathrm{s_N})$ devient la ligne brisée reliant les noeuds $\{(\alpha(R_j), \beta(R_j))\}_{0 \leq j \leq N}$, où $R_0 = \emptyset$ par convention.

1.4.1.3 Fonction de score constante par morceaux quasi-optimale

Soit s_N une fonction de score constante par morceaux définissant une partition ordonnée $\mathcal{P}$ sur $\mathcal{X}$, constituée de N cellules non vides. Il est important de remarquer que la courbe $\mathrm{COR}(\mathrm{s_N})$ associée à cette fonction n'est pas nécessairement concave. Cependant, on peut trouver une permutation $\sigma \in \mathfrak{S}_N$ de $\{1, \cdots, N\}$ sur les cellules de la partition $\mathcal{P}$ permettant de rendre la courbe $\mathrm{COR}(\mathrm{s_N})$ concave. En d'autres termes, la permutation σ nous permet de considérer le *meilleur* (au sens de la courbe COR) ordonnancement des cellules de la partition $\mathcal{P}$.

Avant d'énoncer le Théorème 1, établissant les conditions d'optimalité d'une fonction de score constante par morceaux, nous définissons ci-dessous la notion de *sous-partition.*

Définition 7 *(Sous-partition)*
Soient $\mathcal{P}$ et $\mathcal{P}'$ deux partitions de $\mathcal{X}$. La partition $\mathcal{P}'$ est une sous-partition de $\mathcal{P}$, si toute cellule $C' \in \mathcal{P}'$ peut s'écrire comme une union de cellules $C \in \mathcal{P}$. On notera alors : $\mathcal{P}' \subset \mathcal{P}$.

Théorème 1 *(Conditions d'optimalité [Clémençon et al. 2010])*
Soit $\mathcal{P}$ une partition de $\mathcal{X}$ constituée de $N \geq 1$ cellules non vides, $\mathcal{P} = \{C_j\}_{1 \leq j \leq N}$, et la permutation $\sigma^* \in \mathfrak{S}_N$ telle que

$$\frac{\beta(C_{\sigma^*(1)})}{\alpha(C_{\sigma^*(1)})} \geq \cdots \geq \frac{\beta(C_{\sigma^*(N)})}{\alpha(C_{\sigma^*(N)})}. \tag{1.45}$$

*Alors, la fonction de score s_N^* associée au couple $(\mathcal{P}, \sigma^*)$ est d'*ASC *maximale sur l'ensemble $\bigcup_{\mathcal{P}' \subset \mathcal{P}} \mathcal{S}_{\mathcal{P}}$:*

$$\mathrm{ASC}(\mathrm{s_N^*}) = \max_{\mathrm{s} \in \mathcal{S}_{\mathcal{P}}, \ \mathcal{P}' \subset \mathcal{P}} \mathrm{ASC}(\mathrm{s}).$$

Quand les cellules de $\mathcal{P}$ sont équivalentes par rapport au taux de faux positifs, i.e. $\forall j \in \{1, \ldots, N\}$: $\alpha(C_j) = 1/N$, on a aussi

$$\forall \alpha \in [0, 1], \ \mathrm{COR}(\mathrm{s}, \alpha) \leq \mathrm{COR}(\mathrm{s_N^*}, \alpha),$$

pour tout $s \in \mathcal{S}_{\mathcal{P}}$, $\mathcal{P}' \subset \mathcal{P}$. (Ce résultat reste vrai quand les cellules de $\mathcal{P}$ sont équivalentes par rapport au taux de vrais positifs.)

1.4.2 Approximation récursive et adaptative de la courbe COR^*

Nous allons considérer simultanément le problème de l'approximation de la courbe COR^* par une fonction affine par morceaux et celui de la construction d'une fonction de score constante par morceaux asymptotiquement optimale au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$, définies sur l'espace des courbes COR. Comme nous l'avons déjà indiqué, cette procédure est intrinsèquement liée à la construction d'une partition finie et ordonnée $\mathcal{P}$ de l'espace $\mathcal{X}$, permettant d'estimer une collection finie $\mathcal{C}_N^*$ de N ensembles de niveaux de la fonction de régression, de la forme $\{x \in \mathcal{X} \mid \eta(x) > u_j\}$, où pour tout $j \in \{1, \cdots, N\}$, $u_j \in]0,1[$. Avec ces notations, la fonction de score constante par morceaux quasi-optimale s_N^* associée à la partition $\mathcal{P}$ sera de la forme :

$$\forall\, x \in \mathcal{X},\ s_N^*(x) = \sum_{j=1}^{N} \mathbb{I}\{x \in \mathcal{X} \mid \eta(x) > u_j\}. \tag{1.46}$$

Considérons la famille $\mathcal{S}_N$ des fonctions de score constantes par morceaux munies d'une I-représentation basée sur la séquence croissante de sous-ensembles $(R_j)_{j\geq 1}$ de $\mathcal{X}$ de la forme :

$$\forall\, j \geq 1,\ R_j = \{x \in \mathcal{X}\ :\ \eta(x) > u_j\},$$

pour une séquence décroissante positive $(u_j)_{j\geq 1}$, avec $u_1 > 0$. Le principe général du schéma itératif présenté dans [Clémençon & Vayatis 2009d] consiste à considérer, dans un premier temps, la courbe COR_1^* constituée d'un unique segment confondu avec la diagonale principale du plan $(\mathrm{TFP}, \mathrm{TVP})$, puis à optimiser celle-ci en *brisant* ce segment de manière itérative pour intercaler de nouveaux *noeuds*.

En d'autres termes, à chaque itération, la procédure consiste à scinder une cellule de la partition $\mathcal{P}$ définie sur $\mathcal{X}$, afin d'estimer un nouvel ensemble de niveau de la fonction de régression. En ce qui concerne l'apprentissage de la fonction de score s_N^* constante par morceaux associée à la partition $\mathcal{P}$ de $\mathcal{X}$, cette étape élémentaire consiste à modifier la fonction de score courante s_k^*, $k \geq 1$, en ajoutant un sous-ensemble dans la séquence $(R_j)_{j\geq 1}$ choisi parmi la collection des ensembles de niveaux de η.

Une des particularités de cette procédure est que le choix des noeuds, ou de manière équivalente d'un sous ensemble R_j, $j \geq 1$, se fait de manière adaptative : à chaque itération on introduit le *meilleur* noeud possible, au sens de la minimisation des distances d_1 et d_∞ à la courbe COR^* optimale. Pour ce faire, l'insertion d'un nouveau noeud est basée sur la maximisation de l'ASC empirique calculée *localement* sur la cellule de la partition $\mathcal{P}$ que l'on cherche à scinder. On peut montrer notamment qu'un noeud optimal au sens de l'ASC locale l'est aussi au sens de la norme supérieure et que l'estimateur COR_N^* obtenu par cette procédure itérative converge asymptotiquement vers la courbe COR^* optimale au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$.

Le résultat suivant définit la procédure d'approximation d'une fonction de score s^* optimale par une fonction s_N^* constante par morceaux à N éléments.

Définition 8 *(Approximation constante par morceaux de s^*)*
Soit $s_N \in \mathcal{S}_N$, on pose :

$$\epsilon_N = \arg\max_{\epsilon \in \mathcal{C}^*} d_1(s_N, s_N + \mathbb{I}_\epsilon).$$

Alors, l'approximation d'une fonction de score optimale $s^* \in \mathcal{S}^*$ par une fonction constante par morceaux prenant N valeurs distinctes est donnée par la séquence $(s_N)_{N\geq 1}$ de fonctions telles que :

$$\begin{aligned} s_1^* &= \mathbb{I}_{\mathcal{X}}, \\ s_{j+1}^* &= s_j^* + \mathbb{I}_{\epsilon_j}, \ \textit{pour} \ 1 \leq j \leq N-1. \end{aligned}$$

1.4.2.1 Première itération

On initialise la procédure pour $j = 1$ avec la fonction de score :

$$\forall x \in \mathcal{X} \ , \ s_1^*(x) = \mathbb{I}\{x \in \mathcal{X}\} \equiv 1 \ ,$$

qui attribue la même valeur de score -égale à 1- à toutes les observations de $\mathcal{X}$. La courbe COR_1^* associée à s_1^* -représentée en rouge sur le premier graphe de la Figure 1.7- correspond à la diagonale du plan $(\mathrm{TFP}, \mathrm{TVP})$ reliant l'origine $(0;0)$ au point de coordonnées $(1;1)$, associé au prédicteur constant s_1^*.

La première itération de la procédure consiste à ajouter à la fonction de score s_1^* l'indicatrice d'un sous-ensemble de $\mathcal{X}$ de la forme $\{x \in \mathcal{X} \mid \eta(x) \geq t\}$, où $t \in]0,1[$. Afin de trouver la meilleure approximation de la courbe COR^* par une ligne brisée à deux segments, on choisit t de sorte à maximiser l'ASC calculée localement sur le sous-ensemble $\{x \in \mathcal{X} \mid \eta(x) \geq t\}$, qui correspond dans ce cas précis à l'ASC globale. Comme le montre la Proposition 9 suivante, cette étape consiste à estimer l'ensemble de niveau p de la probabilité à posteriori η.

Proposition 9 *(Première itération [Clémençon & Vayatis 2009d])*
Supposons que la courbe COR^* est différentiable et concave. Alors l'approximation locale à la première itération conduit à la fonction de score binaire suivante :

$$\forall x \in \mathcal{X} \ , \ s_2^*(x) = \mathbb{I}\{x \in \mathcal{X}\} + \mathbb{I}\{\eta(x) > t^*\} \ , \tag{1.47}$$

où $t^* = p = \mathbb{P}\{Y = 1\}$. On a de plus :

$$(d\beta/d\alpha)(t^*) = 1. \tag{1.48}$$

D'après la Proposition 9, à la première itération, le noeud optimal correspond au point où la tangente de la courbe COR^* est parallèle à la diagonale du plan $(\mathrm{TFP}, \mathrm{TVP})$, *i.e.* au segment COR_1^*.

La courbe COR_2^* -représentée en rouge sur le second graphe de la Figure 1.7- est la meilleure approximation de la courbe COR^* par une fonction affine par morceaux à deux segments, au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$. En effet, il est facile de voir qu'à la première itération, l'*incrémentation* ϵ_1 de s_1^*, définie par (1.47) est équivalente à l'incrémentation $\tilde{\epsilon}_1$ obtenue en optimisant la norme supérieure.

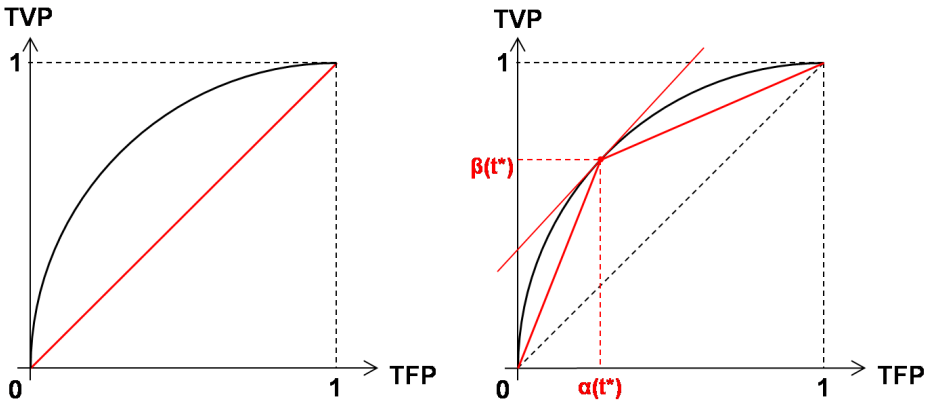


FIGURE 1.7 – Approximation de la courbe COR* : initialisation et première itération.

Proposition 10 *([Clémençon & Vayatis 2009d])*
Soient $\epsilon 1$ et $\tilde{\epsilon}_1$ les incrémentations à la première itération au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$, on a :

$$\epsilon_1 = \arg\max_{\epsilon \in \mathcal{C}^*} d_1(s_1^*, s_1^* + \mathbb{I}_\epsilon), \tag{1.49}$$

$$= \arg\max_{\epsilon \in \mathcal{C}^*} d_\infty(s_1^*, s_1^* + \mathbb{I}_\epsilon) = \tilde{\epsilon}_1. \tag{1.50}$$

1.4.2.2 $N^{\text{ème}}$ itération

Soit $s_N^* \in \mathcal{S}_N^*$ la fonction de score constante par morceaux obtenue après $(N-1)$ itérations. La courbe COR_N^* est la ligne brisée à N segments reliant les points $((\alpha_j, \beta_j))_{0 \leq j \leq N}$ définis par (1.45), où $(\alpha_0, \beta_0) = (0,0)$ et $(\alpha_N, \beta_N) = (1,1)$. A la $N^{\text{ème}}$ itération, on cherche à optimiser l'ASC de COR_N^* en insérant un noeud $(\alpha(t), \beta(t))$, de sorte que $\alpha(t) \in]\alpha_j, \alpha_{j+1}[$ (voir Figure 1.8). On note

$$s_{N+1,t}^{*(j)}(x) = s_N(x) + \mathbb{I}\{\eta(x) > t\},$$

avec $t \in]\mathcal{Q}^*(\alpha_{j+1}), \mathcal{Q}^*(\alpha_j)[$, la fonction de score incrémentée à l'itération N.

On cherche à insérer le noeud $(\alpha(t^*), \beta(t^*))$ qui maximise le gain en ASC, représenté en gris sur la Figure 1.8. On peut écrire l'ASC associée à la fonction $s_{N+1,t}^{*(j)}$ comme la somme d'une constante c_j et d'un terme explicitant la *portion* d'ASC que l'on cherche à maximiser. En effet, d'après la formulation (1.42) établie dans la Proposition 8, on a :

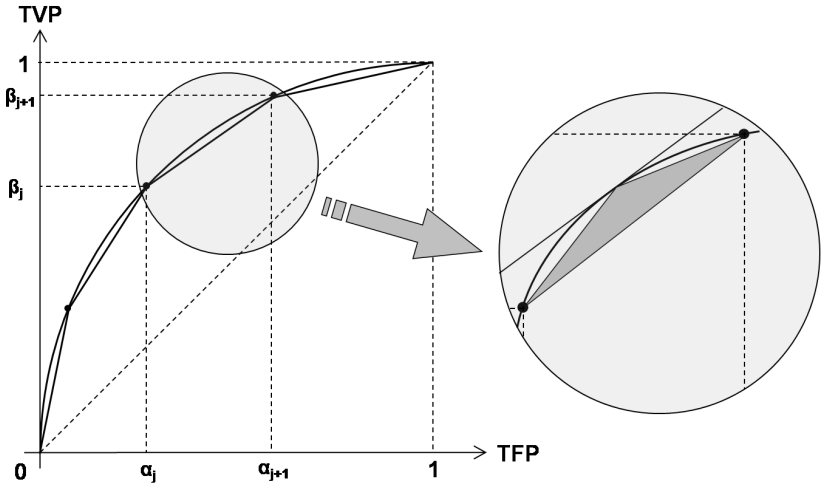


FIGURE 1.8 – $N^{\text{ème}}$ itération de la procédure d'approximation de la courbe COR*.

$$
\begin{aligned}
A^*_{N+1}(t) &= \mathrm{ASC}(\mathrm{s}^{*(\mathrm{j})}_{\mathrm{N+1,t}}) = \frac{1}{2}\cdot\sum_{\mathrm{k}=0}^{\mathrm{N}}(\alpha_{\mathrm{k}+1}-\alpha_{\mathrm{k}})(\beta_{\mathrm{k}+1}+\beta_{\mathrm{k}}) \\
&= \frac{1}{2}\cdot\sum_{k=0}^{j-1}(\alpha_{k+1}-\alpha_k)(\beta_{k+1}+\beta_k)+\frac{1}{2}\cdot(\alpha_t-\alpha_j)(\beta_t+\beta_j) \\
&+ \frac{1}{2}\cdot(\alpha_{j+1}-\alpha_t)(\beta_{j+1}+\beta_t)+\frac{1}{2}\cdot\sum_{k=j+1}^{N}(\alpha_{k+1}-\alpha_k)(\beta_{k+1}+\beta_k) \\
&= \frac{1}{2}\cdot\sum_{k=0}^{j-1}(\alpha_{k+1}-\alpha_k)(\beta_{k+1}+\beta_k)+\frac{1}{2}\cdot\sum_{k=j+1}^{N}(\alpha_{k+1}-\alpha_k)(\beta_{k+1}+\beta_k) \\
&+ \frac{1}{2}\alpha_{j+1}\beta_{j+1}-\frac{1}{2}\alpha_j\beta_j+\frac{1}{2}\beta_t(\alpha_{j+1}-\alpha_j)-\frac{1}{2}\alpha_t(\beta_{j+1}-\beta_j) \\
&= c_j+\frac{1}{2}(\alpha_{j+1}-\alpha_j)\beta(t)-\frac{1}{2}\alpha(t)(\beta_{j+1}-\beta_j). \qquad (1.51)
\end{aligned}
$$

Cette ASC est maximale au point t^* tel que :

$$d\beta(t^*)=\left(\frac{\beta_{j+1}-\beta_j}{\alpha_{j+1}-\alpha_j}\right)d\alpha(t^*).$$

Posons $\alpha^*_j=\alpha(t^*)$, d'après la Proposition 4 on a :

$$\frac{1-p}{p}\cdot\frac{\mathcal{Q}^*(\alpha^*_j)}{1-\mathcal{Q}^*(\alpha^*_j)}=\frac{\beta_{j+1}-\beta_j}{\alpha_{j+1}-\alpha_j}.$$

On en déduit que le nouveau point optimal (α^*_j,β^*_j) dans la courbe COR est tel que :

$$\alpha^*_j=\bar{H}^*(u^*_j)) \quad \text{et} \quad \beta^*_j=\bar{G}^*(u^*_j)$$

où

$$u^*_j=\mathcal{Q}^*(\alpha^*_j)=\frac{p(\beta_{j+1}-\beta_j)}{(1-p)(\alpha_{j+1}-\alpha_j)+p(\beta_{j+1}-\beta_j)}=t^*. \qquad (1.52)$$

1.4.2.3 Partitionnement récursif de l'espace $\mathcal{X}$

Du point de vue de la construction de la partition $\mathcal{P}$, définie sur $\mathcal{X}$ par la fonction de score s_N^*, l'insertion du noeud (α_j^*, β_j^*) dans la courbe COR_N^*, tel que $\alpha_j^* \in]\alpha_j, \alpha_{j+1}[$, revient à scinder $R_{j+1} = \{x \in \mathcal{X} \mid \eta(x) > \mathcal{Q}^*(\alpha_{j+1})\}$ pour intercaler, dans la séquence croissante $(R_j)_{1 \leq j \leq N}$, un sous-ensemble R_j^* contenant $R_j = \{x \in \mathcal{X} \mid \eta(x) > \mathcal{Q}^*(\alpha_j)\}$, de la forme :

$$R_j^* = \{x \in \mathcal{X} \mid \eta(x) > \mathcal{Q}^*(\alpha_j^*)\}.$$

Il est plus facile cependant de raisonner en termes de partition disjointe. En considérant la D-représentation de la fonction de score s_N^* :

$$s_N^* = \sum_{j=1}^{N} (N - j + 1)\ \mathbb{I}_{C_j}$$

où

$$C_j = \{x \in \mathcal{X}\ :\ \mathcal{Q}^*(\alpha_{j+1}) < \eta(x) \leq \mathcal{Q}^*(\alpha_j)\},$$

à la $N^{\text{ème}}$ itération, l'ensemble C_{j+1} est scindé en deux sous-ensembles tels que $C_{j+1} = C_j^* \cup C_{j+1} \setminus C_j^*$ où

$$C_j^* = \{x \in \mathcal{X}\ :\ \mathcal{Q}^*(\alpha_{j+1}) > \eta(x) \geq \mathcal{Q}^*(\alpha_j^*)\}.$$

1.4.2.4 Résultat théorique de convergence

La Proposition 11 suivante montre que la fonction de score constante par morceaux s_N est *quasi-optimale* au sens où elle converge vers une fonction $s^* \in \mathcal{S}^*$ en norme $\mathcal{L}_1$ et $\mathcal{L}_\infty$.

Proposition 11 *(Taux de convergence)([Clémençon & Vayatis 2009d])*
Supposons que la courbe COR^* *est deux fois dérivable et concave et que sa deuxième dérivée est à valeurs dans un intervalle borné ne contenant pas* 0*. Il existe une séquence de fonctions de score constantes par morceaux* $(s_N)_{N \geq 1}$ *telles que, pour tout* $N \geq 1$, $s_N \in \mathcal{S}_N$ *et :*

$$\begin{aligned} \mathrm{ASC}^* - \mathrm{ASC}_{s_N} = d_1(s^*, s_N) &\leq C \cdot N^{-2}, \\ ||\mathrm{COR}^* - \mathrm{COR}_{s_N}||_\infty = d_\infty(s^*, s_N) &\leq C \cdot N^{-2}, \end{aligned}$$

où la constante C *dépend uniquement de la distribution* $\mathcal{P}_{X,Y}$ *des données.*

Il est connu dans la théorie de l'approximation ([Devore & Lorentz 1993]), que le taux $O(N^{-2})$ est atteint par toute approximation linéaire par morceaux, à condition que la taille de la grille soit de l'ordre de $O(N^{-1})$. Par ailleurs, il est important de noter que, la procédure d'approximation introduite ci-dessus étant adaptative, la sélection des noeuds COR_N^* dépend fortement de la courbe cible COR^*. On peut donc s'attendre à ce que ce schéma d'approximation mène à une constante plus fine.

Grâce à la procédure d'approximation que nous venons de décrire, il nous est possible de quantifier l'amélioration, en termes d'ASC, associée à l'ajout d'un nouveau noeud dans l'estimateur de COR^*. Cependant, la procédure TreeRank ne se restreint pas à introduire un unique noeud à chaque itération, mais considère le partitionnement simultané de chaque cellule de la partition $\mathcal{P}$ en cours de construction. Ce schéma d'approximation arborescent, que nous décrivons ci-dessous, permet ainsi d'obtenir, au terme de la $N^{\text{ème}}$ itération, une courbe $\mathrm{COR}_{2^N}^*$ à 2^N segments contre seulement N avec le schéma précédent.

1.4.3 Un schéma d'approximation arborescent

A la différence de la procédure décrite précédemment, le schéma d'approximation, sous-jacent à l'algorithme TreeRank, est structuré de façon arborescente. La procédure d'estimation est initialisée, de même que précédemment, en introduisant un premier noeud, vérifiant la Proposition 9, pour approcher la courbe COR* par une ligne brisée à deux segments. Puis, pour chaque itération suivante, un nouveau noeud est ajouté dans *chacun* des N segments de l'estimateur $\mathrm{COR}^*_{\mathrm{N}}$. Du point du vue du partitionnement de l'espace $\mathcal{X}$, ceci revient à scinder, à chaque itération, toutes les cellules de la partition courante $\mathcal{P}$ de $\mathcal{X}$ en deux sous-ensembles non vides. Afin de décrire ce schéma, nous introduisons quelques notations supplémentaires. Posons $N = 2^D$ avec $D \geq 0$, nous allons considérer les itérations d pour tout $d \in \{0, \cdots, D-1\}$.

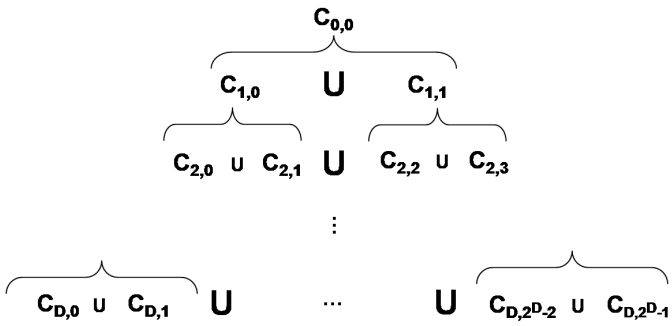


Figure 1.9 – Schéma arborescent de partitionnement récursif de l'espace $\mathcal{X}$.

Comme nous l'avons représenté sur le schéma de la Figure 1.9 ci-dessus, on initialise la procédure, pour $d = 0$, en posant $C_{0,0} = \mathcal{X}$ la cellule constituant la partition initiale $\mathcal{P}$ de $\mathcal{X}$. A la première itération, $d = 1$, la cellule $C_{0,0}$ est scindée en deux sous-ensembles que l'on notera : $C_{0,0} = C_{1,0} \cup C_{1,1}$. Après D itérations, on obtient une partition constituée de 2^D cellules disjointes, notées $C_{D,k}$ pour tout $k \in \{0, \cdots, 2^D - 1\}$. La partition ainsi obtenue est associée à une fonction de score constante par morceaux $s^*_{2^D}$, ayant la D-représentation suivante :

$$\forall x \in \mathcal{X}\ , \quad s^*_{2^D}(x) = \sum_{k=0}^{2^D-1} (2^D - k)\ \mathbb{I}\{x \in C_{D,k}\}.$$

Avec ces nouvelles notations, on peut détailler les étapes de la procédure TreeRank de la même façon que pour la procédure décrite dans la Partie 1.4.2.

1.4.3.1 Première itération

Par convention, on fixe les points extrêmes de l'estimateur de la courbe COR* :

$$\forall d \in \{0, \cdots, D-1\},\ \alpha^*_{d,0} = \beta^*_{d,0} = 0 \quad \text{et} \quad \alpha^*_{d,2^d} = \beta^*_{d,2^d} = 1.$$

Pour $d = 0$, la courbe COR* est estimée par la diagonale du plan (TFP, TVP) reliant les points extrêmes $(\alpha^*_{0,0}; \beta^*_{0,0})$ et $(\alpha^*_{0,2^0}; \beta^*_{0,2^0})$. Pour $d = 1$, on pose :

$$\alpha^*_{1,1} = \bar{H}^*(p) \quad \text{et} \quad \beta^*_{1,1} = \bar{G}^*(p)$$

et la courbe COR* est approchée par la ligne brisée à deux segments représentée en rouge sur la Figure 1.10 ci-dessous.

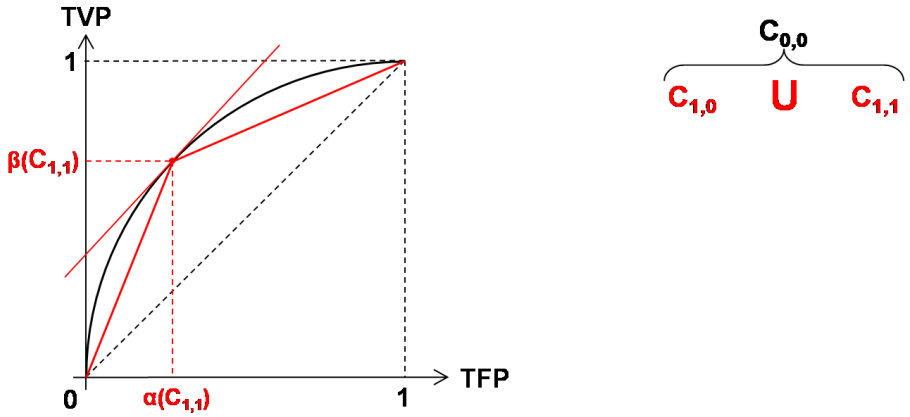


FIGURE 1.10 – Première itération de la procédure TREERANK.

1.4.3.2 $d^{\text{ème}}$ itération

A l'itération $d \geq 2$, l'estimateur de la courbe COR* est défini par la collection de points $\{(\alpha^*_{d,k}, \beta^*_{d,k})\}_{k=0,\ldots,2^d-1}$. Sur chaque intervalle $\left]\alpha^*_{d,k}, \alpha^*_{d,k+1}\right[$, on introduit un nouveau noeud, noté $C_{d+1,2k+1}$, en suivant la procédure décrite précédemment dans la Partie 1.4.2. La Figure 1.11 ci-dessous représente l'itération pour $d = 2$.

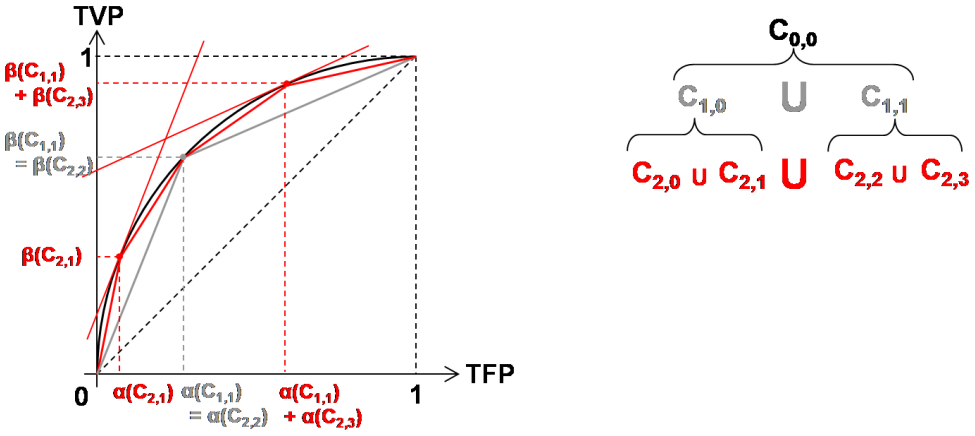


FIGURE 1.11 – Deuxième itération de la procédure TREERANK.

Chaque nouveau noeud $C_{d+1,2k+1}$ est donné par :

$$\alpha^*_{d+1,2k+1} = \bar{H}^* \left(u^*_{d+1,2k+1}\right) \quad \text{et} \quad \beta^*_{d+1,2k+1} = \bar{G}^* \left(\Delta^*_{d+1,2k+1}\right),$$

où

$$u^*_{d+1,2k+1} = \frac{p(\beta^*_{d,k+1} - \beta^*_{d,k})}{(1-p)(\alpha^*_{d,k+1} - \alpha^*_{d,k}) + p(\beta^*_{d,k+1} - \beta^*_{d,k})}.$$

Enfin, on pose $u^*_{d+1,2k} = u^*_{d,k}$ et on met à jour la liste des noeuds en les renommant comme suit :

$$\alpha^*_{d+1,2k} = \alpha^*_{d,k} \quad \text{et} \quad \beta^*_{d+1,2k} = \beta^*_{d,k}.$$

Finalement, pour tout niveau $D \geq 1$ fixé, la partition $\mathcal{P}$ définie sur $\mathcal{X}$ par la fonction $s^*_{2^D}$ est constituée des cellules :

$$\forall\, k \in \{0, \cdots, 2^D - 1\},\, C^*_{D,k} = \{x \in \mathcal{X} \ :\ u^*_{D,k+1} < \eta(x) \leq u^*_{D,k}\},$$

avec par convention $u^*_{D,0} = 0$ et $u^*_{D,2^D} = 1$. Avec ces notations, la I-représentation de la fonction de score $s^*_{2^D}$ repose sur la séquence croissante d'ensembles $R^*_{D,k}$ de la forme :

$$\forall k \in \{1, \ldots, 2^D - 1\},\ R^*_{D,k} = C^*_{D,k} \cup R^*_{D,k-1},$$

avec par convention $R^*_{D,0} = C^*_{D,0}$.

1.4.3.3 Approximation de type *éléments finis*

Afin de définir une expression analytique de la courbe $\mathrm{COR}^*_{2^D}$ affine par morceaux associée à la fonction de score constante par morceaux

$$s^*_{2^D}(x) = \sum_{k=0}^{2^D-1} (2^D - k)\ \mathbb{I}\{x \in C^*_{D,k}\},$$

on considère les *fonctions élémentaires* définies pour tout $d \in \{0, \cdots, D-1\}$ et tout $k \in \{0, \cdots, 2^d - 1\}$ par

$$\phi^*_{d,k}(.) = \phi(.;\ (\alpha^*_{d,k-1}, \alpha^*_{d,k})) - \phi(.;\ (\alpha^*_{d,k}, \alpha^*_{d,k+1})), \tag{1.53}$$

avec les notations suivantes :

$$\forall\, z \in \mathbb{R},\, \phi(z;\ (z_1, z_2)) \quad = \quad \frac{z - z_1}{z_2 - z_1}\mathbb{I}\{z \in [z_1, z_2]\},$$

pour tout couple $(z_1, z_2) \in \mathbb{R}^2$ tel que $z_1 < z_2$. On pose de plus

$$\forall\, d \in \{0, \cdots, D-1\},\, \phi^*_{d,2^d}(.) = \phi(.;\ (\alpha^*_{d,2^d-1}, 1)).$$

Avec ces notations, la procédure TreeRank peut être vue comme une approximation de type *éléments finis* de la courbe COR^*. En effet, en se munissant des fonctions élémentaires définies par l'équation (1.53) et d'une grille $\{\alpha^*_{D,k}\}_{0 \leq k \leq 2^D - 1}$, on peut approcher la courbe COR^* par la fonction affine par morceaux $\mathrm{COR}^*_{2^D}$ définie comme suit :

$$\forall \alpha \in [0,1]\,,\ \mathrm{COR}^*_{2^D}(\alpha) = \mathrm{COR}(\mathrm{s}^*_{2^D}, \alpha) = \sum_{\mathrm{k}=1}^{2^D} \beta^*_{\mathrm{D,k}} \phi^*_{\mathrm{D,k}}(\alpha). \tag{1.54}$$

On peut noter que cet estimateur est croissant et concave, tout comme la courbe COR*. De plus, avec cette représentation, on peut déduire l'expression suivante pour l'estimation correspondante de l'ASC optimale :

$$\mathrm{ASC}(\mathrm{s}^*_{\mathrm{D}}) = \frac{1}{2}\sum_{\mathrm{k}=1}^{2^{\mathrm{D}}-1}(\alpha^*_{\mathrm{D,k+1}} - \alpha^*_{\mathrm{D,k-1}})\beta^*_{\mathrm{D,k}}. \tag{1.55}$$

Nous décrivons l'heuristique TreeRank en détail dans la partie suivante, pouvant être vue comme une version empirique du schéma d'approximation présenté ci-dessus.

1.5 L'algorithme TreeRank

L'algorithme TreeRank proposé dans [Clémençon & Vayatis 2009d] est une version empirique, basée sur l'observation d'un échantillon $\mathcal{D}_n = \{(X_i, Y_i),\, 1 \leq i \leq n\}$ d'observations de $\mathcal{X} \times \{-1, +1\}$, de la procédure d'approximation arborescente que nous venons de décrire, dans laquelle les grandeurs théoriques inconnues sont remplacées par leurs contreparties empiriques. Comme nous venons de le voir, ce schéma d'approximation repose sur une stratégie hiérarchique de partitionnement récursif de l'espace $\mathcal{X}$. La version empirique, que nous proposons, repose sur la construction d'un *arbre binaire d'ordonnancement* représentant cette stratégie de partitionnement, dans le même esprit que les arbres de classification binaire de type CART, introduits dans [Breiman *et al.* 1984]. A ce titre, l'heuristique TreeRank permet d'étendre la notion d'*arbre de décision* au problème de l'ordonnancement. Avant de décrire cet algorithme, nous allons donc introduire la notion d'*arbre d'ordonnancement*.

1.5.1 Arbres binaires d'ordonnancement

Un *arbre binaire d'ordonnancement* est un arbre binaire muni d'une orientation gauche-droite (voir Figure 1.12). Soit $\mathcal{T}_D$ un tel arbre, de profondeur $D \geq 0$, définissant une partition $\mathcal{P}$ finie, disjointe et orientée de l'espace d'entrée $\mathcal{X}$. La *racine* de cet arbre, notée $\mathcal{C}_{0,0}$, correspond à l'espace d'entrée tout entier $\mathcal{X}$.

Chaque *noeud* interne de l'arbre, $\mathcal{C}_{d,k}$, avec $0 \leq d < D-1$ et $0 \leq k < 2^d - 1$, correspond à une cellule de la partition $\mathcal{P}$ de $\mathcal{X}$ et engendre deux *enfants* non vides. Selon l'orientation définie précédemment, le *noeud fils gauche*, noté $\mathcal{C}_{d+1,2k}$, contient les *meilleures* observations du *noeud parent* $\mathcal{C}_{d,k}$ et on note $\mathcal{C}_{d+1,2k+1} = \mathcal{C}_{d,k} \setminus \mathcal{C}_{d+1,2k}$ son complémentaire, le *noeud fils droit*. La *branche* de l'arbre reliant le noeud parent $\mathcal{C}_{d_k}$ à ses enfants représente la règle de partitionnement de la cellule $C_{d,k} \in \mathcal{P}$ en deux sous-ensembles non vides.

Les *noeuds terminaux* de l'arbre, aussi appelés *feuilles terminales*, caractérisent la partition $\mathcal{P}$ de $\mathcal{X}$, les éléments appartenant à une même cellule étant *classés* ex-aequo et l'ordonnancement de ces cellules découlant directement de l'orientation de l'arbre $\mathcal{T}_D$. La fonction de score associée à cette partition, donnée par

$$s_{\mathcal{T}}(x) = \sum_{\mathcal{C}_{d,k}\text{: cellule terminale}} 2^D(1 - k/2^d) \cdot \mathbb{I}\{x \in \mathcal{C}_{d,k}\},$$

peut-être lue directement sur les feuilles terminales de l'arbre, de la gauche vers la droite.

La valeur $s_{\mathcal{T}}(x)$ du score d'une observation $x \in \mathcal{X}$ se calcule rapidement de façon descendante, en exploitant la structure ordonnée en tas. Partant de la valeur initiale 2^D à la racine de l'arbre, à chaque noeud $\mathcal{C}_{d,k}$ interne, on laisse le score inchangé si l'observation est déplacée vers la feuille gauche et on lui soustrait $2^{D-(d+1)}$ si elle est déplacée vers la feuille droite. Cette procédure est illustrée par la Figure 1.12.

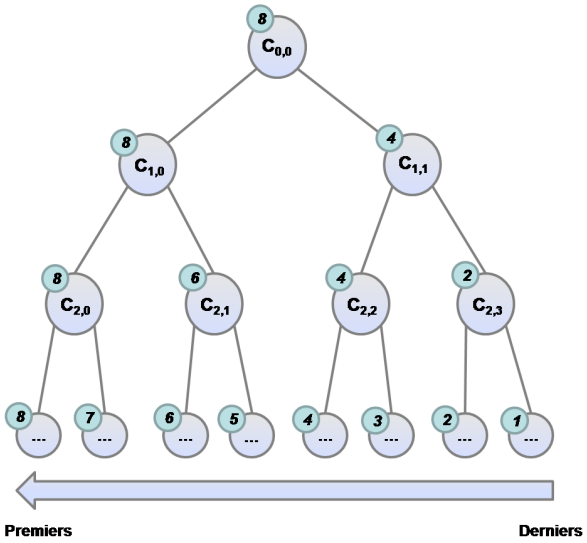


Figure 1.12 – Arbre binaire d'ordonnancement obtenu par le biais de la procédure TreeRank.

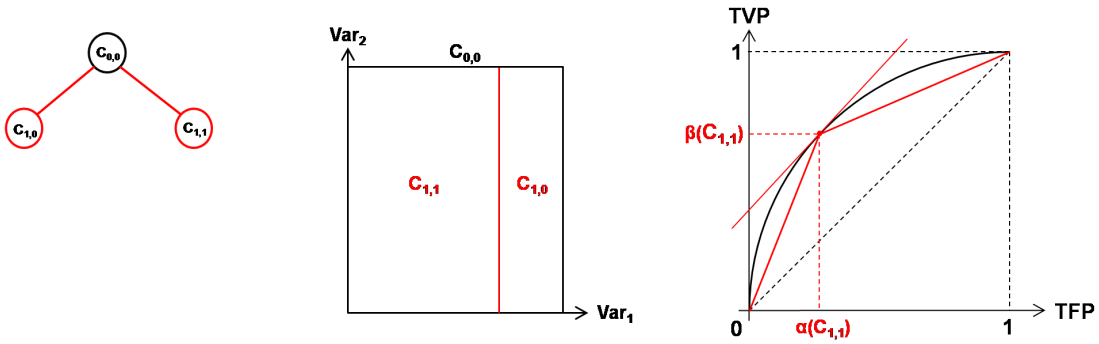
1.5.2 Un algorithme de partitionnement récursif

Nous allons maintenant décrire l'heuristique TreeRank. De la même façon que dans la méthode CART, l'algorithme TreeRank scinde l'espace des entrées $\mathcal{X}$ de manière récursive, perpendiculairement aux axes portés par les variables : à chaque itération, on cherche à scinder une cellule $\mathcal{C}$ selon un niveau choisi pour une variable précise, le niveau et la variable étant choisis de manière adaptative de sorte à maximiser l'ASC locale. En d'autres termes, à chaque itération, on cherche la meilleure règle de partitionnement de la forme $X^{(j)} \geq u_k$ ou encore $X^{(j)} < u_k$, où $X^{(j)}$, $j \geq 1$, désigne la $j^{\text{ème}}$ variable -composante- et u_k, $k \geq 1$, représente un seuil dans le domaine de définition de la variable $X^{(j)}$. Nous avons représenté les deux premières itérations de l'algorithme sur la Figure 1.13 ci-dessous.

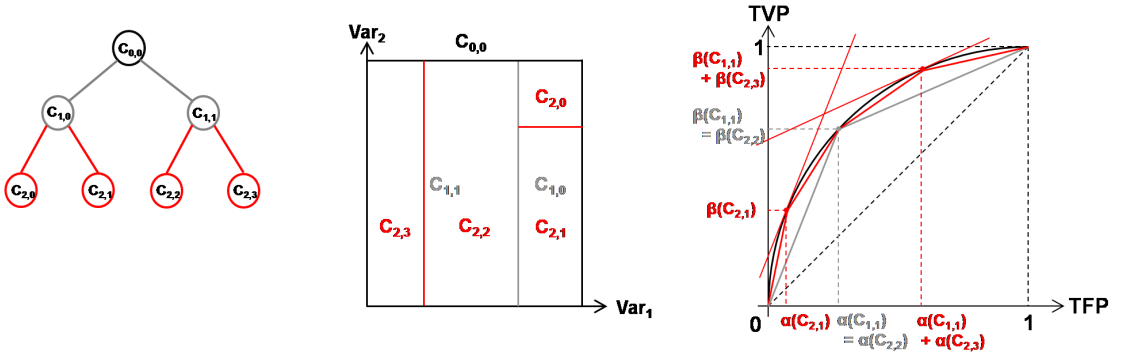
Soit $\mathcal{D}_n = \{(X_1, Y_1), \ \ldots, \ (X_n, Y_n)\}$ un échantillon d'apprentissage constitué de n copies *i.i.d.* du couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$. On pose :

$$n_+ = \sum_{i=1}^{n} \mathbb{I}\{Y_i = 1\} \ \text{ et } \ n_- = \sum_{i=1}^{n} \mathbb{I}\{Y_i = -1\} \ .$$

On suppose que l'on dispose d'une collection $\mathcal{C}$ de sous-ensembles C de l'espace $\mathcal{X}$. On



a. Première itération



b. Deuxième itération

FIGURE 1.13 – Première et deuxième itérations de l'heuristique TREERANK.

définit, les taux empiriques de faux et vrais positifs pour tout $C \subset \mathcal{X}$:

$$\hat{\alpha}(C) = \frac{1}{n_-} \sum_{i=1}^{n} \mathbb{I}\{X_i \in C,\ Y_i = -1\},$$

$$\hat{\beta}(C) = \frac{1}{n_+} \sum_{i=1}^{n} \mathbb{I}\{X_i \in C,\ Y_i = +1\}.$$

Remarque 7 *On peut remarquer que contrairement à l'algorithme de classification* CART*, le partitionnement d'une cellule est basé sur la maximisation du critère* ASC *qui dépend du noeud correspondant, par le biais de ses taux de vrais et faux positifs. Intuitivement, ceci relève du fait que le problème d'ordonnancement est de nature globale et implique de classer toutes les instances les unes par rapport aux autres ou, en d'autres termes, de comparer les cellules les unes par rapport aux autres. Nous reviendrons là-dessus dans le chapitre 2.*

Un des principaux avantages de l'algorithme TREERANK est le faible nombre de paramètres à régler pour apprendre une règle de score. En réalité, une fois la méthode de partitionnement choisie, le seul élément à paramétrer est le critère d'arrêt de la phase d'apprentissage (la profondeur D). Comme nous allons le voir, le choix de ces deux paramètres revêt une importance cruciale, en ce sens qu'ils influent autant sur les propriétés

Algorithme TreeRank

1. **Initialisation.** On pose $C_{0,0} = \mathcal{D}_n$ et par convention, pour tout $d \geq 0$, on a $\alpha_{d,0} = \beta_{d,0} = 0$ et $\alpha_{d,2^d} = \beta_{d,2^d} = 1$.

2. **Itérations.** Pour $d = 0, \ldots, D-1$ et $k = 0, \ldots, 2^d - 1$:

   (a) (Optimisation locale de l'ASC.) On pose la mesure d'entropie

   $$\Lambda_{d,k+1}(C) = (\alpha_{d,k+1} - \alpha_{d,k})\hat{\beta}(C) - (\beta_{d,k+1} - \beta_{d,k})\hat{\alpha}(C) \ .$$

   On cherche le meilleur sous-rectangle $C_{d+1,2k}$ du rectangle $C_{d,k}$ au sens de l'ASC :

   $$C_{d+1,2k} = \underset{C \in \mathcal{C},\ C \subset C_{d,k}}{\arg\max} \ \Lambda_{d,k+1}(C) \ .$$

   On pose $C_{d+1,2k+1} = C_{d,k} \setminus C_{d+1,2k}$.

   (b) (Mise à jour.) On pose :

   $$\begin{aligned} \alpha_{d+1,2k+1} &= \alpha_{d,k} + \hat{\alpha}(C_{d+1,2k}) \qquad &\text{et } \ \alpha_{d+1,2k+2} &= \alpha_{d,k+1} \\ \beta_{d+1,2k+1} &= \beta_{d,k} + \hat{\beta}(C_{d+1,2k}) \qquad &\text{et } \ \beta_{d+1,2k+2} &= \beta_{d,k+1}. \end{aligned}$$

3. **Sortie.** Après D itérations, on obtient une fonction de score constante par morceaux :

   $$\forall\, x \in \mathcal{X},\ s_{2^D}(x) = \sum_{k=0}^{2^D-1} (2^D - k)\ \mathbb{I}\{x \in C_{D,k}\},$$

   ainsi qu'une estimation de l'ASC

   $$\widehat{\mathrm{ASC}}(s_{2^D}) = \frac{1}{2} \sum_{k=1}^{2^D-1} (\alpha_{D,k+1} - \alpha_{D,k-1})\beta_{D,k} = \frac{1}{2} + \frac{1}{2} \sum_{k=0}^{2^{D-1}-1} \Lambda_{D-1,k+1}(C_{D,2k})$$

   et de la courbe $\mathrm{COR}^*_{2^D}$

   $$\forall\, \alpha \in [0,1]\ ,\ \widehat{\mathrm{COR}}(s_{2^D}, \alpha) = \sum_{k=1}^{2^D} \beta_{D,k}\phi_{D,k}(\alpha),$$

   où $\forall\, k \in \{0, \cdots, 2^D - 1\}$, $\phi_{D,k}(.) = \phi(.;\ (\alpha_{D,k-1}, \alpha_{D,k})) - \phi(.;\ (\alpha_{D,k}, \alpha_{D,k+1}))$, et $\phi_{D,2^D}(.) = \phi(.; (\alpha_{D,2^D-1}, 1))$.

théoriques de l'algorithme, que sur ses performances empiriques.

1.5.3 Résultats théoriques

Le résultat suivant montre que, tout comme l'estimateur théorique $\mathrm{COR}(\mathrm{s}_\mathrm{D}^*, .)$ de COR^*, la courbe $\widehat{\mathrm{COR}}(s_D, .)$ produite par l'algorithme TreeRank est concave dès que le sous-ensemble $\mathcal{C}$ est *stable par union*, *i.e.* quand ses éléments sont obtenus par la réunion d'ensembles élémentaires. Notons toutefois, que cette condition n'est pas garantie pour la version de l'algorithme présentée précédemment, reposant sur le partitionnement de l'espace d'entrée $\mathcal{X}$ perpendiculairement à ses axes, dans l'esprit de l'algorithme CART.

Proposition 12 *(Concavité de l'estimateur de* COR^* *[Clémençon & Vayatis 2009d]) Supposons que l'on dispose d'une classe* $\mathcal{C}$ *d'ensembles, stable par union, i.e.* $\forall (C, C') \in \mathcal{C}^2$ *:* $C \cup C' \in \mathcal{C}$*. On considère la fonction de score* s_D *obtenue par le biais de l'algorithme* TreeRank *après* 2^D *itérations. Sa courbe* COR *empirique,* $\widehat{\mathrm{COR}}(s_D, .)$*, est concave.*

Dans le cas précis où la classe $\mathcal{C}$ est stable par union, on peut par ailleurs montrer la consistance de la classe des partitions induites par l'algorithme TreeRank, au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$ définies sur l'espace des courbes COR. Ce résultat de convergence repose sur le contrôle de la complexité de la classe d'ensembles $\mathcal{C}$ au moyen du coefficient de pulvérisation $\mathcal{S}(\mathcal{C}, N)$, lié à la dimension de Vapnik-Chervonenkis notée VC ([Vapnik 1982]).

La classe $\mathcal{C}$ définit une collection de prédicteurs sur l'espace $\mathcal{X}$ et la dimension VC fournit une mesure de sa complexité, en évaluant le cardinal du plus grand ensemble de points pouvant être *pulvérisé* par cette collection, *i.e.* pouvant être séparés parfaitement, sans erreurs de prédiction. Le $N^{\text{ème}}$ coefficient de pulvérisation, $\mathcal{S}(\mathcal{C}, N)$, correspond au nombre de *parties* pouvant être formées par la classe $\mathcal{C}$ parmi N observations.

Théorème 2 *([Clémençon & Vayatis 2009d]) Pour tout* $n \geq 1$*, on considère les fonctions de score* s_N *associées à la partition* $\mathcal{P}_N$ *de* $\mathcal{X}$*, résultant de l'algorithme* TreeRank *appliqué à un échantillon d'apprentissage de taille* N *et une classe de sous-ensembles* $\mathcal{C}_N$*. On suppose que :*

- $\mathcal{X}$ *est borné,*
- $\mathcal{C}_N$ *est stable par union,*
- *la classe* $\mathcal{C}_N$ *est telle que*

$$\lim_{N \to \infty} \frac{\log\left(\mathcal{S}(\mathcal{C}_N, N)\right)}{N} = 0,$$

où $\mathcal{S}(\mathcal{C}_N, N)$ *est le* $N^{\text{ème}}$ *coefficient de pulvérisation de la classe des ensembles* $\mathcal{C}_N$*,*

- *le diamètre d'une cellule quelconque de* $\mathcal{P}_N$ *tend vers 0 quand* N *tend vers l'infini,*

alors on a, quand $N \to \infty$*,*

$$\mathrm{ASC}(\mathrm{s}^*) - \mathrm{ASC}(\mathrm{s}_\mathrm{N}) = \mathrm{d}_1(\mathrm{s}^*, \mathrm{s}_\mathrm{N}) \to 0 \ \ \textit{presque sûrement}\ ,$$

Si de plus

- *la densité de la distribution* H^* *est bornée,*
- *il existe une constante* $c > 0$ *telle que* $H^{*'}(u) \geq 1/c$ *pour tout* $u \in [0, 1]$*,*

alors, on a aussi, quand N *tend vers* ∞*,*

$$d_\infty(s^*, s_N) \to 0 \ \ \textit{presque sûrement}\ .$$

On peut faire quelques remarques sur les hypothèses avancées dans ce théorème. La condition de borne sur l'espace $\mathcal{X}$ est une simplification, qui peut être retirée au prix d'une preuve plus longue (voir l'argument proposé dans [L.Devroye *et al.* 1996]). Quant à l'hypothèse de complexité sur les partitions induites par TreeRank, elle découle de l'approche introduite dans [Lugosi & Nobel 1996] (voir aussi [L.Devroye *et al.* 1996], Chapitre 21). Le résultat repose sur le contrôle du $N^{\text{ème}}$ coefficient de pulvérisation de la collection des ensembles pouvant être obtenus par union des ensembles $C \in \mathcal{C}_N$. En particulier, on peut noter que, étant donnée l'hypothèse de stabilité par union, ce dernier est ici réduit à $\mathcal{S}(\mathcal{C}_N, N)$.

Si l'on formule quelques hypothèses supplémentaires, on peut relâcher l'hypothèse de stabilité par union et établir des taux de convergence pour la fonction de score produite par TreeRank.

Théorème 3 *([Clémençon & Vayatis 2009d]) Supposons que les conditions de la Proposition 11 sont vérifiées. Supposons de plus que la classe $\mathcal{C}$ des sous-ensembles candidats contient tous les ensembles de niveaux R^*_α, $\alpha \in [0,1]$ et qu'il est stable par intersection, i.e. $\forall (C, C') \in \mathcal{C}^2$: $C \cap C' \in \mathcal{C}$. Supposons enfin que la dimension* VC *de $\mathcal{C}$ est finie et égale à V. Pour tout $\delta > 0$, il existe alors une constante c_0 et deux constantes universelles c_1, c_2 telles que l'on ait, avec probabilité au moins $(1-\delta)$, pour tout $D \geq 1$, $n \in \mathbb{N}$:*

$$
\begin{aligned}
d_1(\hat{s}_D, s_D) &\leq c_0^D \left\{ \left(\frac{c_1^2 V}{n} \right)^{\frac{1}{2D}} + \left(\frac{c_2^2 \log(1/\delta)}{n} \right)^{\frac{1}{2D}} \right\}, \\
d_\infty(\hat{s}_D, s_D) &\leq c_0^D \left\{ \left(\frac{c_1^2 V}{n} \right)^{\frac{1}{2(D+1)}} + \left(\frac{c_2^2 \log(1/\delta)}{n} \right)^{\frac{1}{2(D+1)}} \right\}.
\end{aligned}
$$

Soulignons que la borne pour $d_\infty(\hat{s}_D, s^*)$ correspond à une bande de confiance pour $\mathrm{COR}(\hat{\mathrm{s}}_\mathrm{D}, .)$ dans l'espace fonctionnel des fonctions continues à valeurs réelles définies sur $[0,1]$, équipé de la norme supérieure, alors que celle pour $d_1(\hat{s}_D, s^*)$ implique un intervalle de confiance pour la valeur scalaire $\mathrm{ASC}(\hat{\mathrm{s}}_\mathrm{D})$. Par ailleurs, il est important de noter que les taux de convergence formulés dans le théorème précédent dépendent de la profondeur D de l'arbre. Aussi, la convergence nécessite d'avoir une profondeur $D = D_N$ tendant très lentement vers l'infini, comme le montre le corollaire suivant.

Corollaire 4 *([Clémençon & Vayatis 2009a])*
Soit $D = D_n$ tel que $D_n \sim \sqrt{\log n}$, quand $n \to \infty$. Alors, pour tout $\delta > 0$, il existe une constante κ telle que l'on ait, avec probabilité au moins $1-\delta$, pour tout $n \in \mathbb{N}$:

$$
d_i(\hat{s}_{D_n}, s^*) \leq \exp(-\kappa \sqrt{\log n}), \quad i \in \{1, \infty\}.
$$

Ces faibles taux de convergence résultent de la strucure hiérarchique de la partition induite par la construction de l'arbre. On peut noter que, au prix de quelques hypothèses supplémentaires sur la régularité de la fonction de régression, on peut atteindre de meilleures bornes, notamment pour des fonctions de score obtenues à partir de partitions fixes (voir [Clémençon & Vayatis 2009c]).

Une conclusion importante que l'on peut tirer du Théorème 3 est que, si l'on choisit $D_n \sim \sqrt{\log n}$, la courbe COR empirique $\widehat{\mathrm{COR}}(s_{D_n}, .)$ obtenue par l'algorithme TreeRank est un estimateur consistant de la courbe COR^*, au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$ sur l'espace des courbes COR. Dans ce cas, la borne pour la courbe $\mathrm{COR}(\mathrm{s}_{\mathrm{D_n}}, .)$ reste valide.

1.5.4 Résultats expérimentaux

Nous présentons quelques résultats expérimentaux sur données simulées, afin d'illustrer les performances de l'algorithme TreeRank décrit dans la Partie 1.5.2. Pour cela, nous considérons deux exemples, en dimension 2, afin de visualiser facilement les ensembles de niveaux de la probabilité à posteriori η. Dans le premier exemple, *Unif2d*, les données générées sont distribuées selon un mélange de distributions uniformes et pour le second, *GaussCroix2d*, nous considérons un mélange de lois gaussiennes conditionnelles. Dans les deux situations, le taux théorique d'objets positifs $p = \mathbb{P}\{Y = +1\}$ est fixé à $1/2$. Dans chaque cas, l'échantillon généré est divisé en un échantillon d'apprentissage constitué de 2000 observations, servant à construire un arbre orienté $\mathcal{T}$ via l'algorithme TreeRank, et un échantillon test constitué de 500 observations permettant d'estimer la courbe COR de la règle de score ainsi obtenue.

1.5.4.1 Exemple *Unif2d*

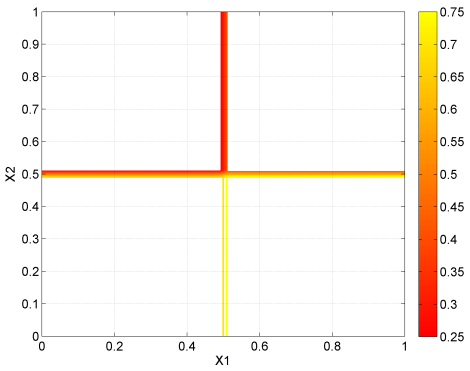
Pour cet exemple, les données sont générées comme suit. On considère le carré unité $\mathcal{X} = [0,1]^2$, que l'on divise en quatre quarts $\mathcal{X}_1 = [0,1/2]^2$, $\mathcal{X}_2 = [1/2,1] \times [0,1/2]$, $\mathcal{X}_3 = [1/2,1]^2$ et $\mathcal{X}_4 = [0,1/2] \times [1/2,1]$ et on note $\mathcal{U}_C$ la distribution uniforme sur un ensemble $C \subset \mathcal{X}$ mesurable. Avec ces notations, les distributions utilisées pour générer les observations s'écrivent comme suit :

$$\begin{aligned} H(dx) &= 0.2 \cdot \mathcal{U}_{\mathcal{X}_1} + 0.1 \cdot \mathcal{U}_{\mathcal{X}_2} + 0.3 \cdot \mathcal{U}_{\mathcal{X}_3} + 0.4 \cdot \mathcal{U}_{\mathcal{X}_4}, \\ G(dx) &= 0.4 \cdot \mathcal{U}_{\mathcal{X}_1} + 0.3 \cdot \mathcal{U}_{\mathcal{X}_2} + 0.2 \cdot \mathcal{U}_{\mathcal{X}_3} + 0.1 \cdot \mathcal{U}_{\mathcal{X}_4}. \end{aligned}$$

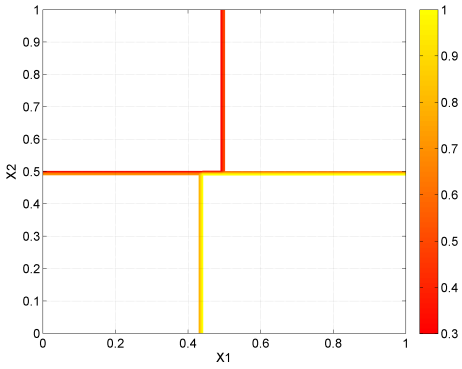
On peut aisément vérifier que dans ce cas précis, la probabilité à posteriori η correspondante est constante par morceaux et donnée par :

$$\eta(x) = 0.7 \cdot \mathbb{I}\{x \in \mathcal{X}_1\} + 0.75 \cdot \mathbb{I}\{x \in \mathcal{X}_2\} + 0.4 \cdot \mathbb{I}\{x \in \mathcal{X}_3\} + 0.2 \cdot \mathbb{I}\{x \in \mathcal{X}_4\}.$$

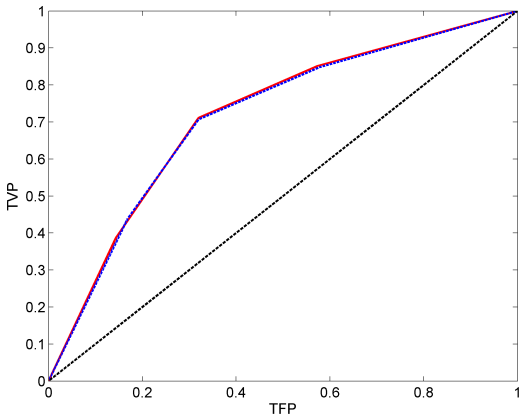
La fonction de régression générée pour cet exemple définit donc quatre ensembles de niveaux, que nous avons représentés sur la Figure 1.14(a), la couleur rouge foncée (resp. jaune clair) traduisant une probabilité à posteriori élevée (resp. faible). Aussi, nous avons paramétré l'algorithme TreeRank en fixant une profondeur $D = 1$, afin d'obtenir un arbre d'ordonnancement comprenant exactement quatre feuilles terminales. La collection des quatre ensembles de niveaux estimés par la procédure d'apprentissage est représentéé sur la Figure 1.14(b). Enfin, la Figure 1.14(c) permet de visualiser la courbe COR optimale (en trait plein rouge) et la courbe COR test (en pointillés bleus) de la règle d'ordonnancement obtenue.
Sur cet exemple simple, où les ensembles de niveaux de la fonction de régression générée ont exactement la même forme que les cellules produites par l'algorithme TreeRank (*i.e.* des rectangles de côtés parallèles aux axes de l'espace $\mathcal{X}$), l'ordonnancement obtenu est quasi-optimal. On constate en effet, que les courbes COR test et optimale sont quasiment confondues et la Figure 1.14(b) montre que les ensembles de niveaux de η sont estimés de façon très précise.



a. Ensembles de niveaux de la probabilité η.



b. Ensembles de niveaux estimés par TreeRank.



c. Courbes COR optimale (rouge) et test (pointillés bleus).

Figure 1.14 – Exemple *Unif2d*

1.5.4.2 Exemple *GaussCroix2d*

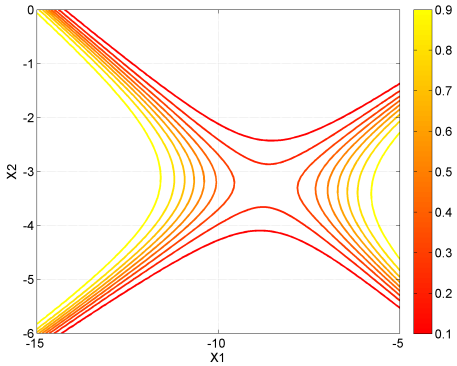
Pour cet exemple, on se place sur l'espace $\mathcal{X} = \mathbb{R}^2$. Soit Z un vecteur aléatoire gaussien de dimension 2, de moyenne m et de matrice de covariance Γ, on notera $\mathcal{N}_C(m, \Gamma)$ la distribution conditionnelle de Z sachant $Z \in C$, pour tout borélien C chargé par la mesure de Lebesgue sur $\mathbb{R}^2$. Muni de cette notation, les lois utilisées pour ce deuxième exemple sont données par :

$$\begin{aligned} H(dx) &= \mathcal{N}_{\mathcal{X}}\left(\begin{pmatrix} -1.79 \\ 0.77 \end{pmatrix}, \begin{pmatrix} 9.45 & 5.91 \\ 5.91 & 3.94 \end{pmatrix}\right), \\ G(dx) &= \mathcal{N}_{\mathcal{X}}\left(\begin{pmatrix} -0.75 \\ -0.60 \end{pmatrix}, \begin{pmatrix} 7.41 & 2.66 \\ 2.66 & 1.10 \end{pmatrix}\right). \end{aligned}$$

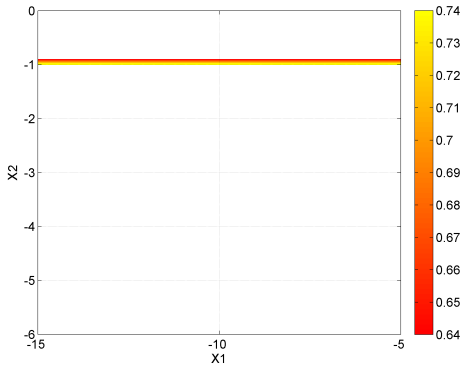
Pour $p = 1/2$, la transformée logistique de la probabilité à posteriori générée est donnée par :

$$\text{logit}\left(\eta(\mathrm{x})\right) = 0.35\mathrm{x}_1^2 - 1.4\mathrm{x}_2^2 + 5.74\mathrm{x}_1 - 10.01\mathrm{x}_2 - 0.1\mathrm{x}_1\mathrm{x}_2 + 7.71.$$

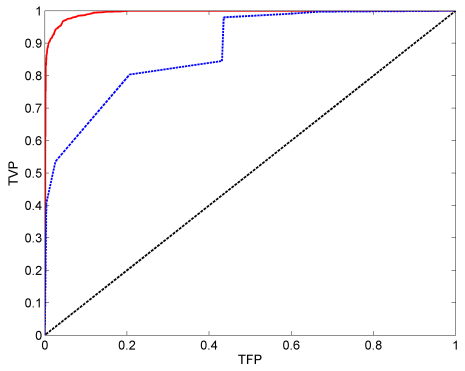
Cette probabilité étant continue, la collection $\mathcal{C}^*$ contient une infinité d'ensembles de niveaux. Nous en avons représentés 8 sur la Figure 1.15(a) ci-dessous, pour donner un aperçu de la forme de ces ensembles. Afin de définir une partition de l'espace d'entrée constituée de 8 cellules disjointes, nous avons fixé à $D = 3$ la profondeur de la procédure d'apprentissage. Les ensembles ainsi obtenus sont donnés sur la Figure 1.15(b). De même que précédemment, les courbes COR optimale et test sont tracées (respectivement en trait plein rouge et pointillés bleus) sur la Figure 1.15(c).



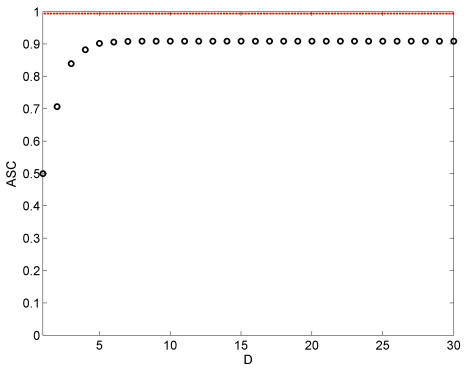
a. Ensembles de niveaux de la probabilité η.



b. Ensembles de niveaux estimés par TreeRank.



c. Courbes COR optimale (rouge) et test (pointillés bleus).



d. Evolution de l'ASC empirique de test avec la profondeur D.

FIGURE 1.15 – Exemple *GaussCroix2d*

L'observation de la Figure 1.15(b) montre clairement que, dans cet exemple, la classe des cellules produites par l'algorithme TreeRank n'est pas suffisamment *riche* pour estimer les ensembles de niveaux de la fonction de régression. Ce constat est renforcé par la comparaison des courbes COR représentées sur la Figure 1.15(c).

L'écart entre les deux courbes s'explique par la présence d'un certain nombre de points *contraignant* fortement la courbe COR test, qui correspondent en réalité aux premières scissions de l'espace $\mathcal{X}$. Ceci est dû au fait que les erreurs commises lors des premières itérations de la procédure d'apprentissage ne peuvent être rattrapées au cours des itérations suivantes et ce, quelle que soit la profondeur de la procédure (cf Chapitre 2). Ceci implique

notamment que la performance de l'algorithme, en termes d'ASC, est limitée. Nous avons représenté sur la Figure 1.15(d) l'évolution de l'ASC empirique de la règle de score estimée par TreeRank en fonction de la profondeur $D \geq 1$ de la procédure d'apprentissage. Cette expérience montre que malgré l'augmentation de la profondeur D, l'ASC empirique est bornée par une valeur très en deçà de l'ASC de la courbe COR optimale.

Ces résultats montrent que la règle de partitionnement choisie pour l'implémentation de l'algorithme TreeRank n'est pas suffisamment flexible pour estimer précisément les ensembles de niveaux de la probabilité à posteriori. Le choix de scinder l'espace perpendiculairement aux axes portés par les variables relève principalement du compromis entre performance et coût de calcul. On peut naturellement considérer d'autres règles de partitionnement, comme par exemple des scissions linéaires dans l'espace $\mathcal{X}$, conduisant à une classe de partitions plus riches mais moins interprétables. L'objectif du Chapitre 2 est de proposer des règles de partitionnement plus flexibles, afin d'améliorer les performances globales de l'algorithme.

1.6 Conclusion et perspectives

Dans ce premier chapitre, la problématique d'ordonnancement binaire a été définie sous la forme d'un problème de scoring et la question de l'estimation d'une fonction de score $s : \mathcal{X} \to \mathbb{R}$, permettant d'ordonner les observations d'un espace $\mathcal{X} \subset \mathbb{R}^q$, $q \geq 1$, a été abordée. Nous avons vu que la mesure de la performance d'une telle règle de score pouvait reposer sur l'estimation d'un critère fonctionnel (la courbe COR) ou d'un critère scalaire (l'ASC) et nous avons montré que le problème de la M-estimation d'une fonction de score optimale pouvait être abordé comme celui de l'approximation de la courbe COR^* par une fonction affine par morceaux, revenant à optimiser récursivement l'ASC empirique calculée localement sur des sous-ensembles de l'espace $\mathcal{X}$.

Nous avons ainsi introduit la méthode de scoring TreeRank, produisant des fonctions de score constantes par morceaux, de courbes COR affines par morceaux *asymptotiquement optimales*. Cette approche, qui repose sur le partitionnement récursif de l'espace $\mathcal{X}$, produit des règles de prédiction pouvant être représentées sous la forme d'arbres d'ordonnancement. L'algorithme TreeRank, mettant en oeuvre cette méthode, a été présenté en détail.

Enfin, des résultats expérimentaux préliminaires ont été présentés et ont permis de mettre en évidence l'impact du choix de la règle de partitionnement sur les performances de la procédure d'apprentissage. L'objet du chapitre suivant est de proposer des méthodes de scissions plus flexibles afin d'améliorer les performances de l'algorithme proposé.

Deuxième partie

Partitionnement, Elagage et Agrégation

Chapitre 2

LeafRank : Procédure d'Optimisation Locale de l'ASC

Dans ce chapitre, nous nous focalisons sur l'*étape d'optimisation locale de l'*ASC de la procédure TreeRank (cf Partie 1.5.2 du Chapitre 1), qui consiste à scinder un sous-ensemble C de l'espace d'entrée $\mathcal{X}$ en deux sous-ensembles non vides et disjoints, afin de maximiser l'ASC calculée sur C. Cette étape est primordiale dans l'algorithme TreeRank, puisque la construction d'un arbre d'ordonnancement $\mathcal{T}_D$, de profondeur $D \geq 1$, repose sur la résolution récursive de ce problème d'optimisation : à chaque itération $d \in \{0, \cdots, D-1\}$, toutes les cellules $C_{d,k}$, pour tout $k \in \{0, \cdots, 2^d - 1\}$, de la partition définie sur l'espace $\mathcal{X}$ par l'arbre courant $\mathcal{T}_d$ sont scindées en deux, pour maximiser l'ASC calculée localement sur chaque cellule. La résolution de cette étape d'optimisation est donc déterminante pour la performance de la fonction de score s_{2^D}, représentée par l'arbre d'ordonnancement $\mathcal{T}_D$.

Notre objectif est d'améliorer la performance globale de l'algorithme en proposant des stratégies de partitionnement plus flexibles que celle considérée dans le chapitre précédant. Rappelons que l'étape d'optimisation de l'heuristique TreeRank proposée dans le chapitre précédent consiste à scinder le sous-ensemble C en deux, perpendiculairement à l'un des axes de l'espace d'entrée $\mathcal{X}$. Dans tout ce chapitre, on notera $L^* \subset C$ et $R^* = C \setminus L^*$ les deux sous-ensembles de C, optimaux au sens de l'ASC *locale*, où L^* est le *noeud fils gauche* de C, *i.e.* le sous-ensemble contenant les *meilleures* observations de C.

Dans la première partie de ce chapitre, nous rappelons les enjeux de cette étape d'optimisation. Puis nous introduisons le principe général d'une procédure, que nous appellerons LeafRank, permettant de résoudre ce problème en scindant une cellule C de la partition définie sur $\mathcal{X}$ en sous-ensembles élémentaires, qui seront ensuite re-fusionnés pour constituer les sous-ensembles L^* et R^*. Dans la Partie 2.2, nous considérons le problème du choix de la méthode de partitionnement d'une cellule $C \subset \mathcal{X}$. Nous présentons notamment une première version de la procédure LeafRank, basée sur la définition d'une partition fixée *à priori* du sous-ensemble à scinder, pour laquelle nous établissons des résultats théoriques de convergence. Cependant, cette approche étant délicate à mettre en oeuvre en pratique lorsque l'on considère un espace d'entrée $\mathcal{X}$ de grande dimension, nous étudions la possibilité de scinder une cellule $C \subset \mathcal{X}$ de manière adaptative. Nous proposons alors une nouvelle interprétation de cette étape d'optimisation, élargissant considérablement les possibilités d'implémentation de la procédure LeafRank. En effet, nous montrons qu'elle peut être vue comme un problème de classification binaire pondérée. Ainsi, n'importe quel

algorithme de classification peut être mis en oeuvre pour scinder une cellule $C \subset \mathcal{X}$.

Nous concluons ce chapitre en proposant deux heuristiques dans la Partie 2.3. Dans la première, la construction d'une partition adaptative de $C \subset \mathcal{X}$, en sous-ensembles élémentaires, repose sur la mise en oeuvre d'une version pondérée de l'algorithme de classification CART introduit dans [Breiman *et al.* 1984]. Quant à la seconde heuristique, elle procède directement à la scission de C en deux sous-ensembles, estimés au moyen de Machines à Vecteurs Supports (SVM pour *Support Vector Machines*, [Vapnik 1996]). Après avoir rappelé le principe de ces deux méthodes de classification, nous discutons des avantages et inconvénients respectifs des deux heuristiques proposées. Afin d'illustrer notre propos, nous présentons également une étude empirique dans laquelle ces deux procédures sont mises en oeuvre sur des jeux de données simulées.

2.1 Procédure LeafRank : scinder pour mieux estimer

La méthode de scoring TreeRank, présentée dans le Chapitre 1, peut-être vue comme une extension des méthodes d'arbres de décision à la problématique d'ordonnancement. Cette approche consiste à scinder l'espace $\mathcal{X}$ de façon récursive, en maximisant l'ASC calculée localement sur les sous-ensembles de l'espace. Elle permet ainsi de construire une fonction de score constante par morceaux *quasi-optimale* [1] sur $\mathcal{X}$, définissant une partition *ordonnée* sur l'espace. La fonction de score obtenue est représentée graphiquement par un arbre d'ordonnancement $\mathcal{T}$, dont les feuilles représentent les cellules de la partition définie sur $\mathcal{X}$, l'ordre de celles-ci découlant directement de l'orientation de l'arbre.

Cependant, si les modèles d'arbres de décision sont particulièrement bien adaptés aux problèmes de classification ou de régression, leur application au problème d'ordonnancement est plus délicate. En effet, les performances des arbres de classification de type CART ([Breiman *et al.* 1984]) par exemple, reposent sur les propriétés locales de leurs feuilles, *i.e.* des cellules de la partition définie sur l'espace d'entrée $\mathcal{X}$. A contrario, le problème d'ordonnancement binaire est une tâche de nature *globale*, qui nécessite de comparer les observations de $\mathcal{X}$, les unes par rapport aux autres. Dans ce contexte, la performance d'une règle de prédiction dépend, non pas des propriétés locales des cellules de la partition définie sur $\mathcal{X}$, mais de l'ordonnancement de ces cellules. Nous considérons ci-dessous un exemple afin d'illustrer en quoi ceci constitue une difficulté pour la construction d'arbres d'ordonnancement.

Nous décrivons, sur la Figure 2.1 ci-dessous, le *cheminement* d'un petit échantillon d'observations le long d'un arbre d'ordonnancement $\mathcal{T}_D$, de profondeur $D = 3$. Dans chaque noeud de $\mathcal{T}_D$, les observations positives sont représentées par le symbole « + » et les négatives par le symbole « - ». Ce graphique montre clairement que si l'on commet des erreurs d'ordonnancement, lors du partitionnement du noeud racine par exemple, celles-ci vont se répercuter le long de $\mathcal{T}_D$, sans que l'on puisse y remédier. En effet, on voit bien que les observations positives du noeud racine, qui se retrouvent affectées au *noeud fils droit* à l'issue du premier partitionnement, échouent automatiquement dans la deuxième « moitié » du classement final, *i.e.* dans le meilleur des cas, la règle de prédiction leur attribuera le rang *médian* des observations. Ainsi, dans l'ordonnancement final, elles seront

1. *i.e.* qui converge vers une fonction de score $s^* \in \mathcal{S}^*$ optimale, au sens des normes $\mathcal{L}_1$ et $\mathcal{L}_\infty$ définies sur l'espace des courbes COR.

moins bien classées que les observations négatives affectées au *noeud fils gauche* à l'issue du premier partitionnement. Contrairement au cadre de la classification, l'augmentation de la profondeur D de l'arbre d'ordonnancement ne permet pas de corriger ces erreurs, qui continueront à se propager inexorablement le long de l'arbre.

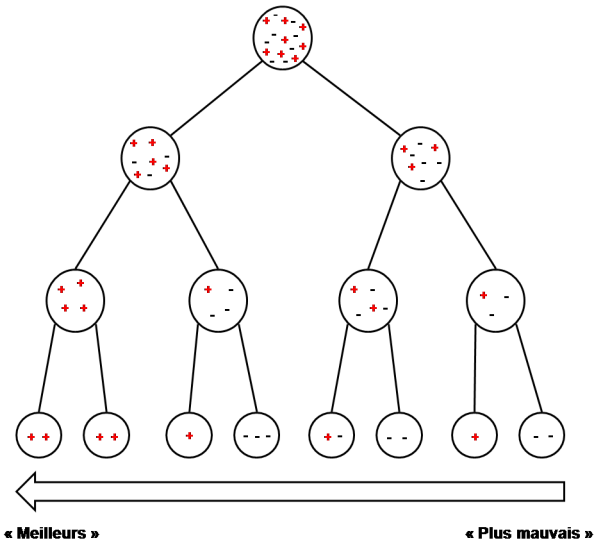


FIGURE 2.1 – Propagation des erreurs d'ordonnancement le long d'un arbre.

L'exemple traité dans la Partie 1.5.4.2 du Chapitre 1 met en évidence ce phénomène. Ces résultats empiriques montrent notamment que l'optimisation de la courbe COR est fortement *limitée* par l'estimation du premier point de la courbe, correspondant au partitionnement du noeud racine (voir Figures 1.15(c) et 1.15(d)). En effet, les erreurs commises à la première itération mènent à l'introduction d'un premier point dans l'estimateur, qui se positionne bien en-deçà de la courbe COR*. Or, ce point, qui contraint fortement l'ASC de l'estimateur, ne peut être amélioré, quel que soit le nombre d'itérations réalisées dans la procédure d'estimation. Il découle de tout ceci que la performance d'une fonction de score, représentée graphiquement par un arbre d'ordonnancement, dépend fortement de l'étape d'*optimisation locale de l'*ASC de la procédure TREERANK, que nous rappelons ci-dessous.

2.1.1 Etape d'optimisation de la procédure TREERANK

Soit C un sous-ensemble quelconque de l'espace d'entrée $\mathcal{X}$. L'étape d'optimisation de la procédure TREERANK consiste à définir une fonction de score binaire sur C de la forme

$$\forall\, x \in C,\ s_1^*(x) = \mathbb{I}\{x \in L^*\} - \mathbb{I}\{x \in C \setminus L^*\},$$

d'ASC maximale, en scindant C en deux sous-ensembles non vides disjoints $L^* \subset C$ et $R^* = C \setminus L^*$. La construction d'un arbre d'ordonnancement $\mathcal{T}_D$, de profondeur $D \geq 1$, repose sur la résolution récursive de ce problème d'optimisation.

A chaque itération $d \in \{0, \cdots, D-1\}$, les cellules $C_{d,k}$, pour tout $k \in \{0, \cdots, 2^d - 1\}$, sont scindées en deux de sorte à maximiser la *portion* d'ASC, calculée localement sur

chaque cellule $C_{d,k}$. Dans le chapitre précédent, nous avons vu que ce problème d'optimisation consistait à introduire un nouveau noeud dans l'estimateur affine par morceaux de la courbe COR* (voir la Figure 2.2 ci-dessous). Or, la Proposition 7 nous a permis d'établir un lien entre l'estimation d'un point de la courbe COR* et l'estimation d'un ensemble de niveau de la fonction de régression. On en déduit donc que le sous-ensemble L^* optimal correspond à un ensemble de niveau de la fonction de régression de la forme $\{x \in C \mid \eta(x) > u\}$, où $u \in]0,1[$.

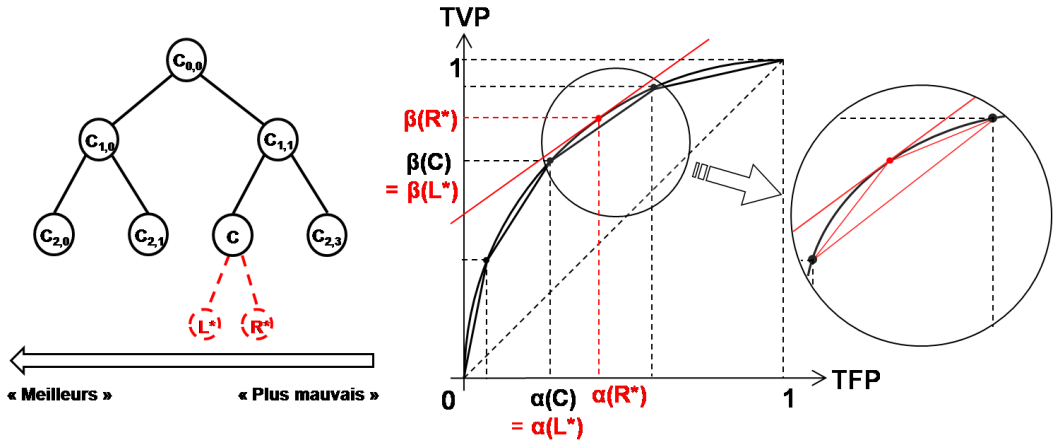


Figure 2.2 – Etape d'optimisation de la procédure TreeRank.

De plus, d'après le Lemme 5 ci-dessous, définissant les fonctions de score binaires optimales $s_1^* \in \mathcal{S}_1^*$, le sous-ensemble $L^* \subset C$ correspond précisément à l'ensemble de niveau p_C de la fonction de régression, où p_C est le taux théorique de positifs dans le sous-ensemble $C \times \{-1,+1\}$.

Lemme 5 *(Fonctions de score binaires optimales)*
Soit $C = \mathcal{X}$ et un couple de variables aléatoires $(X,Y) \in \mathcal{X} \times \{-1,+1\}$. On note $p_C = \mathbb{P}\{Y = +1 \mid X \in C\}$. On considère la fonction de score binaire $s_1^*(x) = \mathbb{I}\{x \in L^*\} - \mathbb{I}\{x \in C \setminus L^*\}$ où $L^* = \{x \in C \mid \eta(x) > p_C\}$. Soit $L \subset C$ un sous-ensemble quelconque mesurable, on pose $s = \mathbb{I}_L - \mathbb{I}_{C\setminus L}$. On a :

$$\mathrm{ASC(s)} = \frac{1}{2} + \frac{1}{2}\left(\beta(\mathrm{L}) - \alpha(\mathrm{L})\right) \leq \mathrm{ASC}(\mathrm{s}_1^*). \tag{2.1}$$

Plus précisément, l'identité suivante est vérifiée :

$$\mathrm{ASC}(\mathrm{s}_1^*) - \mathrm{ASC(s)} = \frac{1}{2\mathrm{p_C}(1-\mathrm{p_C})} \cdot \mathbb{E}[|\eta(\mathrm{X}) - \mathrm{p_C}| \cdot \mathbb{I}\{\mathrm{X} \in \mathrm{L}^* \Delta \mathrm{L}\}], \tag{2.2}$$

où Δ est la différence symétrique entre ensembles. De plus, on a

$$\mathrm{ASC}(\mathrm{s}_1^*) = \frac{1}{2\mathrm{p_C}(1-\mathrm{p_C})} \mathbb{E}\left[\max\{(1-\mathrm{p_C})\eta(\mathrm{X}),\ \mathrm{p_C}(1-\eta(\mathrm{X}))\}\right]. \tag{2.3}$$

Preuve 1 *(Preuve du Lemme 5) L'égalité (2.1) découle directement de l'application de la formule (1.42) de l'*ASC *associée à une fonction de score constante par morceaux donnée dans la Proposition 8 du Chapitre 1. A partir de cette expression et des notations*

introduites dans le lemme, on peut écrire :

$$\begin{aligned} p_C(1-p_C)\cdot(2\cdot \mathrm{ASC(s)}-1) &= p_C(1-p_C)\cdot(2\cdot(\frac{1}{2}+\frac{1}{2}\cdot(\beta(L)-\alpha(L)))-1) \\ &= p_C(1-p_C)\cdot(\beta(L)-\alpha(L)) \\ &= \mathbb{E}[(1-p_C)\cdot\eta(X)\cdot\mathbb{I}\{X\in L\} \\ &+ p_C\cdot(1-\eta(X))\cdot\mathbb{I}\{X\notin L\}] - p_C(1-p_C). \end{aligned}$$

Le lemme résulte alors du fait que

$$\begin{aligned} 2p_C(1-p_C)(\mathrm{ASC(s_1^*)}-\mathrm{ASC(s)}) &= \mathbb{E}\left[(1-p_C)\eta(X)\cdot(\mathbb{I}\{X\in L^*\}-\mathbb{I}\{X\in L\})\right] \\ &+ \mathbb{E}\left[p_C(1-\eta(X))\cdot(\mathbb{I}\{X\notin L^*\}-\mathbb{I}\{X\notin L\})\right] \\ &= \mathbb{E}\left[|\eta(X)-p_C|\cdot\mathbb{I}\{X\in L\Delta L*\}\right]. \end{aligned}$$

Remarque 8 *(Fonction de score binaire optimale versus classifieur de Bayes)*
Le résultat (2.1) montre aussi que l'étape d'optimisation de la procédure TreeRank *n'est pas un problème de classification binaire standard, qui consisterait à scinder le sous-ensemble C en deux, de sorte que $C = \{x \in C \mid \eta(x) > 1/2\} \cup \{x \in C \mid \eta(x) \leq 1/2\}$. En effet, on sait que dans ce cas-là (voir par exemple le Chapitre 4 de [Hastie et al. 2001] ou [Boucheron et al. 2005]), la fonction de score optimale s_1^* correspondrait au classifieur de Bayes, de la forme*

$$g^* : x \in C \to 2\cdot\mathbb{I}\{\eta(x) > 1/2\} - 1.$$

*Or, le Lemme 5 montre que, mise à part dans le cas particulier où les ensembles $\{x \in X \mid \eta(x) > 1/2\}$ et $\{x \in C \mid \eta(x) > p_C\}$ coïncident (à un ensemble μ-négligeable près), l'*ASC *associée au classifieur de Bayes est strictement inférieure à celle de la fonction de score binaire optimale s_1^*.*

La procédure TreeRank reposant sur l'implémentation récursive de cette étape d'optimisation, il apparaît clairement que les performances globales de l'algorithme dépendent de sa capacité à estimer correctement les ensembles de niveaux de la fonction de régression. Dans l'heuristique TreeRank proposée dans le chapitre précédent, ce problème est résolu en partitionnant le sous-ensemble C en deux, perpendiculairement à l'un de ses axes (cf Figure 2.3). Il semble évident que, mises à part dans certaines configurations spécifiques, il y a peu de chances pour que cette méthode de partitionnement simpliste puisse produire des sous-ensembles suffisamment *proches* -géométriquement parlant- des ensembles de niveau de la probabilité η. L'objectif de la procédure LeafRank, que nous décrivons ci-après, est de constituer une collection $\mathcal{C}$ de sous-ensembles $L \subset C$ candidats pour l'étape d'optimisation suffisamment *riche*, au sens où elle contient des sous-ensembles dont la géométrie est proche de celle des ensembles de niveaux de η, généralement inconnue.

2.1.2 La procédure LeafRank

Soit C un sous-ensemble quelconque de l'espace d'entrée $\mathcal{X}$ et $s_1 \in \mathcal{S}_1$ une fonction de score binaire définie sur C, de la forme

$$\forall\, x \in C,\ s_1(x) = \mathbb{I}\{x \in L\} - \mathbb{I}\{x \in C \setminus L\},$$

où L est un sous-ensemble de C et $R = C \setminus L$ son complémentaire. La Définition 9 permet d'expliciter la notion d'ASC *locale*, *i.e.* restreinte à un sous-ensemble C de $\mathcal{X}$.

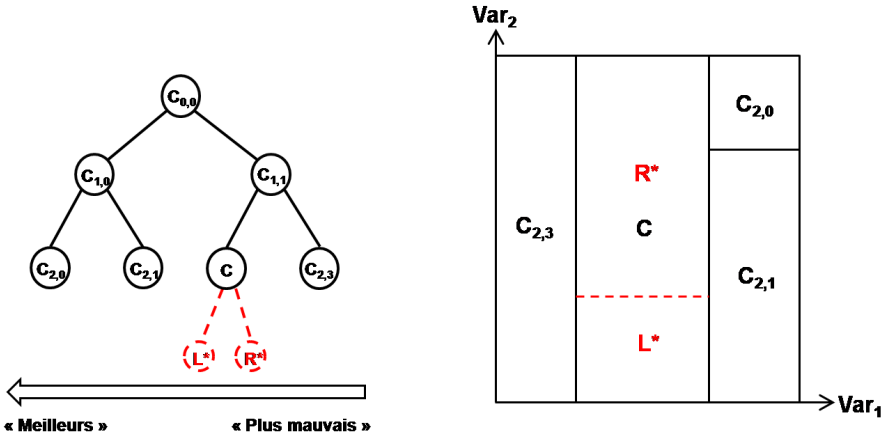


FIGURE 2.3 – Etape d'optimisation de l'heuristique TreeRank.

Définition 9 *(*ASC *locale)*
*Soit C un sous-ensemble quelconque de l'espace $\mathcal{X}$ et $s \in \mathcal{S}$ une fonction de score définie sur $\mathcal{X}$. L'*ASC *restreinte au sous-ensemble C, associée à la fonction s, s'écrit*

$$\begin{aligned} \mathrm{ASC_C(s)} &= \mathbb{P}\{s(X) > s(X') \mid (Y,Y') = (+1,-1), (X,X^{'}) \in C^2\} \\ &+ \frac{1}{2} \cdot \mathbb{P}\{s_1(X) = s_1(X') \mid (Y,Y') = (+1,-1), (X,X^{'}) \in C^2\}, \end{aligned} \tag{2.4}$$

pour tout couple (X,Y) et $(X^{'},Y^{'})$ i.i.d. de $\mathcal{X} \times \{-1,+1\}$.
Si de plus, on note $s_1 \in \mathcal{S}_1 : C \to \mathbb{R}$ la fonction de score binaire définie par (2.1.2), on a

$$\begin{aligned} \mathrm{ASC_C(s_1)} &= \frac{1}{2} + \frac{1}{2}\left(\frac{\beta(L)}{\beta(C)} - \frac{\alpha(L)}{\alpha(C)}\right) \\ &= \frac{1}{2} \cdot \left(1 + \frac{\alpha(C)\beta(L) - \alpha(L)\beta(C)}{\alpha(C)\beta(C)}\right), \end{aligned} \tag{2.5}$$

où L est un sous-ensemble de C.

L'objectif de la procédure LeafRank est de construire une fonction de score binaire s_1^* optimale, *i.e.* maximisant le critère $\mathrm{ASC_C}$. En d'autres termes, l'objectif est de résoudre le problème d'optimisation suivant :

$$\begin{aligned} \max_{s_1 \in \mathcal{S}_1} \mathrm{ASC_C(s_1)} &= \max_{L \in \mathcal{C}} \frac{1}{2} + \frac{1}{2}\left(\frac{\beta(L)}{\beta(C)} - \frac{\alpha(L)}{\alpha(C)}\right) \\ &\equiv \max_{L \in \mathcal{C}} (\alpha(C)\beta(L) - \alpha(L)\beta(C)), \qquad (2.6) \\ &\equiv \max_{L \in \mathcal{C}} (\beta(L) - \alpha(L)), \qquad (2.7) \end{aligned}$$

où $\mathcal{C}$ est une collection de sous-ensembles $L \subset C$ candidats. L'idée générale de la procédure LeafRank est de construire une collection $\mathcal{C}$ suffisamment riche, pour pouvoir résoudre efficacement le problème (2.6). Notre démarche consiste à définir une partition finie et disjointe $\mathcal{P}$ sur le sous-ensemble C, constituée de $J \geq 1$ cellules élémentaires $(C_j)_{1 \leq j \leq J}$, que l'on combine ensuite pour définir les sous-ensembles candidats $L \subset C$. On obtient ainsi une collection $\mathcal{C}$ de sous-ensembles de la forme $L = \cup_{j \in J' \subset J} C_j$.

Le partitionnement du sous-ensemble C par le biais de la procédure LeafRank procède en trois étapes. A la première étape, on définit une partition finie $\mathcal{P}$ sur C, constituée de $J \geq 1$ cellules élémentaires disjointes $(C_j)_j$.

La deuxième étape de la procédure consiste à ordonner les cellules élémentaires de $\mathcal{P}$ selon la décroissance de leur ratio $\beta(C_j)/\alpha(C_j)$, afin de garantir l'optimalité de la fonction de score binaire produite par la procédure. En effet, sur la base de la partition $\mathcal{P}$, on peut définir diverses fonctions de score $s_J \in \mathcal{S}_J$, constantes par morceaux à J segments. Or, d'après le Théorème 1, la fonction de score constante par morceaux quasi-optimale à J segments s_J^* est associée au couple $(\mathcal{P}, \sigma^*)$, où $\sigma^* \in \mathfrak{S}_J$ est la permutation de $\{1, \cdots, J\}$ ordonnant les cellules $(C_j)_j$ par $\beta(C_j)/\alpha(C_j)$ décroissant.

Enfin, la dernière étape de la procédure consiste à trouver la réunion des cellules élémentaires $(C_j)_j$ en deux sous-ensembles $L^* \subset C$ et $R^* = C \setminus L$, optimaux au sens de l'ASC_C. Notons que ce calcul est grandement facilité par l'ordonnancement préalable des cellules de la partition $\mathcal{P}$ de C, effectué à l'étape précédente. En effet, au lieu de considérer l'ensemble des combinaisons possibles des cellules élémentaires $(C_j)_j$, il suffit de trouver la fusion des cellules élémentaires $(C_j)_j$ *adjacentes*[2] permettant de se ramener aux sous-ensembles optimaux $L^* \subset C$ et $R^* = C \setminus L$. L'intégralité de ces étapes est détaillée dans la description de l'heuristique LeafRank donnée ci-après.

La Proposition 13 suivante montre que la fonction de score binaire $\hat{s}_1^* : C \to \mathbb{R}$, obtenue par l'application de l'heuristique LeafRank à l'échantillon $\mathcal{D}_n(C)$, est optimale au sens de l'ASC_C.

Proposition 13 *(Fonction de score binaire optimale)*
Soit $\mathcal{P} = \{C_j\}_{1 \leq j \leq J}$ une partition du sous-ensemble $C \subset \mathcal{X}$ et $\hat{s}_1^*(x) = \mathbb{I}\{x \in L^*\} - \mathbb{I}\{x \in R^*\}$ la fonction de score obtenue par la mise en oeuvre, sur un échantillon $\mathcal{D}_n(C)$ d'observations i.i.d., de l'heuristique LeafRank *basée sur la partition $\mathcal{P}$. Pour toute fonction de score binaire $s_1 = 2 \cdot \mathbb{I}_L + \mathbb{I}_{C \setminus L}$, où $L \subset C$ est une union de cellules de la partition $\mathcal{P}$, on a :*

$$\widehat{\mathrm{ASC}_\mathrm{C}}(s_1) \leq \widehat{\mathrm{ASC}_\mathrm{C}}(\hat{s}_1^*).$$

La preuve de cette proposition découle directement de l'application du Théorème 1 à la distribution empirique des observations $(X_i, Y_i)_{1 \leq i \leq n} \in C \times \{-1, +1\}$ de l'échantillon $\mathcal{D}_n C$.

2.2 Stratégies de partitionnement

La performance de la procédure LeafRank, que nous venons de décrire, repose clairement sur la *richesse* de la collection $\mathcal{C}$ de cellules candidates. Dans cette partie, nous considérons deux approches pour le partitionnement d'un sous-ensemble $C \subset \mathcal{X}$, dans le but de construire des collections $\mathcal{C}$ de cellules candidates suffisamment complexes et riches permettant d'estimer efficacement l'ensemble de niveau p_C de la fonction de régression. Dans un premier temps, nous proposons de définir une partition finie et disjointe $\mathcal{P}$ du sous-ensemble C, indépendamment des données contenues dans la cellule. Ainsi, nous présentons, dans la Partie 2.2.1 suivante, une première procédure LeafRank basée sur la définition à priori d'une partition *uniforme* sur C et étudions les propriétés de convergence

2. au sens où leurs scores selon la fonction s_J^* sont consécutifs

Algorithme LeafRank

1. (Initialisation.) Soit $C \subset \mathcal{X}$ un sous-ensemble contenant les données $\mathcal{D}_n(C) = \{(X_i, Y_i) : 1 \leq i \leq n, X_i \in C\}$.

2. (Construction de la collection $\mathcal{C}$.) On définit une partition $\mathcal{P}$ du sous-ensemble C, constituée de J éléments $\{C_1, \cdots, C_J\}$, avec $J \geq 1$.

3. (Permutation optimale.) on calcule la permutation $\sigma \in \mathfrak{S}_J$ telle que :

$$\frac{\widehat{\beta}(C_{\sigma(1)})}{\widehat{\alpha}(C_{\sigma(1)})} \geq \ldots \geq \frac{\widehat{\beta}(C_{\sigma(J)})}{\widehat{\alpha}(C_{\sigma(J)})}, \tag{2.8}$$

   où $\widehat{\alpha}(.)$ et $\widehat{\beta}(.)$ sont les taux empiriques de faux et vrais positifs calculés sur l'échantillon $\mathcal{D}_n(C)$.

4. (Etape d'optimisation.) $\forall j \in \{1, \ldots, J\}$, on pose $L_j = \bigcup_{l \leq j} C_{\sigma(l)}$ et on calcule l'entropie $\widehat{\Lambda}(j) = \widehat{\beta}(L_j) - \widehat{\alpha}(L_j)$. On pose

$$j^* = \underset{1 \leq j \leq J}{\arg\max} \left\{ \widehat{\Lambda}(j) \right\}.$$

5. (Sortie.) On forme les sous-ensembles :

$$\widehat{L}^* = L_{j^*} \text{ et } \widehat{R}^* = C \setminus L^*.$$

de la fonction de score binaire ainsi obtenue. Puis, dans la Partie 2.2.2, nous considérons le problème du partitionnement de la cellule C de manière adaptative, en nous appuyant sur l'interprétation de l'étape d'optimisation de la procédure TreeRank comme un problème de classification binaire pondérée.

2.2.1 Partition fixée à priori : un premier pas vers la flexibilité

Afin de simplifier les calculs, nous allons nous placer dans le cas où $\mathcal{X} = C = [0,1]^q$ et considérer le cas de la première itération. On note $\mathcal{P}(m)$ la partition de C, constituée de cubes dyadiques de côté 2^{-m}, *i.e.* de sous-ensembles de la forme

$$\prod_{l=1}^{q} \left[\frac{k_l}{2^m}, \frac{k_l + 1}{2^m} \right[\text{ où } 0 \leq k_l < 2^m \text{ pour tout } l \in \{1, \ldots, q\}.$$

On note $\#\mathcal{P}(m) = 2^{mq}$ le cardinal de la partition $\mathcal{P}(m)$. De plus, on pose $\widehat{L}_m = \widehat{L}^*$ et $\widehat{R}_m = \widehat{R}^*$ les sous-ensembles obtenus par l'heuristique LeafRank, basée sur la partition $\mathcal{P}(m)$, appliquée à l'échantillon $\mathcal{D}_n(C)$ et

$$\forall\, x \in C,\ \hat{s}_1^*(x) = \mathbb{I}\{x \in \widehat{L}_m\} - \mathbb{I}\{x \in \widehat{R}_m\},$$

la fonction de score binaire définie sur C, associée à ces sous-ensembles. Grâce à cette collection de cubes élémentaires, on s'attend à pouvoir estimer de façon précise l'ensemble de niveau $\{x \in C :\ \eta(x) \geq p_C\}$, à condition toutefois que les bornes de l'ensemble soient suffisamment lisses et que la longueur du côté des cubes 2^{-m} soit choisie suffisamment petite. Le Théorème 6 suivant permet de formaliser cette intuition.

Théorème 6 *(Partition dyadique)*
Pour tout $m \geq 1$, on note $\mathcal{P}_{2,m}$ la collection des partitions de C constituées de deux sous-ensembles non vides, obtenus en fusionnant des cubes dyadiques de côté 2^{-m}. Supposons de plus que $p_C \in [\underline{p_C},\ \bar{p_C}]$ où $0 < \underline{p_C} < \bar{p_C} < 1$. Il existe alors une constante $c < \infty$, dépendant de $\underline{p_C}$ et $\bar{p_C}$, telle que pour tout $m \geq 1$ et pour $n \geq 1$ suffisamment grand, on a, pour tout $\epsilon \in]0,1[$, avec une probabilité d'au moins $(1-\epsilon)$:

$$\mathrm{ASC_C}(\mathrm{s}_1^*) - \mathrm{ASC_C}(\widehat{\mathrm{s}}_1^*) \leq \mathrm{c} \cdot \frac{2^{\mathrm{mq}}}{\sqrt{\mathrm{n}}} + \left\{ \mathrm{ASC_C}(\mathrm{s}_1^*) - \max_{\mathrm{s} \in \mathcal{S}_{\mathcal{P}_{2,\mathrm{m}}}} \mathrm{ASC_C}(\mathrm{s}) \right\}. \tag{2.9}$$

Preuve 2 *(Preuve du Théorème 6)*

Pour tout $m \geq 1$, soit $\mathcal{C}_m$ la collection de sous-ensembles (non vides) de C obtenue à partir des 2^{mq} cubes dyadiques de côté 2^{-m}, l'ensemble $C = [0,1]^q$ étant exclu. On note $\mathcal{P}_{2,m}$ l'ensemble des partitions de C constituées de deux éléments non vides de $\mathcal{C}_m$. On pose : $\forall m \geq 1$, $\widetilde{L}_m^*$ la cellule fille gauche optimale pouvant être obtenue à partir de la collection $\mathcal{C}_m$ et $\widehat{L}_m^*$ sa contrepartie empirique. Les fonctions de score binaires associées sont notées :

$$\tilde{s}_1^*(x) = 2 \cdot \mathbb{I}\{x \in \widetilde{L}_m^*\} - 1 \text{ et } \widehat{s}_m^*(x) = 2 \cdot \mathbb{I}\{x \in \widehat{L}_m^*\} - 1.$$

De manière classique, on borne le déficit en termes d'$\mathrm{ASC_C}$ *par la somme d'un biais et d'une variance :*

$$\begin{aligned}
\mathrm{ASC_C}(\mathrm{s}_1^*) - \mathrm{ASC_C}(\widehat{\mathrm{s}}_1^*) &= \{\mathrm{ASC_C}(\mathrm{s}_1^*) - \mathrm{ASC_C}(\widetilde{\mathrm{s}}_1^*)\} + \{\mathrm{ASC_C}(\widetilde{\mathrm{s}}_1^*) - \widehat{\mathrm{ASC_C}}(\widetilde{\mathrm{s}}_1^*)\} \\
&+ \{\widehat{\mathrm{ASC_C}}(\widetilde{s}_1^*) - \widehat{\mathrm{ASC_C}}(\widehat{s}_1^*)\} + \{\widehat{\mathrm{ASC_C}}(\widehat{s}_1^*) - \mathrm{ASC_C}(\widehat{\mathrm{s}}_1^*)\} \\
&\leq \{\mathrm{ASC_C}(\mathrm{s}_1^*) - \mathrm{ASC_C}(\widetilde{\mathrm{s}}_1^*)\} + \{\mathrm{ASC_C}(\widetilde{\mathrm{s}}_1^*) - \widehat{\mathrm{ASC_C}}(\widetilde{\mathrm{s}}_1^*)\} \\
&+ \{\widehat{\mathrm{ASC_C}}(\widehat{s}_1^*) - \mathrm{ASC_C}(\widehat{\mathrm{s}}_1^*)\} \\
&\leq \mathrm{ASC_C}(\mathrm{s}_1^*) - \mathrm{ASC_C}(\widetilde{\mathrm{s}}_1^*) + 2 \sup_{\mathrm{s} \in \mathcal{S}_{\mathcal{P}_{2,\mathrm{m}}}} |\widehat{\mathrm{ASC_C}}(\mathrm{s}) - \mathrm{ASC_C}(\mathrm{s})| \ .
\end{aligned}$$

Si l'on considère tout d'abord le terme de variance, on peut exprimer l'$\widehat{\mathrm{ASC_C}}(s)$ *empirique comme suit :*

$$\widehat{\mathrm{ASC_C}}(s) = \frac{n(n-1)}{2n_+n_-}\widehat{U}_n(s) \ ,$$

où

$$\forall\, ((X_i,Y_i),(X_j,Y_j)) \in C^2,\ \widehat{U}_n(s) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} h_s((X_i,Y_i),(X_j,Y_j))$$

est une U-statistique d'ordre 2, de noyau symétrique borné

$$h_s((x_1,y_1),(x_2,y_2)) = \mathbb{I}\{(y_1-y_2)(s(x_1)-s(x_2)) > 0\} + \frac{1}{2}\mathbb{I}\{s(x_1) = s(x_2),\ y_1 \neq y_2\}$$

et d'espérance $U(s) = 2p_C(1-p_C)\mathrm{ASC_C}(\mathrm{s})$. En appliquant l'inégalité exponentielle de Hoeffding pour les U-statistiques (Théorème A de la section 5.6 de [Serfling 1980]), que

l'on combine avec la borne d'union, on obtient, pour tout $\epsilon \in]0,1[$ *et avec une probabilité supérieure à* $(1-\epsilon)$ *:*

$$\forall n \geq 1, \sup_{s \in \mathcal{S}_{\mathcal{P}_{2,j}}} \left| \widehat{U}_n(s) - U(s) \right| \leq \sqrt{\frac{\log(\epsilon/(2\#\mathcal{P}_{2,m}))}{2n}}.$$

Pour obtenir la borne du Théorème, il suffit de remarquer que

$$\left| \widehat{\mathrm{ASC}_{\mathrm{C}}}(s) - \mathrm{ASC_C(s)} \right| \leq \frac{1}{2\underline{p_C}(1-\bar{p_C})} \left| \widehat{U}_n(s) - U(s) \right| + \frac{1}{2} \left\{ \left| \frac{1}{p_C} - \frac{n}{n_+} \right| + \left| \frac{1}{1-p_C} - \frac{n}{n-n_+} \right| \right\}$$

et d'appliquer l'inégalité probabiliste de Hoeffding afin de contrôler les fluctuations de n_+/n *autour de* $p_C \in [\underline{p_C}, \bar{p_C}]$*.*

Remarque 9 *(Partition stable par union)*
Notons que, par construction, la collection $\mathcal{P}_{2,m}$ *est stable par union. Ainsi, d'après la Proposition 12, quand l'étape d'optimisation de l'algorithme* TreeRank *est résolue par le biais de l'heuristique* LeafRank*, basée sur la partition* $\mathcal{P}_{2,m}$*, la courbe* $\widehat{\mathrm{COR}}(s_D, .)$*,* $D \geq 1$*, empirique produite par* TreeRank *est concave.*

On peut contrôler le terme de biais sous certaines hypothèses de régularité sur l'ensemble de niveau $L^* = \{x \in C : \ \eta(x) > p_C\}$. En effet, dans le cas où la densité de μ est bornée par rapport à la mesure de Lebesgue λ sur $\mathbb{R}^d$, le Lemme 5 nous donne :

$$\mathrm{ASC_C(s_1^*) - ASC_C(s_1)} \leq \frac{||\mathrm{d}\mu/\mathrm{dx}||_\infty}{2\mathrm{p_C(1-p_C)}} \cdot \lambda(\mathrm{L^* \Delta L}),$$

pour toute fonction de score $s_1 = 2 \cdot \mathbb{I}_L + \mathbb{I}_{C \setminus L}$ où $L \in \mathcal{P}_{2,m}$.

Par ailleurs, quand les variations de η sont bornées, la frontière ∂L^* correspond à l'ensemble $\partial L^* = \{x \in C : \ \eta(x) = p_C\}$, du fait de la continuité de la probabilité à posteriori. La frontière ∂L^* étant alors de périmètre fini $per(\partial L^*) < \infty$, on peut borner le terme de biais par

$$\min_{L \in \mathcal{P}_{2,m}} \lambda(L^* \Delta L) \leq c \cdot per(\partial L^*) 2^{-mq},$$

pour une constante $c < \infty$ (voir Proposition 9.7 dans [Mallat 1990]). Dans ce cas, on peut obtenir une borne d'ordre $n^{-1/4}$ dans l'expression (2.9), en choisissant un niveau de résolution $m = m(n)$, tel que $2^{m(n)} \sim n^{1/(4q)}$ quand $n \to \infty$.

Comme cela a été montré dans [Clémençon & Vayatis 2009c], on peut encore améliorer les bornes généralisées, sous des hypothèses plus restrictives impliquant un paramètre de régularité θ de ∂L^*, comme par exemple la *taille des cubes élémentaires*. Mais, il faut noter que dans ce cas, le choix d'une valeur optimale de m dépend du paramètre θ. Une approche classique, pour atteindre un taux de convergence optimal, consiste alors à procéder à une sélection de modèle en ajoutant un terme de pénalité bien choisi au critère de l'ASC empirique. Nous renvoyons à la référence [Clémençon & Vayatis 2009c] pour plus de détails sur le processus de sélection de modèle, dans le cas spécifique où l'estimation de règles de score repose sur la définition à priori d'une partition uniforme sur $\mathcal{X}$ et où la régularité de la fonction de régression est prise en compte au moyen d'un paramètre θ. Par ailleurs, la question de la sélection de modèles, produits via la procédure TreeRank, est abordée dans un cadre plus général dans le Chapitre 3.

Enfin, tout comme dans le contexte de la classification, on peut atteindre des taux de convergence plus rapides, à la seule différence qu'ici, la complexité du problème est liée au comportement de la fonction de régression autour de la valeur p_C et non pas $1/2$ (voir la Remarque 8). Sous l'extension suivante de la condition de bruit de Massart ([Massart 2007a]), stipulant qu'il existe une constante $c > 0$ telle que l'on a presque sûrement

$$\forall\, X \in X,\, |\eta(X) - p_C| \geq c\ ,$$

on peut obtenir un taux de convergence d'ordre $O(n^{-1})$. Ce résultat peut être prouvé de la même façon que dans le contexte de la classification binaire. Il repose sur des résultats de concentration relatifs à la variance du terme quantifiant le déficit en termes $\mathrm{ASC_C}$. Notons cependant que cette condition de bruit est incompatible avec l'hypothèse de régularité de la courbe COR^* découlant des hypothèses $(\mathbf{A_1})$ et $(\mathbf{A_2})$ sur la fonction de régression et sur les distributions conditionnelles G^* et H^* (cf Partie 1.1). En effet, cette condition implique que G^* et H^* présentent toutes deux un saut en p_C. Toutefois, il est possible d'affaiblir cette hypothèse en considérant une version modifiée de la condition de bruit de Tsybakov ([Mammen & Tsybakov 1999] et [Tsybakov 2004]) :

$$\mathbb{P}\left\{|\eta(X) - p_C| \leq t\right\} \leq M \cdot t^{\frac{a}{1-a}}$$

pour une valeur $a \in [0, 1]$. D'après l'argumentation présentée dans [Tsybakov 2004], cette condition conduit à un taux de convergence de l'ordre de $n^{1/(2-a)}$. Remarquons que cette condition peut être reformulée comme suit :

$$F^*(p_C + t) - F^*(p_C - t) \leq M \cdot t^{\frac{a}{1-a}},$$

où $F^* = p_C G^* + (1 - p_C)H^*$ représente la fonction de répartition de $\eta(X)$ restreinte au sous-ensemble C. De ce fait, si l'on suppose que G^* et H^* sont différentiables, de dérivées bornées, et que $H^{*\prime} > 0$, on a nécessairement $a = 1/2$ et l'on obtient un taux de convergence d'ordre $n^{-2/3}$.

Ainsi, on peut obtenir de nombreux résultats théoriques pour la procédure LeafRank basée sur une partition $\mathcal{P}$ uniforme, fixée à priori sur le sous-ensemble C, de manière indépendante aux données. Cependant, la mise en oeuvre de cette procédure devient délicate en pratique, dès que la dimension q du sous-ensemble $C \subset \mathcal{X}$ devient grande, son implémentation devenant complexe et très coûteuse en temps de calcul. De plus, il nous semble important de tenir compte de la distribution des données pour définir une partition plus pertinente de la cellule $C \subset \mathcal{X}$ considérée. Dans la partie suivante, nous montrons que l'étape d'optimisation de la procédure TreeRank revient à résoudre un problème de classification binaire pondérée, ce qui nous permet de considérer le problème du partitionnement du sous-ensemble $C \subset \mathcal{X}$ de manière adaptative.

2.2.2 Un problème de classification binaire pondérée

Dans la première partie de ce chapitre, nous avons vu que, dans le cas précis où les sous-ensembles $\{x \in C \mid \eta(x) > p_C\}$ et $\{x \in C \mid \eta(x) > 1/2\}$ sont identiques, l'étape d'optimisation de la procédure TreeRank, qui consiste à estimer une fonction de score binaire quasi-optimale s_1^* sur un sous-ensemble $C \subset \mathcal{X}$, correspond à un problème de classification binaire *standard*. Dans cette partie, nous allons montrer que, même dans le cas général où $p_C \neq 1/2$, le problème de l'estimation de l'ensemble de niveau p_C de la fonction

de régression peut être vu comme un problème de classification binaire pondérée par un coût asymétrique dépendant des données.

Soit L un sous-ensemble quelconque de C, on définit l'erreur de classification pondérée sur C suivante :

$$\mathcal{L}_{C,\varpi}(L) = 2p_C(1-\varpi)\ (1-\beta(L)) + 2(1-p_C)\varpi\ \alpha(L), \tag{2.10}$$

où $\varpi \in]0,1[$ est un coefficient asymétrique dépendant des données. La contrepartie empirique de cette grandeur est donnée par :

$$\widehat{\mathcal{L}}_{C,\varpi}(L) = \frac{2\varpi}{n}\sum_{i=1}^{n}\mathbb{I}\{Y_i=-1,\ X_i\in L\} + \frac{2(1-\varpi)}{n}\sum_{i=1}^{n}\mathbb{I}\{Y_i=+1,\ X_i\notin L\}. \tag{2.11}$$

La Proposition 14 suivante a été introduite initialement dans [Clémençon & Vayatis 2008b]. Elle identifie la solution optimale du problème de classification pondérée, qui consiste à minimiser l'erreur empirique définie par l'égalité (2.11).

Proposition 14 *(Proposition 15 de [Clémençon & Vayatis 2008b])*
Le sous-ensemble de C optimal pour la mesure d'erreur $\mathcal{L}^*_{C,\varpi}$, définie par l'égalité (2.10), est donné par $L^*_\varpi = \{x \in C \mid \eta(x) > \varpi\}$. En effet, pour tout $L \subset C$, on a :

$$\mathcal{L}_{C,\varpi}(L^*_\varpi) \leq \mathcal{L}_{C,\varpi}(L).$$

Plus précisément, l'excès de risque pour un sous-ensemble L quelconque de C est donné par :

$$\mathcal{L}_{C,\varpi}(L) - \mathcal{L}_{C,\varpi}(L^*_\varpi) = 2\mathbb{E}\left[|\eta(X)-\varpi|\cdot\mathbb{I}\{X\in L\Delta L^*_\varpi\}\right].$$

L'erreur optimale, pour tout $X \in C$, est donnée par :

$$\mathcal{L}_{C,\varpi}(L^*_\varpi) = 2\mathbb{E}[\min\{\varpi(1-\eta(X)),(1-\varpi)\eta(X)\}].$$

Cette proposition montre en particulier que quand on prend $\varpi = p_C$, le sous-ensemble optimal de C est donné par $L^* = \{x \in C \mid \ \eta(x) > p_C\}$. De plus, on souligne que dans ce cas particulier, l'erreur de classification pondérée s'exprime en fonction de l'$\mathrm{ASC_C}$ comme suit :

$$\mathcal{L}_{C,p_C}(L) = 4p_C(1-p_C)(1-\mathrm{ASC_C}(\mathrm{s}_1)), \tag{2.12}$$

où $s_1(x) = \mathbb{I}\{x \in L\} - \mathbb{I}\{x \in C \setminus L\}$.

Notons de plus que, le taux théorique de positifs p_C étant inconnu, il faut l'estimer à partir des données disponibles. La contrepartie empirique de l'erreur de classification pondérée $\mathcal{L}_{C,p_C}$ s'écrit donc en fonction de son estimateur $\widehat{p_C} = n_+/n$, où n_+ est le nombre d'observations positives dans l'échantillon $\mathcal{D}_n(C)$:

$$\widehat{\mathcal{L}}_{C,\widehat{p_C}}(L) = 4\widehat{p_C}(1-\widehat{p_C})(1-\widehat{\mathrm{ASC}}_C(s_1)).$$

Ainsi, l'heuristique TreeRank peut être reformulée comme une *imbrication* de problèmes de classification binaire pondérée. En effet, d'après le résultat précédent, la procédure qui consiste à maximiser récursivement l'ASC empirique, calculée localement sur chaque cellule $C \subset \mathcal{X}$, sur l'ensemble des fonctions de score binaires de la forme $\{\mathbb{I}_L - \mathbb{I}_{C\setminus L} : L \in \mathcal{C}\}$ revient à minimiser récursivement, pour chaque cellule C, le risque empirique de classification pondérée par $\varpi(C) = \widehat{p_C}$, le taux empirique de positifs dans C, sur une classe

ALGORITHME TREERANK : MINIMISATION DU RISQUE EMPIRIQUE PONDÉRÉ

1. (**Initialisation.**) On considère l'ensemble $C = \mathcal{X}$ et on pose $C_{0,0} = \mathcal{D}_n$.

2. (**Itérations.**) Pour $d = 0, \ldots, D-1$ et $k = 0, \ldots, 2^d - 1$:

   (a) (*C*oefficient d'assymétrie.) Calculer le taux de positifs dans la cellule $C_{d,k}$ de cardinal $n_{C_{d,k}}$:

   $$\varpi(C_{d,k}) = \frac{1}{n_{C_{d,k}}} \sum_{i=1}^{n_{C_{d,k}}} \mathbb{I}\{X_i \in C_{d,k},\ Y_i = +1\}.$$

   (b) (*M*inimisation du risque empirique pondéré.) Résoudre le problème de classification pondérée relatif à la cellule $C_{d,k}$, au coefficient d'asymétrie $\varpi = \varpi(C_{d,k})$ et à l'échantillon $\mathcal{D} = \mathcal{D}_n \cap (C_{d,k} \times \{-1, +1\})$:

   $$\widehat{L}^* = \operatorname*{arg\,min}_{L \subset C_{d,k}} \widehat{\mathcal{L}}_{C_{d,k},\varpi}(L).$$

   (c) Poser $C_{d+1,2k} = \widehat{L}^*$ et $C_{d+1,2k+1} = C_{d,k} \setminus C_{d+1,2k}$.

3. **Sortie.** Arbre binaire orienté $\mathcal{T}_D = \{C_{d,k} :\ 0 \leq d \leq D,\ 0 \leq k \leq 2^d - 1\}$.

$\mathcal{C}$ de sous-ensembles candidats. Le détail des étapes de cette heuristique, appliquée à un échantillon $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$ de copies *i.i.d.* du couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$, est donné ci-dessous.

Cette interprétation de l'étape d'optimisation comme un problème de classification binaire pondérée permet d'envisager de nombreuses pistes pour l'implémentation pratique de la procédure LEAFRANK. En effet, on peut désormais mettre en oeuvre n'importe quel algorithme de classification binaire pour maximiser l'ASC locale à chaque itération de l'algorithme TREERANK. On peut par exemple envisager de construire une collection $\mathcal{C}$ de cellules, candidates pour la scission d'un sous-ensemble $C \subset \mathcal{X}$, de manière adaptative en appliquant aux observations de C une version pondérée de l'algorithme de classification CART ([Breiman *et al.* 1984]) ou une approche de type *k-plus proches voisins* (voir par exemple [Gyorfi *et al.* 2002]). En outre, cette nouvelle approche nous permet aussi de considérer la scission d'une cellule $C \subset \mathcal{X}$ directement en deux sous-ensembles, sans passer par la définition d'une partition sur C, en utilisant des algorithmes de classification plus flexibles comme les Machines à Vecteurs Supports ou d'autres méthodes à noyau (voir par exemple [Hastie *et al.* 2001]).

2.3 Deux exemples d'implémentation

La performance de l'algorithme TREERANK repose sur la capacité de la procédure LEAFRANK à estimer efficacement les ensembles de niveaux de la probabilité η, dont la géométrie est généralement inconnue à priori. Il convient donc en pratique d'utiliser une

procédure d'optimisation suffisamment flexible ou d'en essayer plusieurs afin de choisir la plus adaptée au problème considéré. Dans cette partie, nous proposons deux heuristiques LeafRank reposant respectivement sur l'algorithme de classification CART et sur la mise en oeuvre de SVM. Nous discutons des avantages de chacune de ces approches et présentons une étude empirique illustrant leurs performances sur des jeux de données simulés.

2.3.1 Une version pondérée de l'algorithme de classification CART

Rappelons que notre objectif est de construire une collection $\mathcal{C}$ de cellules suffisamment riche pour estimer l'ensemble de niveau p_C de la fonction de régression, où $C \subset \mathcal{X}$. Afin que cette collection soit la plus pertinente possible et tienne compte des données observées, nous cherchons à définir une partition $\mathcal{P}$ sur C de manière adaptative. Or dans la méthode CART, l'estimation d'une règle de classification $s_2 \in \mathcal{S}_2$ sur un espace $\mathcal{X}$, de la forme $s_2(x) = \mathbb{I}\{x \in L\} - \mathbb{I}\{x \in C \setminus L\}$, pour tout $x \in \mathcal{X}$ et $L \subset C$, repose sur la construction d'une partition adaptative $\mathcal{P}$ sur cet espace. Il apparaît clairement que la mise en oeuvre récursive de cet algorithme de classification à chaque itération de la procédure TreeRank permettra de produire des collections $\mathcal{C}$ de cellules candidates bien plus riches que dans la première version de l'algorithme, détaillée dans le Chapitre 1. Mais nous allons voir que le principal avantage de cette approche réside dans l'*interprétabilité* quelle confère à la règle d'ordonnancement.

2.3.1.1 L'algorithme CART

L'algorithme de classification CART, proposé dans [Breiman *et al.* 1984], procède à la scission récursive et dyadique de l'espace $\mathcal{X} \subset \mathbb{R}^q$, $q \geq 1$, perpendiculairement à ses axes. A chaque itération, on cherche la *meilleure* variable et le *meilleur* seuil associé pour scinder un sous-ensemble $C \subset \mathcal{X}$, de sorte à minimiser une mesure $\gamma(s_2)$ de l'impureté des cellules de la partition induite par s_2 sur $\mathcal{X}$. Parmi les critères d'impureté les plus fréquemment utilisés on peut citer :

- l'*erreur de classification* : $\gamma(s_2) = 1 - \max(p, 1-p)$,
- l'*indice de Gini* : $\gamma(s_2) = 2p(1-p)$,
- la *déviation* : $\gamma(s_2) = -p\log(p) - (1-p)\log(1-p)$,

où p est le taux théorique de positifs dans l'espace $\mathcal{X} \times \{-1, +1\}$.

Soit un couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$, on note $X = (X^{(1)}, \cdots, X^{(q)})$, où pour tout $i \in \{1, \cdots, q\}$, $X^{(i)}$ représente la $i^{\text{ème}}$ composante de X, ou encore la $i^{\text{ème}}$ *variable*. Avec ces notations, on peut expliciter la règle de partitionnement d'une cellule $C \subset \mathcal{X}$, qui pour chaque itération de l'algorithme CART est de la forme $X^{(i)} \leq c_j$ ou son complémentaire $X^{(i)} > c_j$, si la variable $X^{(i)}$ est continue, et de la forme $X^{(i)} = c_j$ ou son complémentaire $X^{(i)} \neq c_j$ s'il s'agit d'une variable catégorielle. Ainsi, chaque cellule de la partition $\mathcal{P}$ de $\mathcal{X}$ finalement obtenue est caractérisée par des intersections de règles simples de ce type. Dans le cas particulier où les q composantes du problème sont de nature continue, ces

règles seront de la forme

$$\cap_{i\in\mathcal{I}\subset\{1,\cdots,q\}}\{X^{(i)} \leq c_m^i\} \bigcap \cap_{j\in\mathcal{J}\subset\{1,\cdots,q\}}\{X^{(j)} > c_m^j\}, \tag{2.13}$$

où pour tout $m \in \mathbb{N}$, c_m^i représente la $m^{\text{ème}}$ valeur seuil de la variable $X^{(i)}$.

Une propriété intéressante de la méthode CART réside dans le fait que la règle de prédiction estimée peut être représentée sous la forme d'un arbre binaire, que l'on appellera *arbre de classification*. Notons $\mathcal{T}$ l'arbre de classification représenté sur la Figure 2.4 ci-dessous. De la même façon que dans le cadre des *arbres d'ordonnancement* définis dans le Chapitre 1, un arbre de classification est constitué de *noeuds* représentant des sous-ensembles de l'espace d'entrée $\mathcal{X}$. En particulier, la *racine* de $\mathcal{T}$ représente l'espace $\mathcal{X}$ dans son ensemble. Chaque *noeud interne* C de $\mathcal{T}$ engendre deux *enfants*, qui correspondent aux deux *sous-rectangles* disjoints de C, optimaux au sens du critère d'impureté choisi.

Pour tout noeud interne C, on peut définir une *branche* comme étant un *sous-arbre* de $\mathcal{T}$, que l'on notera $\mathcal{T}(C)$, de racine C et contenant tous les descendants de C. Enfin, on appelle *feuilles terminales* les noeuds de $\mathcal{T}$ n'ayant pas été scindés. Ces feuilles représentent les cellules de la partition $\mathcal{P}$ définie sur $\mathcal{X}$ par la mise en oeuvre de l'algorithme CART sur un échantillon d'apprentissage $\mathcal{D}_n$, constitué de n copies *i.i.d.* du couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$. Elles caractérisent entièrement la règle de prédiction $s_{\mathcal{T}} \in \mathcal{S}_2$, représentée par l'arbre $\mathcal{T}$, qui leur attribue le label binaire majoritairement représenté dans chacune d'entre elles. Ainsi, pour prédire l'étiquette binaire associée à une nouvelle observation $X' \in \mathcal{X}$, indépendante des observations $(X_i)_{1\leq i\leq n}$ de l'échantillon d'apprentissage $\mathcal{D}_n$ et identiquement distribuée, il suffit de lui faire *parcourir* l'arbre de classification de la racine jusqu'à l'une de ses feuilles, selon les règles d'affectation telles que celles définies par l'expression (2.13), et de lui attribuer l'étiquette binaire associée à la feuille terminale dans laquelle elle se trouve.

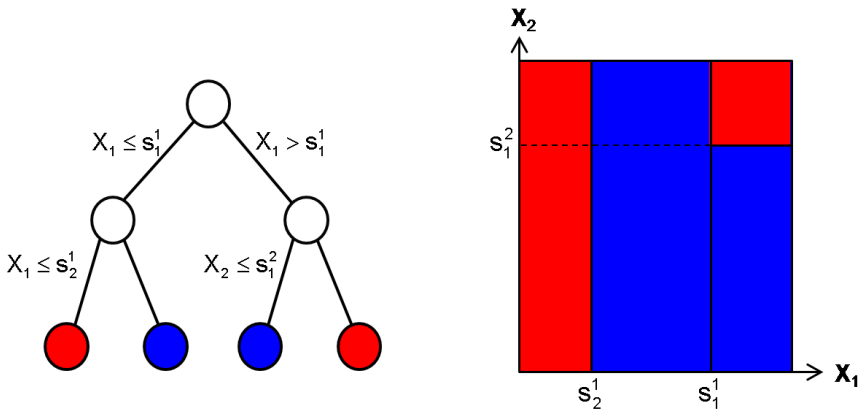


FIGURE 2.4 – Algorithme de classification CART.

Enfin, dans le cas de l'utilisation d'une procédure récursive, telle que l'algorithme CART, il est important de contrôler la complexité du modèle obtenu en déterminant le nombre d'itérations nécessaire et suffisant pour garantir les bonnes performances de la

règle de prédiction. De plus, il semble plus pertinent de choisir ce niveau de complexité de manière adaptative, en fonction des observations disponibles. Une possibilité est de poursuivre la procédure d'apprentissage jusqu'à ce qu'un critère d'arrêt, défini au préalable, soit satisfait. En règle générale, on limite la croissance de l'arbre de classification en fixant un gain minimal pour le critère d'impureté $\gamma(s_{\mathcal{T}})$ ou en définissant la taille minimale en deçà de laquelle un noeud ne peut plus être scindé. Cependant, ces deux méthodes nécessitent de paramétrer manuellement l'algorithme *à priori* et ne garantissent en aucun cas l'*optimalité* du classifieur obtenu.

Dans [Breiman *et al.* 1984], les auteurs proposent une approche différente, qui consiste à parcourir un arbre $\mathcal{T}$ de ses feuilles jusqu'à sa racine et à *élaguer* ses branches, afin de minimiser l'impureté $\gamma(s_{\mathcal{T}})$ pénalisée par une mesure de la complexité du modèle. Lorsque la sélection de modèle est effectuée au moyen de cette *procédure d'élagage*, la construction d'un arbre de classification procède en 3 étapes, que nous décrivons brièvement dans l'encadré ci-dessous.

Algorithme Cart : construction d'un arbre élagué

(**Initialisation.**) Soit une mesure d'impureté $\gamma(.)$, un critère d'arrêt fixé à priori et un échantillon d'apprentissage $\mathcal{D}_n = \{(X_i, Y_i),\, 1 \leq i \leq n\}$, où pour tout $i \in \{1, \cdots, n\}$, $(X_i, Y_i) \in \mathcal{X} \times \{-1, +1\}$.

(a) (**Apprentissage.**) Construire un arbre de classification *maximal* $\mathcal{T}_{max}$ au moyen de l'algorithme CART, en partitionnant l'espace $\mathcal{X}$ de manière récursive et dyadique, de sorte à minimiser l'impureté empirique $\gamma_n(\hat{s}_{\mathcal{T}_{max}})$ associée au classifieur $\hat{s}_{\mathcal{T}_{max}}$, jusqu'à ce que le critère d'arrêt soit satisfait.

(b) (**Elagage.**) Parcourir $\mathcal{T}_{max}$, des feuilles vers la racine, en *élaguant* les branches de sorte à construire une suite de sous-arbres de $\mathcal{T}_{max}$ *emboîtés*, minimisant le critère pénalisé suivant :

$$\mathrm{crit}_{\lambda}(\mathcal{T}) = \gamma_{\mathrm{n}}(\hat{\mathrm{s}}_{\mathcal{T}}) + \lambda \frac{|\mathcal{T}|}{\mathrm{n}}, \tag{2.14}$$

pour différentes valeurs du paramètre de régularisation $\lambda \in \mathbb{R}_+$, avec $|\mathcal{T}|$ le nombre de feuilles terminales de l'arbre de classification $\mathcal{T}$.

(c) (**Sélection de modèle.**) Sélectionner le *meilleur* sous-arbre de la suite obtenue, au sens du critère d'impureté pénalisé, à l'aide d'un échantillon de *validation*, indépendant de l'échantillon $\mathcal{D}_n$ et identiquement distribué, ou par une procédure de *validation croisée*.

En règle générale, lorsque l'on cherche à optimiser un critère pénalisé, tel que celui défini par l'Equation (2.14), la difficulté principale réside dans la nature continue du paramètre de régularisation. Ce problème ne se pose pas dans le contexte spécifique de l'élagage d'un arbre de classification $\mathcal{T}_{max}$, puisqu'il n'existe qu'un nombre fini de sous-arbres de $\mathcal{T}_{max}$. Cependant, l'exploration exhaustive de l'ensemble de cette *collection* d'arbres reste une tâche difficile, qui représente un coût de calcul important. Tout l'intérêt de la procédure d'élagage, proposée dans [Breiman *et al.* 1984], repose sur le fait que l'on peut se restreindre à

une suite finie de sous-arbres *emboîtés*[3], résumant toute l'information de $\mathcal{T}_{max}$. En effet, les auteurs ont montré qu'il existait une séquence finie de valeurs $\lambda_0 = 0 < \lambda_1 < \cdots < \lambda_K$, où $K \geq 1$, correspondant à une suite $(\mathcal{T}_k)_{0 \leq k \leq K}$ de sous-arbres *emboîtés* de $\mathcal{T}_{max}$, notés $\mathcal{T}_k \subset \mathcal{T}_{max}$, *optimaux* au sens où, quel que soit $k \in \{1, \cdots, K\}$, pour tout $\lambda \in [\lambda_k, \lambda_{k+1}[$, $\mathcal{T}_k = \arg\min_{\mathcal{T} \subset \mathcal{T}_{max}} \text{crit}_\lambda(\mathcal{T})$. Nous revenons plus en détail sur les fondements de cette procédure, que nous transposons aux arbres d'ordonnancement, dans le Chapitre 3.

Pour revenir au problème initial de la résolution de l'étape d'optimisation de la procédure TreeRank, nous proposons de mettre en oeuvre une version pondérée de cet algorithme de classification, dans laquelle, à chaque itération, le critère d'impureté empirique est pondéré par le taux empirique d'observations positives dans la cellule en cours de partitionnement. Nous allons voir ci-dessous que, grâce à cette approche, nous obtenons des règles de score interprétables et que nous sommes à même d'évaluer l'impact des différentes variables sur l'ordonnancement des observations de $\mathcal{X}$.

2.3.1.2 Une règle de score interprétable

Quand la procédure LeafRank repose sur la mise en oeuvre d'une version pondérée de l'algorithme de classification CART, l'arbre d'ordonnancement $\mathcal{T}_D$, $D \geq 1$, produit par l'algorithme TreeRank représente en réalité une collection d'arbres. En effet, comme on peut le voir sur la Figure 2.5 ci-dessous, chaque noeud de $\mathcal{T}_D$ est caractérisé par un arbre de classification. Pour éviter toute confusion, nous parlerons dorénavant d'arbre d'ordonnancement ou d'*arbre principal*, pour le modèle arborescent obtenu via l'algorithme TreeRank et de *sous-arbres*, pour désigner les arbres de classification obtenus à chaque itération via l'algorithme CART.

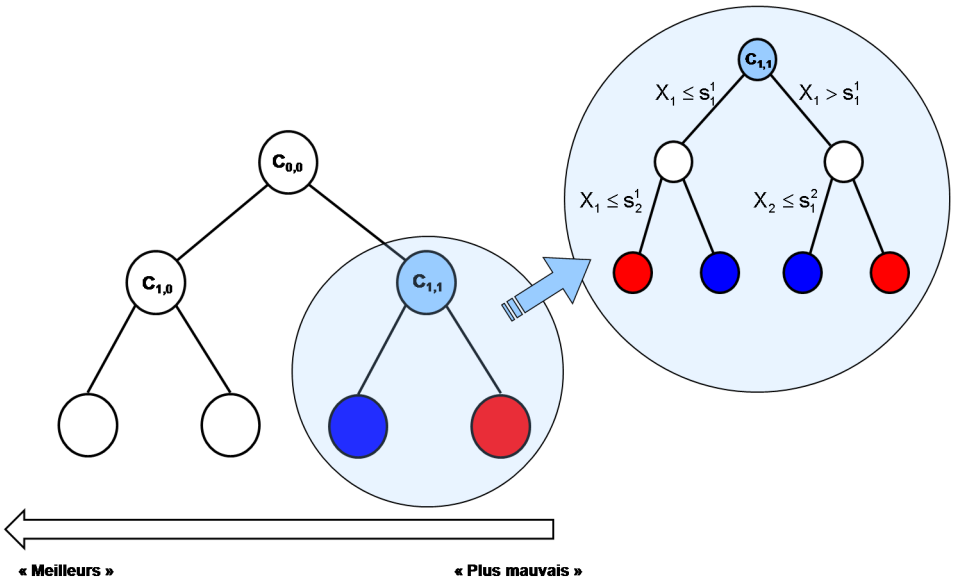


FIGURE 2.5 – Une collection d'arbres de classifications.

Le principal avantage de cette approche est que la règle de score induite par un tel arbre

3. au sens où ces arbres ont même racine et que chacun d'entre eux est obtenu par élagage du précédent

d'ordonnancement hérite des propriétés d'interprétabilité des règles de prédiction définies par les *sous-arbres*. Soit $\mathcal{T}_D$, $D \geq 1$, un arbre principal produit par la mise en oeuvre de l'algorithme TreeRank, dans lequel l'étape d'optimisation est résolue au moyen d'une version pondérée de l'algorithme CART, sur un échantillon $\mathcal{D}_n$ constitué de n copies *i.i.d.* du couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$, où $\mathcal{X} \subset \mathbb{R}^q$, $q \geq 1$. On rappelle qu'à l'issue de la procédure LeafRank, chaque noeud $C_{d,k}$ de $\mathcal{T}_D$, où $d \in \{1, \cdots D\}$ et $k \in \{0, \cdots, 2^d - 1\}$, est caractérisé par une règle de la forme :

$$\bigcup_{k \in \mathcal{K} \subset \{1, \cdots, q\}} \{\cap_{i \in \mathcal{I} \subset \mathcal{K}} \{X^{(i)} \leq c_m^i\}\} \bigcap \{\cap_{j \in \mathcal{J} \subset \mathcal{K}} \{X^{(j)} > c_m^j\}\}, \tag{2.15}$$

où pour tout $m \in \mathbb{N}$, c_m^i représente la $m^{\text{ème}}$ valeur seuil de la variable $X^{(i)}$ (dans le cas particulier où les q composantes du problème sont de nature continue). Ainsi, les *sous-arbres* de $\mathcal{T}_D$ étant *assemblés* selon une structure arborescente, la règle de score associée à $\mathcal{T}_D$ peut elle aussi être décrite par des intersections de règles de la même forme que celle définie par l'expression (2.15).

Cette propriété est particulièrement appréciable dans des applications telles que l'aide au diagnostic médical par exemple, où il est essentiel de pouvoir interpréter la règle d'ordonnancement obtenue. Par ailleurs, cette bonne interprétabilité du modèle peut permettre de déterminer quels sont les paramètres biologiques contribuant le plus aux variations du *score* des observations, à condition de s'équiper d'une mesure adéquate de la variabilité du score. Dans la partie suivante, nous proposons un outil permettant de mesurer l'importance relative des paramètres du modèle sur la règle d'ordonnancement produite par cette nouvelle version de l'algorithme TreeRank.

2.3.1.3 Mesure de l'importance relative des variables

Soit $\mathcal{T}_D$ un arbre d'ordonnancement, de profondeur $D \geq 1$, représentant une collection de sous-arbres de classification et induisant une règle de score $s \in \mathcal{S}$ sur l'espace $\mathcal{X}$. Nous nous sommes inspirés de l'heuristique, introduite dans [Breiman *et al.* 1984] pour le contexte de la classification binaire, pour proposer une mesure de l'influence des q composantes $X = (X^{(1)}, \ldots, X^{(q)})$ du modèle sur les prédictions de la règle de score s.

Pour tout noeud $C_{d,k}$ de $\mathcal{T}_D$, où $d \in \{0, \cdots, D-1\}$ et $k \in \{0, \cdots, 2^d - 1\}$, on note $\mathcal{T}(C_{d,k})$ le sous-arbre de classification de racine $C_{d,k}$, induisant une règle de score binaire sur $\mathcal{C}_{d,k}$, notée $s_{\mathcal{T}(C_{d,k})}$. De plus, pour tout noeud C_m de $\mathcal{T}(C_{d,k})$, on note $v(m)$ l'index de la composante utilisée pour scinder C_m et on pose $\Delta\widehat{\text{ASC}}(C_m)$ le gain, en termes d'ASC empirique, induit par ce partitionnement. On rappelle par ailleurs que, si l'on note $C_m = L_m \cup R_m$, où L_m est le noeud fils de C_m contenant le plus d'observations positives, alors le gain en ASC empirique s'écrit

$$\Delta\widehat{\text{ASC}}_{C_m(s_{\mathcal{T}(C_{d,k})})} = (\widehat{\alpha}(C_m)\widehat{\beta}(L_m) - \widehat{\beta}(C_m)\widehat{\alpha}(L_m))/2.$$

Avec ces notations, on propose, pour tout $j \in \{1, \cdots, q\}$, la mesure suivante de la perti-

nence de la $j^{\text{ème}}$ composante par rapport à la règle de prédiction $s_{\mathcal{T}(C_{d,k})}$:

$$\mathcal{I}_j(\mathcal{T}(C_{d,k})) = \sum_{\substack{C_m \text{ :} \\ \text{noeuds internes} \\ \text{de } \mathcal{T}(C_{d,k})}} \left(\frac{n_+(L_m)n_-(L_m)}{n_+(C_m)n_-(C_m)}\right)^2 \cdot \left(\Delta\widehat{\text{ASC}}(m)\right)^2 \cdot \mathbb{I}\{v(m)=j\} \quad (2.16)$$

où pour tout sous-ensemble $C \subset \mathcal{X}$, $n_+(C)$ et $n_-(C)$ désignent respectivement le nombre d'observations positives et négatives dans $C \times \{-1,+1\}$. Finalement, l'importance relative de la $j^{\text{ème}}$ composante au niveau de l'arbre principal est obtenue en sommant la quantité (2.16) sur les noeuds internes de l'arbre principal $\mathcal{T}_D$:

$$\mathcal{I}_j = \sum_{\substack{C_{d,k} \text{ : noeuds} \\ \text{internes de } \mathcal{T}_D}} \mathcal{I}_j(\mathcal{T}(C_{d,k})). \quad (2.17)$$

On souligne que le calcul de cette mesure d'importance relative des variables est directe et triviale, en ce sens qu'elle fait intervenir des quantités calculées automatiquement pendant la construction de l'arbre principal d'ordonnancement.

Ainsi, la mise en oeuvre récursive de l'algorithme de classification CART, pour scinder les noeuds d'un arbre d'ordonnancement, nous permet de proposer une nouvelle heuristique TreeRank. Les résultats empiriques, que nous présentons dans la Partie 2.3.3, mettent notamment en évidence la flexibilité de cette approche, qui conserve malgré tout de bonnes propriétés d'interprétabilité. Cependant, cette flexibilité peut s'avérer encore insuffisante dans certaines configurations, où les ensembles de niveaux de la fonction de régression sont très lisses et donc difficiles à approcher précisément par des sous-ensembles de côtés perpendiculaires aux axes de $\mathcal{X}$. Aussi, dans la partie suivante, nous considérons une deuxième approche, beaucoup plus flexible que celle que nous venons de présenter, qui repose sur la mise en oeuvre de Machines à Vecteurs Supports.

2.3.2 Implémentation récursive de Machines à Vecteurs Supports

Dans le contexte de classification, les Machines à Vecteurs Supports (SVM), introduites dans [Vapnik *et al.* 1992], permettent de résoudre des problèmes de discrimination, possiblement non-linéaires, en grande dimension. Cette approche repose sur les considérations théoriques relatives au problème de l'apprentissage statistique présentées dans [Vapnik 1998] et formule le problème de discrimination comme un problème d'optimisation quadratique sous-contrainte. Elle produit ainsi des hyperplans séparateurs de *marge* maximale, la *marge* définissant la distance entre l'hyperplan et les observations *les plus proches* de celui-ci, qui constituent les *vecteurs supports*. Dans cette partie, nous décrivons le principe général de cette méthode de classification. Nous renvoyons aux Chapitres 4 et 12 de [Hastie *et al.* 2001] ou encore aux références [Schölkopf & Smola 2002] et [Taylor & Cristianini 2000] pour plus de détails sur cette approche.

Afin de décrire cette méthode, nous considérons, dans un premier temps, un problème de classification binaire dans lequel les observations $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$, où pour

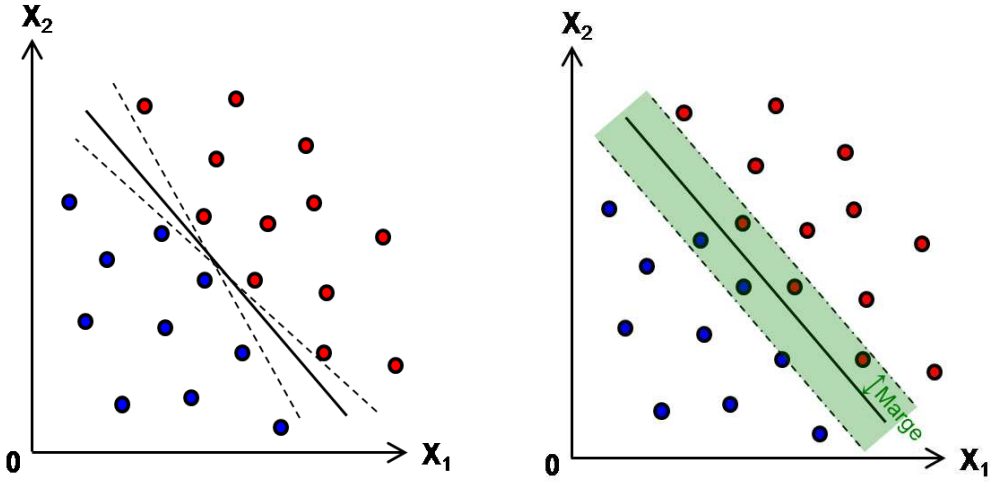
tout i, $(X_i, Y_i) \in \mathcal{X} \times \{-1, +1\}$ et $\mathcal{X} \subset \mathbb{R}^q$, $q \geq 1$, sont *linéairement séparables*. Puis, nous explicitons, dans un deuxième temps, la généralisation de cette approche, dans le cas *non séparable*, avant de présenter l'*extension*, proposée dans [Cortes & Vapnik 1995], dans laquelle les contraintes du problème d'optimisation quadratique sont assouplies par l'introduction de *variables ressorts*. Enfin, nous revenons au problème initial de la résolution de l'étape d'optimisation de la procédure TreeRank et formalisons l'adaptation de cette approche au cas spécifique d'un problème de classification binaire pondérée, avant de discuter des avantages et des inconvénients liés à la mise en oeuvre récursive de SVM.

2.3.2.1 Cas séparable

Dans le cas spécifique où les observations sont séparables, l'objectif est de trouver un *hyperplan* d'équation $h(x) = \mathbf{w}^T x + w_0 = 0$, où $\mathbf{w} \in \mathbb{R}^q$ est un vecteur de poids et $w_0 \in \mathbb{R}$, permettant de discriminer les deux classes représentées dans l'échantillon $\mathcal{D}_n$. Autrement dit, on cherche à déterminer la fonction de discrimination linéaire h satisfaisant la condition de séparabilité suivante :

$$\forall\, i \in \{1, \cdots, n\},\ Y_i h(X_i) \geq 0 \iff Y_i(\mathbf{w}^T X_i + w_0) \geq 0. \tag{2.18}$$

Comme on peut le voir sur la Figure 2.6(a) ci-dessous, il est possible de définir plusieurs hyperplans séparateurs. Cependant, il a été montré dans [Vapnik *et al.* 1992], que le séparateur linéaire *optimal*, en termes de performances de prédiction sur un nouvel échantillon de $\mathcal{X}$, correspond à l'hyperplan de *marge* maximale. La *marge* définit la distance entre l'hyperplan et les observations de l'échantillon *les plus proches* de celui-ci, on appelle ces points les *vecteurs supports*. Dans l'exemple schématisé sur la Figure 2.6(b), on dénombre 5 vecteurs supports dans la marge de l'hyperplan optimal.



a. Hyperplans séparateurs. b. Vecteurs Supports.

Figure 2.6 – SVM : cas séparable.

Le problème de classification peut donc s'écrire de la façon suivante :

$$\max_{\mathbf{w}, w_0} \{\min_{i \in \mathcal{I}} \frac{Y_i(\mathbf{w}^T X + w_0)}{\|\mathbf{w}\|}\}, \tag{2.19}$$

où le sous-ensemble $\mathcal{I} \subset \{1, \cdots, n\}$ contient les index des observations de $\mathcal{D}_n$ correspondant aux vecteurs supports et la quantité $(Y_i(\mathbf{w}^T X + w_0))/\|\mathbf{w}\|$ représente la projection orthogonale du vecteur support X_i sur l'hyperplan. Au moyen d'une renormalisation adéquate, on peut exprimer le problème défini par l'expression (2.19) sous la forme du problème d'optimisation quadratique suivant :

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \text{ sous contrainte que } Y_i(\mathbf{w}^T X_i + w_0) \geq 1, \forall\, i \in \{1, \cdots, n\}. \tag{2.20}$$

Ainsi, le problème de discrimination, défini sous sa forme dite *primale*, peut être résolu de manière classique par la méthode des *Multiplicateurs de Lagrange*. Cependant, on préfère considérer la formulation *duale*, définie en annulant les dérivées partielles du lagrangien et en ré-injectant les contraintes obtenues dans son expression, donnée par :

$$\begin{aligned} &\max \sum_{i=1}^{n} l_i - \frac{1}{2} \sum_{k,m=1}^{n} l_k l_m Y_k Y_m X_k^T X_m \\ &\text{sous les contraintes } l_i \geq 0, \forall\, i \in \{1, \cdots, n\} \text{ et } \sum_{k=1}^{n} l_k Y_k = 0. \end{aligned} \tag{2.21}$$

La résolution de ce problème d'optimisation quadratique convexe permet de déterminer la séquence $(l_i^*)_i$ des multiplicateurs de Lagrange optimaux et de définir l'équation de l'hyperplan optimal pour tout $X \in \mathcal{X}$:

$$h(X) = \sum_{i=1}^{n} l_i^* Y_i (X \cdot X_i) + w_0. \tag{2.22}$$

On peut remarquer que les seules observations entrant en ligne de compte, pour la définition de l'hyperplan optimal, sont les vecteurs supports $(X_i)_{i \in \mathcal{I}}$. Ce sont en effet les seuls points pour lesquels les contraintes du problème d'optimisation sont actives. De plus, d'après l'Equation (2.22), l'hyperplan optimal ne dépend des observations de $\mathcal{X}$ que par le biais de leur produit scalaire avec les vecteurs supports. C'est sur cette propriété cruciale que repose la généralisation de la méthode SVM au cas non-séparable.

2.3.2.2 Cas non-séparable

Dans le cas où les observations de l'échantillon $\mathcal{D}_n$ ne sont pas séparables linéairement, l'idée est d'appliquer une transformation non-linéaire Φ aux entrées $(X_i)_i$ afin de les *transposer* dans un espace $\mathcal{H}$, de dimension très supérieure à $q = \#(\mathcal{X})$. On peut en effet s'attendre à ce que la transformation Φ *rende* les observations de $\mathcal{X}$ linéairement séparables dans $\mathcal{H}$. Dans cette nouvelle configuration, la méthode SVM va trouver l'hyperplan optimal dans l'espace de *re-description* $\mathcal{H}$, solution du problème d'optimisation suivant :

$$\begin{aligned} &\max \sum_{i=1}^{n} l_i - \frac{1}{2} \sum_{k,m=1}^{n} l_k l_m Y_k Y_m \Phi(X_k)^T \Phi(X_m) \\ &\text{sous les contraintes } l_i \geq 0, \forall\, i \in \{1, \cdots, n\} \text{ et } \sum_{k=1}^{n} l_k Y_k = 0. \end{aligned} \tag{2.23}$$

Ce problème est bien plus difficile à résoudre que dans le cas séparable, car il nécessite de calculer le produit scalaire des observations dans l'espace $\mathcal{H}$ de grande dimension. Aussi, a-t-on recours en pratique à une astuce, désignée dans la littérature sous le terme de (*Kernel*

Trick), qui facilite grandement la résolution de ce problème d'optimisation. Cette astuce repose sur l'utilisation de *fonctions noyaux* correspondant à un produit scalaire dans un espace de grande dimension, *i.e.* de la forme

$$\begin{aligned} \mathbf{K} : \mathcal{X} &\to \mathbb{R} \\ (X_i, X_j) &\to \Phi(X_i)^T \Phi(X_j). \end{aligned}$$

Avec ces notations, l'hyperplan optimal solution du problème d'optimisation défini par l'équation (2.27) s'écrit :

$$h(X) = \sum_{i=1}^{n} l_i^* Y_i \mathbf{K}(X, X_i) + w_0, \tag{2.24}$$

et correspond à un hyperplan optimal *non-linéaire* sur l'espace $\mathcal{X}$, tel que celui schématisé sur la Figure 2.7.

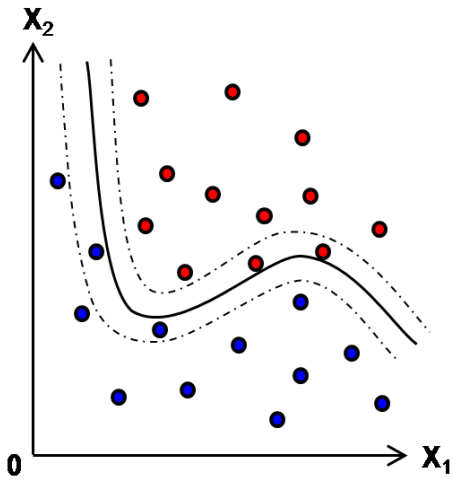


Figure 2.7 – SVM : cas non-séparable.

Grâce à la mise en oeuvre de la fonction noyau $\mathbf{K}$, il n'est plus nécessaire d'expliciter la transformation non-linéaire Φ et tous les calculs peuvent être faits dans l'espace d'entrée initial $\mathcal{X}$. On peut dès lors envisager des transformations Φ très complexes, transportant les données de $\mathcal{X}$ dans des espaces de re-description $\mathcal{H}$ de dimension possiblement infinie par exemple. En pratique, on ne définit ni la transformation Φ, ni l'espace $\mathcal{H}$, mais on choisit plutôt la fonction noyau $\mathbf{K}$. Parmi les noyaux les plus communs, on peut citer les trois exemples suivants :

- le *n*oyau linéaire : $\forall\,(X, X') \in \mathcal{X}^2$, $\mathbf{K}(X, X') = X \cdot X'$,
- le *n*oyau polynomial : $\forall\,(X, X') \in \mathcal{X}^2$, $k \leq 0$ et $c \in \mathbb{R}$, $\mathbf{K}(X, X') = (X \cdot X')^k$ ou $(c + X \cdot X')^k$,
- le *n*oyau gaussien : $\forall\,(X, X') \in \mathcal{X}^2$ et $\sigma \in \mathbb{R}$, $\mathbf{K}(X, X') = e^{-\|X - X'\|^2/\sigma}$.

2.3.2.3 Condition de marge souple

En règle générale, il n'est pas possible de séparer linéairement les observations, même dans l'espace de re-description $\mathcal{H}$ de grande dimension. Dans [Cortes & Vapnik 1995], les auteurs proposent une extension de la méthode SVM, qui *tolère* et contrôle les erreurs de classification. En effet, dans cette nouvelle approche, l'objectif est de trouver un hyperplan de marge maximale minimisant le nombre d'erreurs de classification.

Pour ce faire, les auteurs proposent d'introduire des *variables ressorts* $(\xi_i)_i$ permettant de relâcher les contraintes du problème d'optimisation, qui deviennent

$$Y_i(\mathbf{w}^T X_i + w_0) \geq 1 - \xi_i \text{ et } \xi_i \geq 0, \forall\, i \in \{1, \cdots, n\}. \tag{2.25}$$

Nous avons représenté les variables ressorts associées aux vecteurs supports d'un problème de discrimination sur la Figure 2.8.

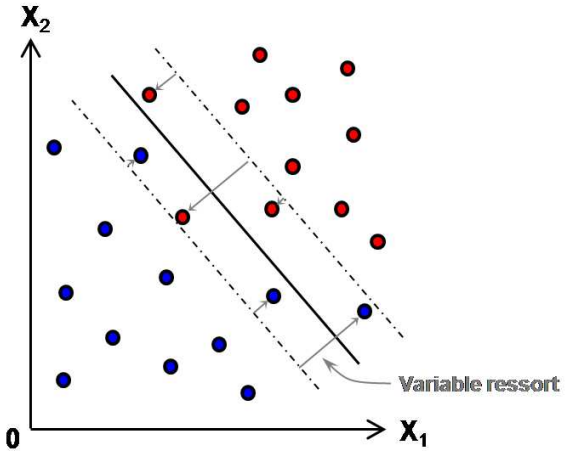


FIGURE 2.8 – Condition de marge souple : variables ressorts.

Ce relâchement des contraintes équivaut à introduire un nouveau terme dans la formulation primale du problème d'optimisation, pénalisant les observations dont les variables ressorts sont trop importantes. Avec ces nouvelles notations, la forme primale est donnée par :

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{n}\xi_i \\ &\text{sous les contraintes } Y_i(\mathbf{w}^T X_i + w_0) \geq 1 - \xi_i \text{ et } \xi_i \geq 0, \forall\, i \in \{1, \cdots, n\}, \end{aligned} \tag{2.26}$$

où la constante c permet de contrôler le compromis entre la largeur de la marge et l'erreur de classification. L'estimation statistique de cette constante est un problème difficile. Une approche possible, proposée dans [Hastie *et al.* 2004], consiste à résoudre le problème d'optimisation (2.28) pour toutes les valeurs du coefficient c, en construisant un *chemin de régularisation* via une heuristique inspirée de l'algorithme LARS ([Efron *et al.* 2004]). Toutefois, la pratique la plus courant rester de paramétrer manuellement ce coefficient.

De la même façon que précédemment, on peut considérer la forme duale du problème,

qui s'écrit (dans le cas non-séparable) :

$$\max \sum_{i=1}^{n} l_i - \frac{1}{2} \sum_{k,m=1}^{n} l_k l_m Y_k Y_m \Phi(X_k)^T \Phi(X_m) \tag{2.27}$$
$$\text{sous les contraintes } 0 \leq l_i \leq c, \forall\, i \in \{1, \cdots, n\} \text{ et } \sum_{k=1}^{n} l_k Y_k = 0.$$

2.3.2.4 Adaptation au problème de classification binaire pondérée

Rappelons que l'objectif de la procédure LeafRank est de résoudre un problème de classification binaire pondérée par un coût asymétrique ϖ, égal au taux de positif dans le noeud $C_{d,k}$ à scinder, avec $d \in \{0, \cdots, D-1\}$ et $k \in \{0, \cdots, 2^d - 1\}$. Une formalisation de ce problème pour les Machines à Vecteurs Supports est proposée dans le Chapitre 3 de [Joachims 2002a] (voir aussi [Bach *et al.* 2006]). Dans cette contribution, l'auteur montre que trouver un couple $(\mathbf{w}, w_0)$ solution du problème pondéré revient à résoudre le problème d'optimisation sous-contrainte suivant :

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^{n} c_i \xi_i \tag{2.28}$$
$$\text{sous les contraintes } Y_i(\mathbf{w}^T X_i + w_0) \geq 1 - \xi_i \text{ et } \xi_i \geq 0, \forall\, i \in \{1, \cdots, n\},$$

où, pour tout $i \in \{1, \cdots, n\}$, $c_i = c_+ \mathbb{I}\{Y = +1\} + c_- \mathbb{I}\{Y = -1\}$, avec c_+ et c_- les coûts associées à chaque classe (respectivement $(1 - \varpi)$ et ϖ dans notre cas). De manière analogue, la forme duale de ce problème devient

$$\max \sum_{i=1}^{n} l_i - \frac{1}{2} \sum_{k,m=1}^{n} l_k l_m Y_k Y_m \Phi(X_k)^T \Phi(X_m) \tag{2.29}$$
$$\text{sous les contraintes } 0 \leq l_i \leq c_i, \forall\, i \in \{1, \cdots, n\} \text{ et } \sum_{k=1}^{n} l_k Y_k = 0.$$

Soulignons que, dans le cas particulier où les coûts associés à chaque classe sont inconnus, il est possible de construire un *chemin de régularisation* afin de résoudre ce problème pondéré pour tous les couples (c_+, c_-), en mettant en oeuvre la méthode proposée dans [Bach *et al.* 2006], qui étend la démarche introduite dans [Hastie *et al.* 2004] pour les problèmes de classification non pondérée. Notons enfin que dans notre cas particulier où $c_+ = (1 - \varpi) = (1 - p)$ et $c_- = \varpi = p$, une solution au problème pondéré peut aussi être approchée en mettant en oeuvre une heuristique SVM *non pondérée* sur les données d'apprentissage ré-échantillonnées de manière adéquate.

2.3.2.5 Flexibilité et interprétabilité de la règle de score

D'après ce que nous venons de voir, les classifieurs SVM permettent donc de définir des séparateurs non-linéaires de marge maximale, en transposant les données d'entrée dans un espace de re-description de plus grande dimension. Il apparaît donc clairement que la mise en oeuvre de cette approche garantit une grande flexibilité dans la résolution de l'étape d'optimisation de la procédure TreeRank. Malheureusement, la règle de score produite

par cette nouvelle version de l'heuristique TreeRank n'est plus *intrinsèquement* interprétable, comme c'était le cas dans les deux versions précédemment introduites. Notons toutefois que dans ce cas précis, il reste possible d'obtenir des informations sur l'impact des q prédicteurs du problème.

Les *graphes de dépendances partielles*, introduits dans ([Friedman 2001]), permettent par exemple d'évaluer la *dépendance* de la règle d'ordonnancement s produite vis-à-vis de chaque prédicteur, comme suit : soit un sous-ensemble d'index $\mathcal{I} \subset \{1, \cdots, q\}$ et son complémentaire, que l'on note $\mathcal{J} = \{1, \cdots, q\} \setminus \mathcal{I}$. Avec ces notations, pour tout $X \in \mathcal{X}$, on pose $X^{\mathcal{I}}$ le vecteur constitué des composantes de X indexées par le sous-ensemble $\mathcal{I}$. Considérons maintenant une observation $X \in \mathcal{X}$, dont on *ré-organise* les q composantes en posant $X = (X^{\mathcal{I}}, X^{\mathcal{J}})$. On peut évaluer la dépendance de la règle d'ordonnancement $s \in \mathcal{S}$, induite sur $\mathcal{X}$, vis-à-vis du sous-ensemble des prédicteurs indexés par $\mathcal{I}$, en étudiant la variabilité de la *fonction de dépendance partielle* donnée par

$$\forall\, x \in \mathcal{X},\ s(x^{\mathcal{I}} \mid \mathcal{J}) = \mathbb{E}\left[s((x^{\mathcal{I}}, X^{\mathcal{J}}))\right],$$

en la remplaçant par sa contrepartie empirique calculée sur l'échantillon $\mathcal{D}_n$:

$$\forall\, x \in \mathcal{X},\ \hat{s}(x^{\mathcal{I}} \mid \mathcal{J}) = \frac{1}{n}\sum_{i=1}^{n} s((x^{\mathcal{I}}, X_i^{\mathcal{J}})).$$

En particulier, on peut visualiser graphiquement cette fonction pour un sous-ensemble $\mathcal{I}$ de cardinal $\#(\mathcal{I}) = 2$. Nous renvoyons à l'Annexe $A.2$ de [Friedman 2001] pour une discussion sur la pertinence de ces graphes de dépendances partielles et pour plus de détails sur leur implémentation dans le cas de fonctions de score constantes par morceaux représentées par des arbres.

2.3.3 Résultats expérimentaux

Afin d'illustrer et de comparer les deux nouvelles versions de l'heuristique TreeRank proposées dans ce chapitre, nous proposons de les appliquer à des jeux de données simulées. Pour simplifier la présentation des résultats, nous parlerons de l'algorithme $\mathrm{TRK}_{\mathrm{CART}}$, pour la version basée sur l'implémentation récursive de l'algorithme CART et de $\mathrm{TRK}_{\mathrm{SVM}}$, pour la version reposant sur la mise en oeuvre de classifieurs SVM.

Nous allons considérer trois exemples en dimension $q = 2$, reposant sur des mélanges de lois gaussiennes conditionnelles. Nous considérons tout d'abord un premier exemple *GaussLin2d*, sur lequel nous appliquons l'heuristique $\mathrm{TRK}_{\mathrm{CART}}$. Puis, nous comparons les deux heuristiques sur deux autres jeux de données. Nous reprenons notamment l'exemple *GaussCroix2d*, déjà introduit dans le Chapitre 1, et nous en proposons un nouveau, *GaussQuad2d*.

De même que précédemment, le taux théorique d'objets positifs $p = \mathbb{P}\{Y = +1\}$ est fixé à $1/2$ et l'échantillon généré est divisé en un échantillon d'apprentissage et un échantillon test, respectivement utilisés pour construire les arbres d'ordonnancement et estimer les courbes COR des règles de score associées. De plus, dans ces trois exemples on se placera sur $\mathcal{X} = \mathbb{R}^2$ et on utilisera les mêmes notations que dans la Partie 1.5.4.2 du Chapitre 1.

2.3.3.1 Exemple *GaussLin2d*

Dans cet exemple, les données sont générées sur $\mathcal{X} = [0,1]^2$ selon les lois conditionnelles suivantes :

$$\begin{aligned} H(dx) &= \mathcal{N}_{[0,1]^2}\left(\begin{pmatrix} 2 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 1 & 0.25 \\ 0.25 & 1.15 \end{pmatrix}\right), \\ G(dx) &= \mathcal{N}_{[0,1]^2}\left(\begin{pmatrix} -1 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 1 & 0.15 \\ 0.15 & 1.25 \end{pmatrix}\right), \end{aligned}$$

où l'on rappelle que $G(dx)$ et $H(dx)$ sont les distributions conditionnelles de la variable aléatoire $X \in \mathcal{X}$ sachant respectivement les événements $Y = +1$ et $Y = -1$. Pour $p = 1/2$, la transformée logistique de la probabilité à posteriori s'écrit :

$$\text{logit}\left(\eta(\mathrm{x})\right) = 0.02\mathrm{x}_1^2 + 0.05\mathrm{x}_2^2 - 3.08\mathrm{x}_1 + 0.53\mathrm{x}_2 - 0.11\mathrm{x}_1\mathrm{x}_2 + 1.32.$$

De la même façon que dans l'exemple *GaussCroix2d*, traité dans le Chapitre 1, la fonction de régression définit une collection infinie d'ensembles de niveaux sur $\mathcal{X}$. Nous en avons représenté 8 sur la Figure 2.9(a) ci-dessous et la Figure 2.9(b) représente les 8 ensembles estimés au bout de $D = 3$ itérations récursives d'une version pondérée de l'algorithme de classification CART. Enfin, les courbes COR optimale (trait plein rouge) et test (pointillés bleus) sont tracées sur la Figure 2.9(c).

Sur cet exemple, les ensembles de niveaux de la fonction de régression générée semblent linéaires, alors qu'ils sont en réalité quadratiques. Ceci est un effet d'échelle, qui est lié à la distance importante entre les moyennes des lois gaussiennes considérées. La comparaison des courbes COR indique que l'algorithme est très performant sur ces données même si la représentation des ensembles de niveaux semble quelque peu approximative. Cependant, étant donné le manque de précision que l'on peut observer sur la Figure 2.9(b), on pressent que cette heuristique risque d'être mise en défaut par des configurations plus complexes, avec par exemple des ensembles de niveaux de forme quadratique. Les deux exemples traités ci-dessous permettent d'illustrer ce problème.

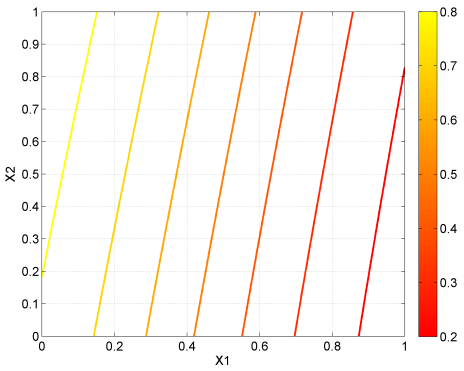
2.3.3.2 Exemples *GaussCroix2d* et *GaussQuad2d*

Dans cette partie, nous considérons deux exemples, sur lesquels nous allons comparer les deux heuristiques proposées dans ce chapitre. L'exemple *GaussCroix2d* a déjà été introduit, dans la Partie 1.5.4.2 du Chapitre 1. Quant au second, *GaussQuad2d*, il repose sur les lois gaussiennes conditionnelles suivantes :

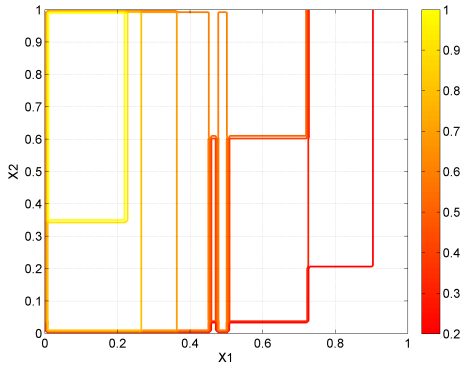
$$\begin{aligned} H(dx) &= \mathcal{N}_{[0,1]^2}\left(\begin{pmatrix} 0.64 \\ 0.51 \end{pmatrix}, \begin{pmatrix} 0.063 & 0.008 \\ 0.008 & 0.07 \end{pmatrix}\right), \\ G(dx) &= \mathcal{N}_{[0,1]^2}\left(\begin{pmatrix} 0.44 \\ 0.47 \end{pmatrix}, \begin{pmatrix} 0.076 & 0.002 \\ 0.002 & 0.079 \end{pmatrix}\right). \end{aligned}$$

On peut écrire la transformée logistique de la probabilité à posteriori comme suit :

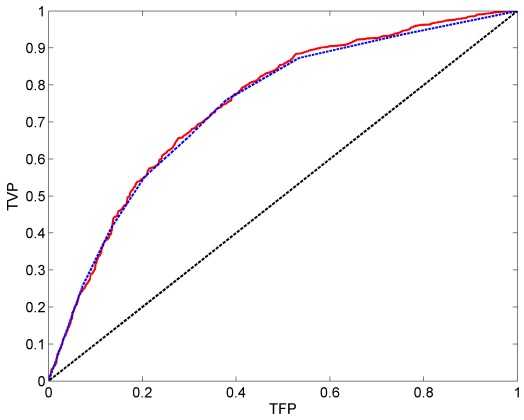
$$\text{logit}\left(\eta(\mathrm{x})\right) = 1.43\mathrm{x}_1^2 + 0.92\mathrm{x}_2^2 - 3.70\mathrm{x}_1 - 0.40\mathrm{x}_2 - 1.51\mathrm{x}_1\mathrm{x}_2 + 1.77.$$



a. Ensembles de niveaux de la probabilité η.



b. Ensembles de niveaux estimés par TreeRank.

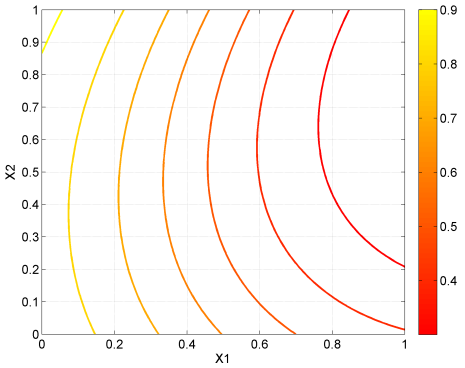


c. Courbes COR optimale (rouge) et test (pointillés bleus).

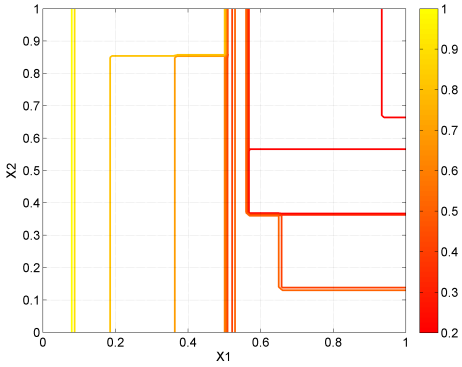
FIGURE 2.9 – Exemple *GaussLin2d*

Sur ces deux jeux de données, nous avons appliqué les algorithmes proposés pour une profondeur fixée $D = 3$. Les résultats obtenus sont résumés dans les Figures 2.10 pour l'exemple *GaussQuad2d* et 2.11 pour la configuration *GaussCroix2d*. Sur chacune de ces deux figures, nous avons représenté 8 ensembles de niveaux de la probabilité à posteriori générée, sur le graphe a. Les 8 sous-ensembles estimés par les algorithmes TRK_{CART} et TRK_{SVM} sont tracés respectivement sur les graphes b et c. Enfin, le graphe d permet de comparer les courbes COR obtenues. Comme précédemment, la courbe optimale apparaît en trait plein rouge et les courbes COR test obtenues par le biais de TRK_{CART} et TRK_{SVM} sont tracées respectivement en pointillés bleus « trait-trait » et en pointillés verts « point-point ».

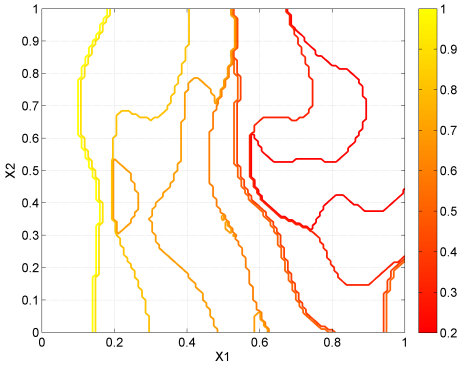
La comparaison des graphes b et c, sur les deux exemples traités, montre clairement la grande flexibilité de l'heuristique TreeRank basée sur la mise en oeuvre récursive de Machines à Vecteurs Supports. En effet, cette approche permet d'estimer des ensembles de niveaux beaucoup plus lisses que ceux obtenus par le biais de l'heuristique TRK_{CART}.



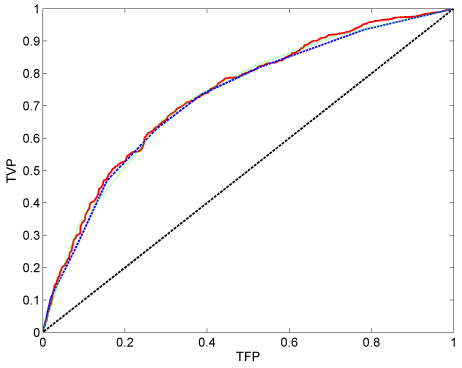
a. Ensembles de niveaux de la probabilité η.



b. Ensembles de niveaux estimés par TRK_{CART}.



c. Ensembles de niveaux estimés par TRK_{SVM}.



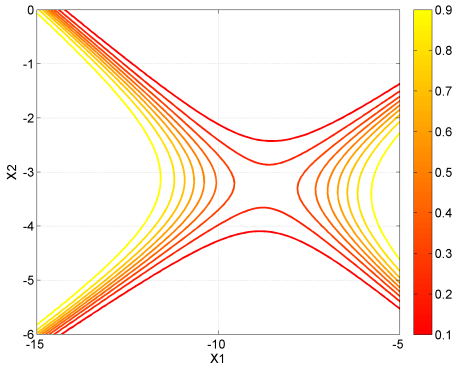
d. Courbes COR optimale (rouge) et test (pointillés bleus (TRK_{CART}) et verts(TRK_{SVM})).

Figure 2.10 – Exemple *GaussQuad2d*

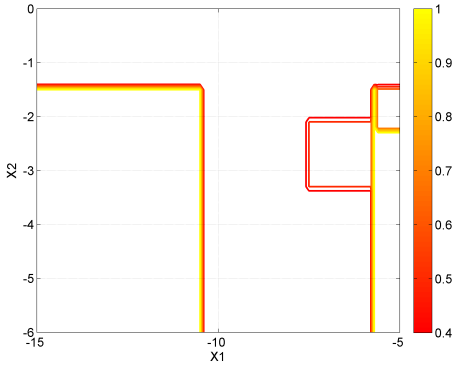
Sur le premier exemple, *GaussQuad2d*, la représentation des courbes COR, donnée par le graphe *d* de la Figure 2.10, montre toutefois que les deux heuristiques ont des performances comparables. Cependant, le gain apporté par l'algorithme TRK_{SVM} est bien visible sur le second exemple *GaussCroix2d*, la courbe COR test obtenue via TRK_{SVM} étant quasiment confondue avec la courbe COR optimale, alors que la courbe COR test produite par TRK_{CART} se trouve bien en deçà.

Concernant l'exemple *GaussCroix2d*, nous avons vu, dans la Partie 1.5.4.2 du Chapitre 1, que les performances de l'algorithme TreeRank, au sens du critère ASC, étaient fortement limitées par les erreurs d'ordonnancement réalisées dès les premières scissions. Au vu des résultats que nous venons de présenter, ce problème semble être résolu lorsque la procédure LeafRank est mise en oeuvre au moyen de classifieurs SVM.

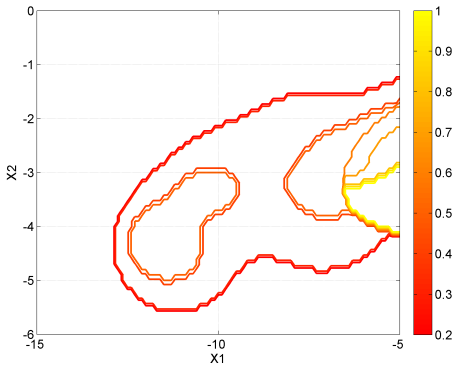
Cependant, il est moins évident de conclure à propos de l'heuristique TRK_{CART}, les performances moyennes obtenues sur cet exemple pouvant s'expliquer au moins de deux façons différentes. Premièrement, il est possible que, tout comme dans le chapitre précédent, les



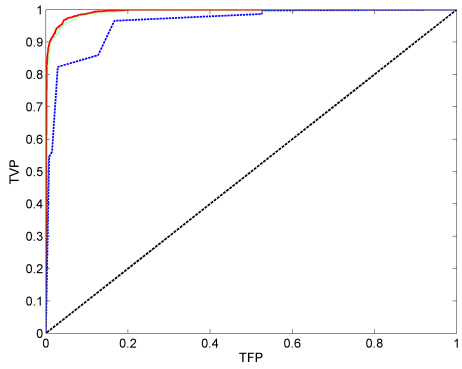
a. Ensembles de niveaux de la probabilité η.



b. Ensembles de niveaux estimés par $\mathrm{TRK}_{\mathrm{CART}}$.



c. Ensembles de niveaux estimés par $\mathrm{TRK}_{\mathrm{SVM}}$.

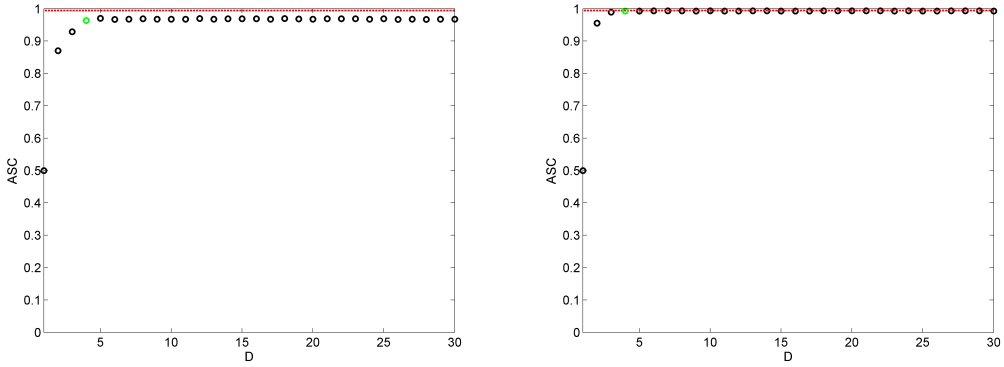


d. Courbes COR optimale (rouge) et test (pointillés bleus ($\mathrm{TRK}_{\mathrm{CART}}$) et verts($\mathrm{TRK}_{\mathrm{SVM}}$)).

FIGURE 2.11 – Exemple *GaussCroix2d*

erreurs commises lors des premières itérations soient encore trop importantes et limitent fortement l'optimisation de la courbe COR. Une deuxième explication possible est que la valeur fixée, pour la profondeur D de la procédure d'apprentissage, est insuffisante pour permettre d'estimer correctement les ensembles de niveaux de la probabilité à posteriori. Afin de vérifier ces hypothèses, nous avons évalué les performances de ces deux heuristiques, sur le jeu de données *GaussCroix2d*, en fonction de la profondeur D de la procédure d'apprentissage. Les graphes a et b de la Figure 2.12 suivante, illustrent l'évolution de l'ASC empirique de test, en fonction de la profondeur D, pour les algorithmes $\mathrm{TRK}_{\mathrm{CART}}$ et $\mathrm{TRK}_{\mathrm{SVM}}$ respectivement, par rapport à l'ASC optimale (tracée en trait plein rouge). Nous avons situé, en vert, le résultat obtenu pour la configuration des heuristiques présentées précédemment, *i.e.* pour la profondeur $D = 3$.

On voit clairement, sur la Figure 2.12*b*, que l'heuristique $\mathrm{TRK}_{\mathrm{SVM}}$ permet d'atteindre l'ASC empirique maximale. Ce n'est par contre pas le cas pour l'heuristique $\mathrm{TRK}_{\mathrm{CART}}$. Même si les résultats sont bien meilleurs que ceux obtenus par la première version de l'algorithme TREERANK, présentée dans le Chapitre 1, on voit que l'ASC empirique maximale



a. Heuristique TRK_{CART}. b. Heuristique TRK_{SVM}.

Figure 2.12 – Ex. *GaussCroix2d* - Evolution de l'ASC empirique avec la profondeur D.

atteinte par TRK_{CART} reste légèrement inférieure à l'ASC de référence (tracée en rouge).

Par ailleurs, la Figure 2.12*a* met en évidence un deuxième élément important, dont nous n'avons pas tenu compte jusqu'ici : l'importance du choix de la profondeur D de la procédure d'apprentissage. Sur ce graphe, nous avons représenté en vert l'ASC empirique atteinte par l'heuristique TRK_{CART}, pour une profondeur que nous avions fixée à $D = 3$. Or, nous constatons que l'on peut atteindre de meilleures performances en augmentant cette profondeur. Cette expérimentation nous permet ainsi d'illustrer le phénomène classique du *sous-apprentissage* : la profondeur D fixée de la procédure d'apprentissage ne permet de *capturer* que partiellement l'information contenue dans les données d'apprentissage, limitant ainsi les performances de la règle d'ordonnancement en termes de prédiction sur un nouvel échantillon d'observations.

De plus, l'observation des deux graphes de la Figure 2.12 montre aussi que l'augmentation de la profondeur ne permet d'améliorer les performances des heuristiques que jusqu'à un certain point, au-delà duquel l'ASC empirique n'augmente plus. Ainsi, choisir une profondeur D trop importante implique un coût de calcul supplémentaire sans pour autant apporter de gains en termes de performances. Cet exemple souligne donc l'importance du choix d'une profondeur D *suffisante*, permettant d'atteindre des performances maximales tout en limitant le nombre de calculs. Dans la Partie 2.3.1.1 de ce chapitre, nous avons vu que, dans le contexte de la classification binaire, des méthodes ont été proposées pour limiter la croissance d'un arbre de classification. On peut naturellement envisager de les appliquer dans notre contexte et de contrôler la profondeur de la procédure d'apprentissage en fixant, par exemple, un gain minimal en termes d'ASC, pour chaque itération, ou bien en fixant la taille minimale d'un noeud pouvant être scindé. Cependant, ces méthodes ne garantissent pas l'*optimalité* de la règle de prédiction obtenue. Aussi, nous proposons d'autres procédures pour sélectionner la *bonne* taille d'un arbre d'ordonnancement, de manière automatique et adaptative, dans le Chapitre 3.

2.4 Conclusion et perspectives

Nous avons montré que la méthode de scoring TreeRank repose sur la résolution récursive d'une étape d'optimisation de l'ASC locale, revenant à estimer un ensemble de niveau donné de la probabilité à posteriori η. Les performances globales de l'algorithme dépendent donc de la capacité de la règle de partitionnement choisie à estimer précisément ces ensembles. Dans ce chapitre, nous avons proposé une procédure, appelée LeafRank, visant à résoudre ce problème d'optimisation, qui peut être interprété comme un problème de classification binaire pondérée, par un coût asymétrique dépendant des données.

Cette nouvelle vision du problème, nous permet d'utiliser n'importe quel algorithme de classification, de manière récursive, afin d'estimer les ensembles de niveaux de la fonction de régression. Nous avons ainsi proposé deux heuristiques LeafRank, TRK_{CART} et TRK_{SVM}, basées respectivement sur une version pondérée de l'algorithme de classification CART et sur l'utilisation de classifieurs SVM. Une étude empirique nous a permis de montrer que, dans les deux cas, nous avons amélioré les performances globales de l'algorithme TreeRank. Il apparaît clairement que l'heuristique TRK_{SVM} apporte la plus grande flexibilité, cependant, l'heuristique TRK_{CART} présente le grand intérêt de produire des règles de score interprétables.

Naturellement, on pourrait imaginer utiliser n'importe quel algorithme de classification. Dans l'idée de conserver l'interprétabilité de la règle de score, on peut penser par exemple aux méthodes du bagging ([Breiman 1996b]) et des forêts aléatoires ([Breiman 2001]), qui permettraient d'obtenir des ensembles avec des frontières plus lisses que ceux estimés au moyen de TRK_{CART}. Nous reviendrons plus en détail sur ces méthodes dans le Chapitre 4. On pourrait même envisager, qu'à chaque itération de la procédure TreeRank, l'algorithme sélectionne automatiquement, en fonction des données, une procédure d'optimisation parmi une collection de candidates. La procédure d'apprentissage s'adapterait ainsi à une évolution de la géométrie des ensembles de niveaux de η. Le seul inconvénient de cette approche est qu'elle nécessiterait d'évaluer les différentes méthodes, à chaque itération, afin de choisir la plus performante, ce qui augmenterait nettement le temps de calcul.

Par ailleurs, l'étude empirique présentée dans ce chapitre met en évidence un problème, dont nous n'avions pas tenu compte jusqu'à présent. En effet, les résultats obtenus soulignent l'impact de la profondeur de la procédure d'apprentissage sur les performances de la règle de prédiction. Il s'agit là du problème de *sélection de modèle*, crucial dans le contexte de l'apprentissage automatique. En effet, la profondeur de la procédure permet de contrôler la complexité de la règle de prédiction et donc de garantir la bonne *généralisation* de cette règle, en termes de prédictions sur un nouvel échantillon d'observations. Sélectionner une bonne profondeur revient à établir un compromis : si la profondeur choisie est trop faible, la règle de score n'est pas assez riche pour *capturer* l'information contenue dans les données, on parle du phénomène de *sous-apprentissage* ; à contrario, une profondeur trop grande conduit à un modèle trop proche des données, dont il *capture* les spécificités, c'est le phénomène du *sur-apprentissage*. Dans les deux cas, la règle de prédiction ne se généralise pas correctement sur de nouvelles observations et ses performances en prédictions s'en trouvent amoindries. L'objectif du chapitre suivant est de proposer des procédures permettant de sélectionner automatiquement la profondeur optimale d'un arbre d'ordonnancement, en fonction des données.

Chapitre 3

Elagage d'un arbre d'ordonnancement

L'algorithme TreeRank, tel que nous l'avons introduit jusqu'ici, est paramétré par la profondeur $D \geq 1$ de l'arbre d'ordonnancement $\mathcal{T}_D$, que l'on cherche à construire à partir d'un échantillon $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$ d'observations de l'espace $\mathcal{X} \times \{-1, +1\}$. Or, les résultats expérimentaux présentés dans la Partie 2.3.3 du Chapitre 2 montrent que ce paramètre a une grande influence sur les performances de la règle de score associée à $\mathcal{T}_D$. En effet, nous avons vu qu'il est important de choisir une profondeur D *suffisante* pour que cette règle de prédiction se généralise correctement à tout l'espace $\mathcal{X}$, mais qu'a contrario, choisir une profondeur trop *importante* génère des calculs inutiles, au sens où ils ne s'accompagnent d'aucune amélioration des performances de la règle de prédiction.

Ce chapitre aborde le problème du choix de la profondeur *optimale* d'un arbre d'ordonnancement, au sens du critère ASC. Il s'agit là du problème de la *sélection de modèle*, crucial lorsque l'on cherche à définir des règles *optimales* en termes de performances en prédiction. Dans la première partie de ce chapitre, nous présentons ce problème plus en détail et introduisons les deux grandes approches proposées dans la littérature, visant à le résoudre : la *validation* et la *pénalisation*.

Nous proposons ensuite de construire un arbre d'ordonnancement *optimal* en deux étapes : d'abord en estimant un arbre de profondeur maximale puis, en *élaguant* ses branches, de sorte à optimiser l'ASC empirique pénalisée par un terme tenant compte de la complexité de la règle de prédiction.

Nous détaillons alors deux méthodes de pénalisation : une première qui consiste à optimiser l'ASC empirique pénalisée *linéairement* par une mesure de la complexité du modèle et une seconde qui repose quant à elle, sur l'optimisation de l'ASC structurelle, au sens de l'approche proposée dans [Vapnik 1982].

3.1 Sélection de modèle

On peut distinguer deux objectifs différents en théorie de l'apprentissage : la construction de modèles *explicatifs*, dont le but principal est d'identifier les variables caractéristiques de la *réponse* considérée, $Y \in \{-1, +1\}$ dans notre cas, et l'apprentissage de modèles

dans un but de *prédiction*. C'est dans ce deuxième cas de figure que nous nous plaçons ici. Notre objectif est en effet de sélectionner des règles de score *optimales* en termes de prédiction, *i.e. généralisables* à tout l'espace $\mathcal{X}$, sans que leurs performances ne soient altérées. Nous allons voir que ce problème nécessite d'établir un compromis entre le *biais* et la *variance* des prédictions.

En effet, soit $\mathbf{X_m} = \{X_1, \cdots, X_m\}$ un ensemble d'observations indépendant de l'échantillon d'apprentissage $\mathcal{D}_n$, constitué de $m \geq 1$ copies *i.i.d.* d'une variable aléatoire $X \in \mathcal{X}$. Si l'on considère un arbre $\mathcal{T}_{D_0}$, de faible profondeur D_0, sa fonction de score $s_{2^{D_0}}$ associée prendra un petit nombre de valeurs sur l'espace $\mathcal{X}$. Les prédictions de $s_{2^{D_0}}$ sur $\mathbf{X_m}$ auront donc une faible variance, au sens où elles sont peu dispersées autour de leur espérance, mais dans le même temps, leur biais est important, *i.e.* l'écart entre l'espérance des prédictions et l'espérance de la probabilité à posteriori est grand. On se place dans ce cas en situation de *sous-apprentissage*, au sens où, l'arbre $\mathcal{T}_{D_0}$ ne permet pas de capturer l'essentiel de l'information contenue dans l'échantillon $\mathcal{D}_n$.

A contrario, si l'on considère les prédictions d'une fonction de score associée à un arbre $\mathcal{T}_{D_\infty}$ de profondeur D_∞ importante, leur biais est faible et leur variance importante, la fonction $s_{2^{D_\infty}}$ étant beaucoup plus *complexe* que $s_{2^{D_0}}$. Cette situation de *sur-apprentissage* reflète que l'arbre $\mathcal{T}_{D_\infty}$ est trop proche des observations de $\mathcal{D}_n$, au sens où il en a capturé les caractéristiques générales, *héritées* de la distribution sur l'espace $\mathcal{X}$, mais aussi les spécificités propres.

Ces deux situations opposées ont une même conséquence : la fonction de score obtenue ne présente pas de bonnes performances en prédiction. Elle est, selon le cas considéré, trop *simple* ou trop *complexe* pour pouvoir se généraliser correctement sur l'espace $\mathcal{X}$.

Dans le cas spécifique de la procédure TreeRank, introduite dans les chapitres précédents, la construction d'une règle de score repose sur la maximisation récursive de l'ASC empirique, calculée sur l'échantillon d'apprentissage $\mathcal{D}_n$. On parlera de l'ASC empirique d'*apprentissage* ou de *resubstitution*, par analogie avec l'approche ERM (*Empirical Risk Minimization*) (cf Chapitre 7 de [Hastie & Tibshirani 1990]). Pour toute fonction de score $s \in \mathcal{S}$, on notera $\widehat{\mathrm{ASC}}^{(a)}$ cette quantité, qui s'écrit

$$\begin{aligned} \widehat{\mathrm{ASC}}^{(a)}(s) &= \frac{1}{n_+ n_-} \sum_{i:\, Y_i=+1} \sum_{j:\, Y_j=-1} \mathbb{I}\{s(X_i) > s(X_j)\} \\ &+ \frac{1}{2} \cdot \frac{1}{n_+ n_-} \sum_{i:\, Y_i=+1} \sum_{j:\, Y_j=-1} \mathbb{I}\{s(X_i) = s(X_j)\}, \end{aligned} \tag{3.1}$$

où $n_+ = \sum_{i=1}^n \mathbb{I}\{Y_i = +1\} = n - n_-$ est le nombre d'observations positives dans l'échantillon $\mathcal{D}_n$. En optimisant ce critère empirique, on peut atteindre une complexité de la règle de score suffisante pour éviter le phénomène de *sous-apprentissage*. Toutefois, cette optimisation ne garantit pas d'obtenir une règle de score *optimale* au sens de la prédiction. En effet, comme nous l'avons schématisé sur la Figure 3.1 ci-dessous, on peut améliorer l'ASC de resubstitution en construisant une fonction de score de plus en plus *proche* des données d'apprentissage. Dans ce cas, le modèle $\hat{s}$ défini par

$$\hat{s} \in \arg\max_{s \in \mathcal{S}} \widehat{\mathrm{ASC}}^{(a)}(s),$$

résume toute l'information contenue dans les données d'apprentissage : les caractéristiques héritées de la distribution $\mathcal{P}_{X,Y}$ et les spécificités de l'échantillon $\mathcal{D}_n$. Ainsi, l'estimateur empirique $\widehat{\mathrm{ASC}}^{(a)}(\hat{s})$ sur-estime les performances de la règle de score $\hat{s}$, qui, du fait de sa proximité avec les données d'apprentissage, sera peu performante pour ordonner les nouvelles observations de l'échantillon $\mathbf{X_m}$.

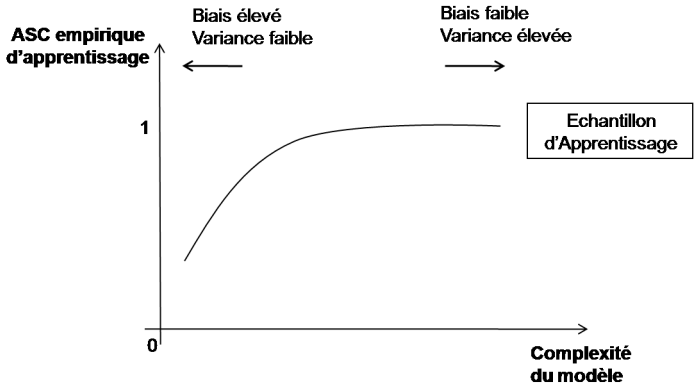


FIGURE 3.1 – Evolution de l'ASC empirique d'apprentissage avec l'augmentation de la complexité du modèle.

Deux approches différentes ont été proposées dans la littérature pour contourner ce problème et sélectionner la meilleure règle de prédiction. La première repose sur l'utilisation de données dites de *validation*, indépendantes des données d'apprentissage, qui permettent d'évaluer la capacité de généralisation du modèle. Nous présentons ces méthodes de *validation* dans la Partie 3.1.1 suivante. La deuxième approche consiste à optimiser le critère empirique calculé sur les données d'apprentissage, mais pénalisé par un terme tenant compte de la complexité du modèle, de sorte à limiter sa taille. Un panorama des méthodes de *pénalisation* est donné dans la Partie 3.1.2. Nous renvoyons au Chapitre 7 de [Hastie *et al.* 2001], ainsi qu'aux références [Bartlett *et al.* 2002] et [Boucheron *et al.* 2005] pour plus de détails sur le problème de la sélection de modèle, de manière générale et dans le cas particulier de la classification binaire.

3.1.1 Méthodes de validation

Soit $\widetilde{\mathcal{S}} \subset \mathcal{S}$ une collection de règles de score apprises à partir de l'échantillon d'apprentissage $\mathcal{D}_n$. L'idée sous-jacente aux méthodes de validation est de sélectionner le modèle $\hat{s}^* \in \widetilde{\mathcal{S}}$ maximisant le critère empirique suivant

$$\begin{aligned} \widehat{\mathrm{ASC}}^{(v)}(s) &= \frac{1}{n'_+ n'_-} \sum_{i:\, Y'_i=+1} \sum_{j:\, Y'_j=-1} \mathbb{I}\{s(X'_i) > s(X'_j)\} \\ &+ \frac{1}{2} \cdot \frac{1}{n'_+ n'_-} \sum_{i:\, Y'_i=+1} \sum_{j:\, Y'_j=-1} \mathbb{I}\{s(X'_i) = s(X'_j)\}, \end{aligned} \tag{3.2}$$

calculé sur un échantillon $\mathcal{D}'_{n'} = \{(X'_1, Y'_1), \ldots, (X'_{n'}, Y'_{n'})\}$ constitué de $n' \geq 1$ copies *i.i.d.* du couple (X, Y), indépendantes des observations de $\mathcal{D}_n$, avec $n'_+ = \sum_{i=1}^{n'} \mathbb{I}\{Y'_i = +1\} = n' - n'_-$. L'ASC empirique de *validation* ainsi définie, fournit une estimation plus réaliste de

la performance de la fonction de score s que l'ASC de resubstitution, donnée par l'expression (3.1). En effet, comme on peut le voir sur le schéma de la Figure 3.2 ci-dessous, l'ASC de validation décroît quand le modèle devient trop complexe, mettant ainsi en évidence le phénomène de *sur-apprentissage*.

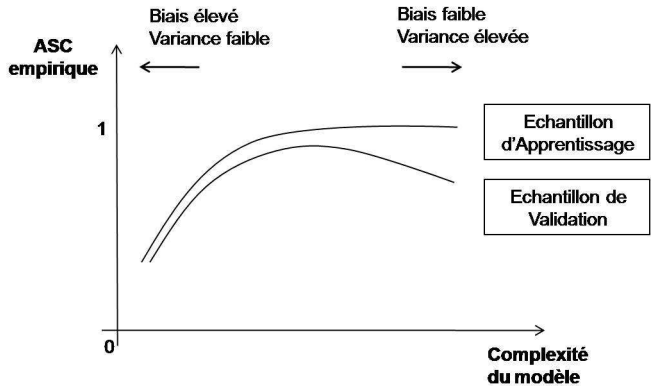


Figure 3.2 – Evolution de l'ASC empirique d'apprentissage et de validation avec l'augmentation de la complexité du modèle.

Ce critère empirique permet finalement d'évaluer la capacité de la règle de score considérée à se *généraliser* sur tout l'espace $\mathcal{X}$. Sa maximisation conduit donc à trouver la fonction de score *optimale* au sens de la prédiction parmi la collection $\widetilde{\mathcal{S}}$. Selon les observations disponibles pour l'apprentissage d'une telle règle de score, on peut considérer différentes méthodes de validation.

3.1.1.1 Validation « hold-out »

La méthode de validation la plus classique, parfois désignée sous le terme de validation « hold-out » (voir [Arlot 2007]), repose sur la construction de deux échantillons indépendants, servant respectivement pour l'apprentissage d'une collection de modèles et pour la sélection du *meilleur* d'entre eux, au sens des performances en prédiction.

Si l'on dispose d'un nombre $n \geq 1$ *suffisant* d'observations de l'espace $\mathcal{X} \times \{+1, -1\}$, il convient de les scinder en deux sous-échantillons indépendants $\mathcal{D}_{n_a}^{(a)}$ et $\mathcal{D}_{n_v}^{(v)}$, où $n_a + n_v = n$ et d'utiliser les observations de $\mathcal{D}_{n_a}^{(a)}$ pour apprendre une collection de modèles $\widetilde{\mathcal{S}} \subset \mathcal{S}$, telle que

$$\forall \hat{s} \in \widetilde{\mathcal{S}},\ \hat{s} \in \arg\max_{s \in \mathcal{S}} \widehat{\mathrm{ASC}}^{(a)}(s),$$

puis de calculer l'ASC de validation associée à chaque fonction $\hat{s} \in \widetilde{\mathcal{S}}$, évaluant leur performance sur l'échantillon $\mathcal{D}_{n_v}^{(v)}$. Le modèle optimal, au sens de la prédiction, parmi la collection $\widetilde{\mathcal{S}}$ est alors défini par

$$\hat{s}^* \in \arg\max_{\hat{s} \in \widetilde{\mathcal{S}}} \widehat{\mathrm{ASC}}^{(v)}(\hat{s}).$$

Remarque 10 (ASC **généralisée)** *Soit $\hat{s}^* \in \mathcal{S}$ la règle de score définie sur $\mathcal{X}$, sélectionnée par validation selon la procédure décrite ci-dessus. Il est vrai que l'*ASC *empirique de*

*validation fournit une estimation plus réaliste de la performance de $\hat{s}^*$ que l'*ASC *empirique de resubstitution ; cependant, si l'on souhaite disposer d'une estimation non biaisée des performances de cette règle de score, i.e. de ce que l'on appellera l'*ASC *généralisée, l'idéal est de disposer d'un troisième échantillon, dit échantillon test, indépendant des données d'apprentissage et de validation. En effet, la fonction $\hat{s}^*$ ayant été sélectionnée à partir des données de validation, le critère empirique $\widehat{\mathrm{ASC}}^{(v)}(s)$ sur-estime ses performances. Dans nos travaux cependant, nous ne nous sommes pas intéressés à ce problème d'estimation. Aussi, nous confondrons abusivement dans ce manuscrit la notion d'échantillon de validation et d'échantillon test.*

Cette méthode de sélection de modèle est particulièrement simple à mettre en oeuvre et l'indépendance entre les échantillons, utilisés respectivement pour l'apprentissage, la sélection et éventuellement pour l'estimation de l'ASC généralisée (cf Remarque 10), permet d'obtenir des résultats théoriques de consistance et des taux de convergence pour la fonction de score ainsi obtenue. Toutefois, le principal obstacle à la mise en pratique de cette méthode réside dans le fait qu'elle requiert de disposer d'un nombre conséquent d'observations pour la construction des échantillons indépendants. Lorsque ce n'est pas le cas, on lui préfère les méthodes de *validation croisée*, dans lesquelles la sélection d'un modèle optimal repose sur le *ré-échantillonnage* des observations disponibles selon divers schémas.

3.1.1.2 Validation croisée

Les procédures de *validation croisée* permettent de généraliser la méthode de validation « hold-out » et de l'appliquer même dans le cas où la taille de l'échantillon initial $\mathcal{D}_n$ est restreinte. Le principe est de réaliser des découpes successives de l'échantillon $\mathcal{D}_n$ et d'appliquer sur chacune d'entre elles la procédure de validation « hold-out ». La sélection du modèle optimal repose alors sur l'optimisation de l'ASC empirique de validation, moyennée sur l'ensemble des itérations. Plusieurs méthodes de validation croisée ont été proposées dans la littérature, reposant sur divers schémas de ré-échantillonnage.

Dans toutes ces méthodes, la sélection du modèle *optimal* reposant sur l'optimisation du critère empirique moyenné sur plusieurs itérations, la propriété d'indépendance entre les procédures d'apprentissage et de validation n'est plus vérifiée. Il devient donc difficile d'établir des résultats théoriques sur la consistance des règles de prédictions sélectionnées. Citons toutefois les récentes avancées présentées dans [Cornec 2009]. Cette approche reste malgré tout une des plus populaires auprès des praticiens, en raison de sa facilité de mise en oeuvre, même si, comme nous allons le voir, certaines questions peuvent se poser, selon le choix du schéma de ré-échantillonnage. On peut cependant compter sur de nombreuses comparaisons empiriques entre différentes configurations, pour guider le choix des praticiens (voir par exemple [Efron 1983], [Efron 1986] et [Efron & Tisbshirani 1997]).

Le « leave-one-out » a été la première méthode de validation croisée proposée dans la littérature ([Allen 1974], [Stone 1974] et [Geisser 1975]). Elle consiste à retirer successivement chacune des observations (X_i, Y_i) de l'échantillon $\mathcal{D}_n$ et à les utiliser pour estimer le critère empirique de validation associé au(x) modèle(s) appris sur l'échantillon $\mathcal{D}_n^{-i}$, privé de l'observation (X_i, Y_i). Cette approche est couramment appliquée dans les contextes de la classification et de la régression, mais ne permet pas de calculer l'ASC empirique de validation, le calcul de ce critère nécessitant de comparer les scores d'au moins deux observations $(X_i, X_j) \in \mathcal{X}^2$, pour $i \neq j$.

On peut par contre appliquer une variante de cette approche, le « leave-m-out », qui consiste à utiliser successivement tous les sous-échantillons $\mathcal{D}_m \subset \mathcal{D}_n$, constitués de $m \geq 2$ observations de l'espace $\mathcal{X} \times \{+1, -1\}$, comme échantillons de validation pour le(s) modèle(s) appris sur leur complémentaire $\mathcal{D}_n \setminus \mathcal{D}_m$. Considérer les sous-échantillons de $\mathcal{D}_n$ à m éléments, au lieu des singletons, permet de réduire la variabilité de l'estimation, cependant, cela pose une nouvelle question, celle du choix de la taille m des sous-ensembles à considérer. Par ailleurs, la procédure du « leave-m-out » reposant sur l'exploration exhaustive de tous les sous-ensembles $\mathcal{D}_m$ de $\mathcal{D}_n$, son coût, en termes de temps de calcul, devient rapidement un facteur limitant.

La validation croisée par blocs (« V-fold cross validation »), proposée dans [Geisser 1975], est une alternative intéressante. Son temps de calcul raisonnable et sa simplicité en font la procédure de sélection de modèle la plus utilisée en pratique. Cette méthode repose sur la définition d'une partition disjointe $\mathcal{B} = (\mathcal{D}_n^{(v)})_{1 \leq v \leq V}$ de $\mathcal{D}_n$ en $V \geq 2$ cellules non-vides de tailles équivalentes. De même que précédemment, chaque *bloc* $\mathcal{D}_n^{(v)}$, pour tout $v \in \{1, \cdots, V\}$, est utilisé comme échantillon de validation pour le(s) modèle(s) appris sur son complémentaire $\mathcal{D}_n \setminus \mathcal{D}_n^{(v)}$. On sélectionne finalement le modèle qui optimise le critère empirique de validation moyenné sur les V blocs.

On trouve notamment, dans le Chapitre 7 de [Hastie *et al.* 2001], une étude empirique mettant en évidence les performances intéressantes de cette méthode de sélection. Les auteurs soulignent toutefois que la variabilité de l'estimateur obtenu peut être importante et qu'il convient de l'évaluer et d'en tenir compte dans l'interprétation des résultats. Par ailleurs, tout comme pour la méthode « leave-m-out », une question se pose quant au paramétrage de cette procédure de validation. En effet, il est montré dans l'introduction de [Arlot 2007], que le choix du nombre V de blocs a une influence non négligeable sur la qualité de l'estimateur. De plus, l'auteur remet en cause la règle *tacite* selon laquelle un nombre de blocs $5 \leq V \leq 10$ serait un bon choix, spécialement dans le cadre non-asymptotique, où le nombre n d'observations est limité. Il convient donc en pratique d'apporter une attention particulière au paramétrage de cette procédure de validation croisée.

3.1.2 Méthodes de pénalisation

Une deuxième approche au problème de la sélection de modèle, radicalement différente, consiste à optimiser un critère empirique pénalisé par un terme permettant d'évaluer la complexité du modèle. Elle a été développée sur la base d'un constat selon lequel le critère de resubsitution sur-estime l'ASC en ne tenant pas compte de la complexité du modèle estimé. Les méthodes de pénalisation reposent donc sur l'optimisation d'un critère plus *réaliste*, qui pénalise les modèles trop complexes au profit de règles plus facilement *généralisables*. Là encore, on distingue deux types de méthodes.

On trouve d'une part les méthodes de *régularisation*, qui consistent à optimiser un critère pénalisé *linéairement* par une mesure de la complexité du modèle. Dans ce cas, le problème revient à trouver une règle de prédiction $\hat{s}^* \in \mathcal{S}$ telle que

$$\hat{s}^* \in \arg\max_{s \in \mathcal{S}} \widehat{\mathrm{ASC}}^{(a)}(s) - \lambda \cdot ||s||_{\mathcal{L}},$$

où $\lambda \in \mathbb{R}_+$ est un paramètre de régularisation et la quantité $||s||_{\mathcal{L}}$ représente une mesure de la complexité de la règle $s \in \mathcal{S}$ pour une norme $\mathcal{L}$ donnée. Deux problèmes se posent lors de la mise en oeuvre de cette approche : le choix *à priori* d'une pénalité adéquate et l'estimation du coefficient de régularisation, pouvant prendre une infinité de valeurs. En règle générale, ce problème d'estimation est résolu au moyen de méthodes de type *Monte-Carlo* ou d'une procédure de validation croisée. Les *chemins de régularisation* sont une autre solution possible à ce problème d'optimisation (voir par exemple [Efron *et al.* 2004] et le Chapitre 16 de [Hastie *et al.* 2001]).

D'autre part, de nombreuses méthodes de pénalisation reposent sur l'optimisation d'un critère empirique pénalisé par un terme, s'écrivant comme une fonction non linéaire de la complexité du modèle et pouvant être calculé à partir des observations de l'échantillon $\mathcal{D}_n$. Dans [Vapnik 1982], cette approche est désignée sous le terme de *minimisation du risque structurel*, la pénalité considérée reflétant la structure du modèle. L'objectif de ces méthodes est d'estimer une pénalité *optimale*, au sens où elles permettent de sélectionner des règles de prédiction asymptotiquement consistantes. Notons de plus, que suite aux avancées récentes en théorie de la mesure, les travaux présentés dans [Massart 2007a] (voir aussi [Massart 2007b]) permettent d'élargir cette notion d'optimalité à des situations non-asymptotiques. En effet, à condition de vérifier certaines *inégalités de concentration*, on peut établir des *inégalités oracles*, garantissant l'optimalité des pénalités considérées pour un nombre d'observations n fixé. Le calcul de ces pénalités optimales nécessite de résoudre trois problèmes délicats, relatifs à la détermination de la structure de la pénalité, à son estimation et enfin, au choix d'une mesure de la complexité de la collection de règles de prédiction candidates.

3.1.2.1 Une pénalité idéale

Dans un premier temps, il faut identifier la *forme* de la pénalité. Dans le contexte de l'optimisation empirique du critère ASC, on peut définir la notion de *pénalité idéale*, de la même façon que pour les méthodes de type ERM. En effet, la problématique d'ordonnancement binaire, telle que nous l'avons définie dans le Chapitre 1, consiste à estimer la solution du problème d'optimisation

$$s^* \in \arg\max_{s \in \mathcal{S}} \mathrm{ASC(s)}, \tag{3.3}$$

à partir d'un échantillon $\mathcal{D}_n = \{(X_i, Y_i),\, 1 \leq i \leq n\}$ d'observations de $\mathcal{X} \times \{-1, +1\}$. Pour ce faire, l'approche ERM vise à construire une règle de score $\hat{s}^*$ maximisant l'ASC empirique calculée sur l'échantillon $\mathcal{D}_n$, *i.e.* telle que

$$\hat{s}^* \in \arg\max_{s \in \mathcal{S}} \widehat{\mathrm{ASC}}(s).$$

Comme nous l'avons déjà indiqué, ce critère empirique a tendance à sur-estimer les performances de la règle de prédiction et son optimisation conduit à des fonctions de score, dont les performances sont en-deçà de celles de la solution s^* définie par l'expression (3.3). Cependant, il est facile de voir que l'optimisation de l'ASC empirique, pénalisée par un terme de la forme

$$\mathrm{pen}_{id}(\hat{s}) = \widehat{\mathrm{ASC}}(\hat{s}) - \mathrm{ASC}(\hat{\mathrm{s}}), \tag{3.4}$$

permet de résoudre le problème (3.3). Malheureusement, la pénalité (3.4) est inaccessible, car elle dépend de la distribution inconnue des observations par le biais du terme ASC(s). Son estimation est une des principales difficultés des méthodes de pénalisation et on peut distinguer, une fois encore, deux approches différentes : une première, qui consiste à estimer directement la pénalité idéale, à partir des observations de l'échantillon $\mathcal{D}_n$ et une seconde, qui repose sur le calcul d'une borne supérieure pour la déviation entre le critère ASC et sa contrepartie empirique. Nous décrivons successivement ces deux approches dans les parties suivantes.

3.1.2.2 Pénalités ré-échantillonnées

Certaines méthodes proposent des estimateurs sans biais de la pénalité idéale $\text{pen}_{id}(\hat{s})$ (ou de son espérance ou encore de son majorant $\text{pen}_{id,g}(\hat{s}) = \sup_{s\in\mathcal{S}} \text{pen}_{id}(\hat{s})$), calculés à partir des observations de $\mathcal{D}_n$. L'idée générale, sous-jacente à ces méthodes, est d'approcher l'écart entre l'ASC et sa contrepartie empirique en *ré-échantillonnant* les observations selon le principe de l'heuristique *bootstrap*, introduite dans [Efron 1979], que nous avons schématisée sur la Figure 3.3 ci-dessous. De nombreuses pénalités *ré-échantillonnées* ont été proposées selon ce principe.

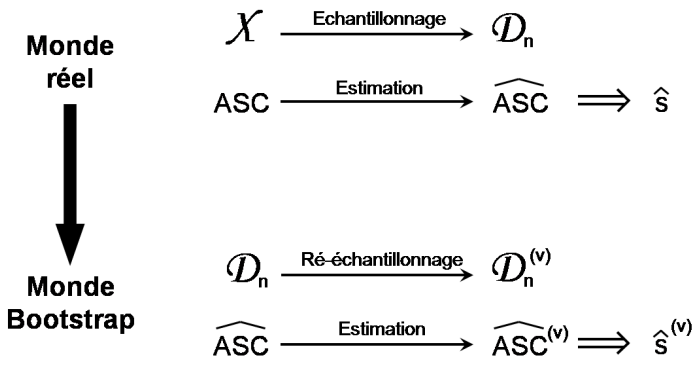


Figure 3.3 – Ré-échantillonnage : l'heuristique *bootstrap* d'Efron.

Soit $v \in \{1, \cdots, V\}$, où $V \geq 2$, on note $\hat{s}^{(v)}$ la fonction de score définie de sorte que

$$\hat{s}^{(v)} \in \arg\max_{s\in\mathcal{S}} \widehat{\text{ASC}}^{(v)}(s),$$

où $\widehat{\text{ASC}}^{(v)}(s)$ représente l'ASC empirique associée à la fonction de score $s \in \mathcal{S}$, calculée sur le $v^{\text{ème}}$ *ré-échantillon* des données d'apprentissage $\mathcal{D}_n$. Avec ces notations, on peut citer par exemple la *pénalité bootstrap*, proposée dans [Efron 1983]. Dans le contexte de l'ordonnancement binaire, cette dernière est définie par

$$\text{pen}_{Efron}(\hat{s}^V) = \mathbb{E}_B\left[\widehat{\text{ASC}}^V(\hat{s}^V) - \widehat{\text{ASC}}(\hat{s}^V)\right], \tag{3.5}$$

où l'espérance est prise sur les V versions ré-échantillonnées des estimations.

Dans le contexte de la classification binaire, d'autres pénalités ont été considérées, comme

par exemple la *pénalité de Rademacher globale*, introduite dans [Koltchinskii 2001] ou encore [Bartlett *et al.* 2002], pouvant être vue comme un cas particulier des *pénalités bootstrap à poids échangeables*, introduites dans [Fromont 2007], et qui fournit un estimateur sans biais du majorant $\text{pen}_{id,g}(\hat{s})$. D'autres pénalités tiennent compte du *positionnement*, en termes d'ASC dans notre cas, du modèle s parmi la collection explorée. C'est le cas par exemple des *pénalités de Rademacher locales* (voir [Koltchinskii 2006]).

Enfin, on peut citer les travaux plus récents, présentés dans [Arlot 2007], introduisant notamment une nouvelle pénalité, pouvant être vue comme une version *validation croisée par blocs* de la pénalité pen_{Efron}. De même que précédemment, posons $\mathcal{B} = (\mathcal{D}_n^{(v)})_{1 \leq v \leq V}$, $V \geq 2$, une partition disjointe de l'échantillon d'apprentissage $\mathcal{D}_n$. En notant $\widehat{\text{ASC}}^{(-v)}$ l'ASC empirique estimée sur le complémentaire de $\mathcal{D}_n^{(v)}$ et $\hat{s}^{(-v)}$ la fonction de score maximisant ce critère, cette *pénalité V-fold*, transposée au contexte de l'ordonnancement binaire, est donnée par

$$\text{pen}_{V-fold}(\hat{s}) = \frac{C}{V} \sum_{v=1}^{V} (\widehat{\text{ASC}}^{(-v)}(\hat{s}^{(-v)}) - \widehat{\text{ASC}}(\hat{s}^{(-v)})), \tag{3.6}$$

où la constante C est telle que $C \geq V - 1$. Dans ces travaux, l'auteur introduit aussi une nouvelle famille de *pénalités ré-échantillonnées à poids échangeables*, dont la *pénalité bootstrap* d'Efron est un cas particulier. Nous renvoyons à la contribution [Arlot 2007] pour plus de détails sur cette famille de pénalités et pour un recueil plus complet des méthodes de sélection de modèle par pénalisation.

3.1.2.3 Bornes supérieures pour la déviation du critère ASC

L'estimation directe de la pénalité idéale restant un problème d'estimation complexe, une autre approche repose sur le calcul de bornes supérieures pour la pénalité idéale (son espérance ou son majorant), tenant compte de la complexité du modèle estimé. Ces méthodes procèdent en deux étapes : dans un premier temps, elles consistent à borner la déviation entre le critère ASC et sa contrepartie empirique, puis elles procèdent à la sélection de la règle de prédiction associée à la borne la plus *faible*, définissant la pénalité du problème d'optimisation.

Comme nous l'avons déjà souligné dans le Chapitre 1, il n'est pas évident de mesurer la complexité d'une collection de règles de classification ou d'ordonnancement binaire. Les résultats théoriques présentés dans ce manuscrit, reposent sur le contrôle de cette complexité au moyen de la dimension de Vapnik-Chervonenkis (dimension VC) ou des *coefficients de pulvérisation* ([Vapnik 1996]) de la collection de modèles estimée par le biais de l'algorithme TreeRank. Comme nous allons le voir par la suite, l'utilisation de cette mesure conduit à la détermination de pénalités indépendantes de la distribution des observations de $\mathcal{X} \times \{-1, +1\}$. Notons aussi que d'autres mesures ont été proposées dans la littérature, comme par exemple les *moyennes conditionnelles de Rademacher* (voir par exemple [Giné & Zinn 1984] et [Fromont 2007]), qui peuvent être calculées à partir de l'échantillon des observations et conduisent ainsi à des pénalités dépendantes de l'échantillon $\mathcal{D}_n$.

Le principal avantage lié à l'utilisation des pénalités optimales, quelles qu'elles soient, par rapport aux méthodes de régularisation présentées précédemment, réside dans le fait

qu'elles permettent d'établir des résultats théoriques, notamment sur la consistance des règles de prédiction sélectionnées. Malheureusement, l'utilisation de ces pénalités en pratique est loin d'être évidente, car ces dernières sont toutes définies *à une constante près* seulement, qui reste à estimer, et ce problème d'estimation peut s'avérer délicat.

Aussi, dans ce chapitre, nous considérons ces deux formes de pénalisation : nous proposons tout d'abord une heuristique reposant sur l'optimisation de l'ASC empirique pénalisée linéairement par le nombre de feuilles de l'arbre d'ordonnancement produit, avant d'envisager une approche plus théorique, reposant sur le calcul d'une pénalité *optimale*, pour laquelle nous établissons un résultat de consistance des règles de prédiction sélectionnées. Mais avant de détailler ces deux approches, nous formulons dans la partie suivante, le problème de la sélection de la profondeur $D \geq 1$ de la procédure d'apprentissage comme un problème de sélection de modèle et nous introduisons le principe du processus d'*élagage* d'un arbre d'ordonnancement.

3.2 Elaguer un arbre binaire d'ordonnancement

L'objectif de cette partie est de formuler le problème de la détermination de la profondeur $D \geq 1$ *optimale* d'un arbre d'ordonnancement, au sens des performances de la règle de prédiction associée, comme un problème de sélection de modèle. Pour cela, nous introduisons tout d'abord quelques notions et notations supplémentaires.

Soit $\mathcal{T}_D$ un arbre d'ordonnancement de profondeur $D \geq 1$, représentant une partition $\mathcal{P}$ finie et ordonnée de l'espace d'entrée $\mathcal{X}$. On notera $\widetilde{\mathcal{T}_D}$ l'ensemble de ses feuilles terminales et $|\widetilde{\mathcal{T}_D}| = \#(\widetilde{\mathcal{T}_D}) = \#(\mathcal{P}(\mathcal{T}_D))$ le cardinal de cet ensemble. De plus, on affecte à tout noeud $\mathcal{C}_{d,k}$ de $\mathcal{T}_D$, où $d \in \{0, \cdots, D\}$ et $k \in \{0, \cdots, 2^d - 1\}$, un poids scalaire $\omega(C_{d,k})$, satisfaisant les contraintes suivantes :

- (*Elagage*) $\forall\, d \in \{0, \cdots, D-1\}$ et $\forall\, k \in \{0, \ldots, 2^d - 1\}$, $\omega(C_{d,k}) \in \{0, 1\}$.
- (*Hérédité*) si $\omega(C_{d,k}) = 1$, alors pour tout noeud $C_{d',k'}$ tel que $C_{d,k} \subset C_{d',k'}$, on a $\omega(C_{d',k'}) = 1$.

Toute séquence de poids $\omega = (\omega(C_{d,k}))_{d,k}$, pour $0 \leq d \leq D$ et $0 \leq k \leq 2^d - 1$, satisfaisant ces deux conditions est dite *admissible* et caractérise les noeuds d'un *sous-arbre* $\mathcal{T}(\omega)$ de $\mathcal{T}_D$. Un noeud $C_{d,k}$ constitue une feuille terminale de l'arbre $\mathcal{T}(\omega)$ si $\omega(C_{d,k}) = 1$ et $\omega(C_{d',k'}) = 0$, pour tout couple (d', k') tel que $C_{d',k'} \subset C_{d,k}$. Les feuilles de $\mathcal{T}(\omega)$ ainsi définies forment une partition $\mathcal{P}(\mathcal{T}(\omega))$ de l'espace $\mathcal{X}$. Muni de ces notations, on peut définir la notion de *sous-arbre élagué*.

Définition 10 *(Sous-arbre élagué)*
Soit $\mathcal{T}$ un arbre d'ordonnancement défini sur $\mathcal{X}$ et deux séquences admissibles ω_1 et ω_2 telles que

$$\{C_{d,k} \in \mathcal{T} : \omega_1(C_{d,k}) = 0\} \subset \{C_{d,k} \in \mathcal{T} : \omega_2(C_{d,k}) = 0\}.$$

La partition $\mathcal{P}(\mathcal{T}(\omega_1))$ de $\mathcal{X}$ est une sous-partition de $\mathcal{P}(\mathcal{T}(\omega_2))$, notée $\mathcal{P}(\mathcal{T}(\omega_1)) \subset \mathcal{P}(\mathcal{T}(\omega_2))$, au sens de la Définition 7 du Chapitre 1. On dira alors que $\mathcal{T}(\omega_1)$ est un sous-arbre élagué de $\mathcal{T}(\omega_2)$, i.e. un sous-arbre de $\mathcal{T}(\omega_2)$ de même racine, et on le notera $\mathcal{T}(\omega_1) \subseteq \mathcal{T}(\omega_2)$.

Ainsi, l'ensemble de séquences admissibles que l'on note

$$\Omega = \{\omega = (w(C_{d,k}))_{d,k},\, 0 \leq d \leq D \text{ et } 0 \leq k \leq 2^d - 1\},$$

représente la collection des sous-arbres élagués de $\mathcal{T}_D$, ou en d'autres termes, la collection de règles d'ordonnancement $(s_\omega)_{\omega\in\Omega}$ engendrée par $\mathcal{T}_D$, où pour tout $\omega \in \Omega$, la fonction de score $s(\omega)$ est définie par

$$\forall\, x \in \mathcal{X},\; s_\omega(x) = \sum_{C_{d,k}\in\mathcal{P}(\mathcal{T}(\omega))} (2^D - 2^{D-d}k) \cdot \mathbb{I}\{x \in C_{d,k}\}. \tag{3.7}$$

Nous avons schématisé sur la Figure 3.4 ci-dessous, trois exemples de pré-ordres induits sur $\mathcal{X}$ par un arbre d'ordonnancement $\mathcal{T}_D$ et deux sous-arbres élagués de $\mathcal{T}_D$.

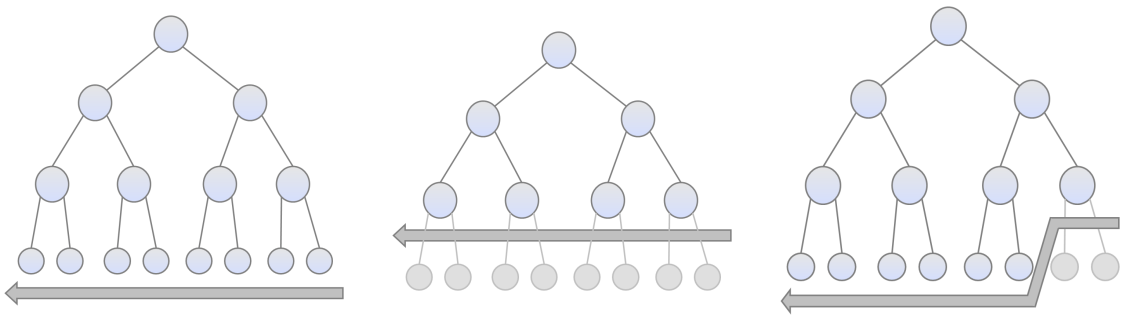


FIGURE 3.4 – Une collection de sous-arbres d'ordonnancement.

En s'appuyant sur cette structure, définir la taille optimale de l'arbre d'ordonnancement $\mathcal{T}_D$ revient à sélectionner une séquence de poids admissible $\omega^* \in \Omega$ telle que

$$\omega^* \in \arg\max_{\omega\in\Omega} \mathrm{ASC}(\mathrm{s}_\omega). \tag{3.8}$$

Dans le contexte de la classification et de la régression, diverses méthodes ont été proposées, comme par exemple dans [Breiman *et al.* 1984] et [Nobel 2002], pour résoudre ce problème spécifique de la sélection de modèles, représentés par des arbres de décision. Ces méthodes reposent sur un principe commun qui consiste à parcourir l'arbre $\mathcal{T}_D$, de ses feuilles terminales vers sa racine, et à *élaguer* ses *branches*, afin de minimiser le risque empirique pénalisé par une mesure de la complexité du modèle, la notion de *branche* étant précisée dans la Définition 11 ci-dessous.

Définition 11 *(Branche d'un arbre)*
Soit $C_{d,k}$, où $d \in \{0, \cdots, D-1\}$ et $k \in \{0, \cdots, 2^d - 1\}$, un noeud interne de $\mathcal{T}_{max}$. La branche $\mathcal{T}_{max}(C_{d,k})$ de $\mathcal{T}_{max}$ issue de $C_{d,k}$ est le sous-arbre de $\mathcal{T}_{max}$ de racine $C_{d,k}$.

Avec la mise en oeuvre d'une telle procédure d'*élagage*, l'apprentissage d'un arbre d'ordonnancement *optimal* se fait en deux temps, de la même façon que pour l'algorithme de classification CART (cf Partie 2.3.1 du Chapitre 2). La première étape de la procédure d'apprentissage consiste à construire un arbre d'ordonnancement $\mathcal{T}_{max}$, le plus grand possible. En règle générale, on poursuit le partitionnement de $\mathcal{X}$ jusqu'à ce que chaque feuille

terminale de $\mathcal{T}_{max}$ ne contienne plus qu'un *petit* nombre d'observations fixé à priori. Puis, la deuxième étape de l'algorithme consiste à sélectionner le plus *petit* sous-arbre de $\mathcal{T}_{max}$ établissant le meilleur compromis biais-variance, *i.e.* le sous-arbre de $\mathcal{T}_{max}$ contenant le moins de feuilles terminales et dont la règle de score associée présente les meilleures performances en termes de prédiction. Nous avons schématisé le principe de cette procédure sur la Figure 3.5 ci-dessous.

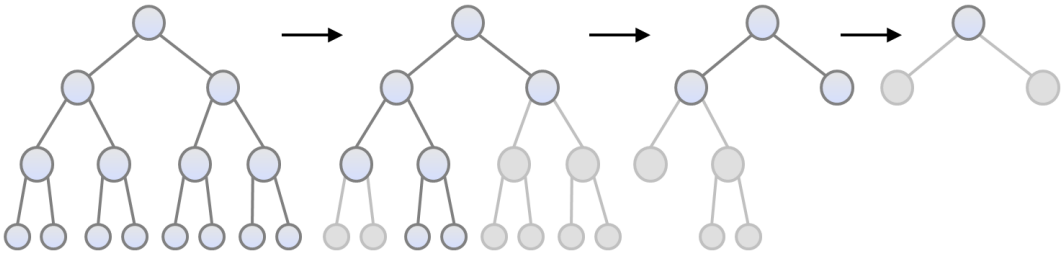


Figure 3.5 – Différentes étapes de la procédure d'élagage.

Dans la suite de ce chapitre, nous présentons deux procédures d'élagage, la première reposant sur l'optimisation de l'ASC empirique pénalisée linéairement par le cardinal de la partition induite sur $\mathcal{X}$ et la seconde sur l'optimisation de l'ASC structurelle, *i.e.* de l'ASC empirique pénalisée par une fonction non linéaire de la complexité du modèle et du nombre d'observations.

3.3 Elagage par optimisation de l'ASC « régularisée »

La première approche, que nous présentons ici, pour définir, de manière automatique et adaptative, la profondeur D d'un arbre d'ordonnancement garantissant de bonnes performances de la règle de prédiction associée, s'inspire de la procédure d'*élagage*, introduite dans [Breiman *et al.* 1984] pour l'algorithme de classification CART. Dans cette partie, nous proposons de transposer directement cette procédure à la problématique d'ordonnancement binaire. Nous renvoyons donc à la référence principale [Breiman *et al.* 1984] ou aux contributions de [Gey 2002] et [Ghattas 2000] pour une description de cette procédure de sélection de modèle dans le contexte de la régression et de la classification.

De manière analogue au cadre de la classification, nous allons voir que l'élagage des *branches* d'un arbre en vue de maximiser l'ASC empirique, pénalisée linéairement par une mesure de la complexité du modèle [1], permet de construire une suite de sous-arbres imbriqués, *suffisante* pour sélectionner le meilleur arbre en termes de performances en prédiction de sa règle de score associée. Nous détaillons ensuite les étapes de la construction de cette suite et décrivons deux procédures de validation, permettant finalement de sélectionner le sous-arbre optimal. Nous considérons ici directement l'analogie avec la sélection de modèle dans le contexte de l'ordonnancement binaire.

1. que l'on précisera ultérieurement

3.3.1 Une pénalisation linéaire

Soit $\mathcal{D}_n$ un échantillon constitué de n copies *i.i.d.* du couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$. On note $\mathcal{T}_{max}$ l'arbre maximal, obtenu par l'application de la procédure TreeRank sur l'échantillon $\mathcal{D}_n$, et $D \geq 1$ sa profondeur. L'ensemble des feuilles terminales $C_{D,k}$ de $\mathcal{T}_{max}$, pour $k \in \{0, \cdots, 2^D - 1\}$, caractérise la fonction de score $s_{max} : \mathcal{X} \to \mathbb{R}$. Cette dernière définit une partition finie disjointe et orientée $\mathcal{P}(\mathcal{T}_{max})$ de $\mathcal{X}$, dont le cardinal, $\#(\mathcal{P}(\mathcal{T}_{max})) = 2^D$, fournit une estimation naturelle de la complexité du modèle représenté par $\mathcal{T}_{max}$.

Soit Ω la collection de poids admissibles induite par $\mathcal{T}_{max}$. Pour tout $\omega \in \Omega$, on note $s(\omega)$ la fonction de score associée au sous-arbre $\mathcal{T}(\omega)$ élagué de $\mathcal{T}_{max}$, définie par l'expression (3.7). La transposition de la procédure d'élagage au cadre de l'ordonnancement consiste à trouver le *modèle* $\omega_\lambda \in \Omega$ maximisant l'ASC pénalisée par la complexité du modèle, notée ASCP_λ, où

$$\begin{aligned} \omega_\lambda &\in \arg\max_{\omega \in \Omega} \widehat{\mathrm{ASCP}}_\lambda(s_\omega) \\ &\in \arg\max_{\omega \in \Omega} \widehat{\mathrm{ASC}}(s_\omega) - \lambda \cdot |\widetilde{\mathcal{T}}(\omega)|, \end{aligned} \tag{3.9}$$

et $\lambda \in \mathbb{R}_+$ est un paramètre de régularisation, que l'on appelle généralement la *température*.

La principale difficulté dans la résolution d'un problème de *régularisation* du type (3.9) réside dans le fait que le paramètre λ est continu et qu'il peut donc prendre une infinité de valeurs. Cependant dans le cadre de la sélection d'un sous-arbre optimal, le problème de la détermination de la température λ revient à choisir la bonne valeur du paramètre parmi une séquence finie $\Lambda = \{\lambda_0, \cdots, \lambda_K\}$, où $K \geq 1$. En effet, il n'existe qu'un nombre fini de sous-arbres de $\mathcal{T}_{max}$ et l'exploration exhaustive de cette collection permet de déterminer la valeur optimale du paramètre de régularisation.

On peut formaliser cette propriété par le Théorème 7 suivant, analogue au Théorème 3.10 dans [Breiman *et al.* 1984], selon lequel un sous-arbre d'ordonnancement $\mathcal{T}(\omega_{\lambda_k})$, $k \in \{0, \cdots, K\}$, associé à la séquence admissible ω_{λ_k}, solution du problème (3.9) pour la température λ_k, est optimal pour une *plage* de valeurs du paramètre λ délimitée par les valeurs *seuil* λ_k et λ_{k+1}. Afin d'alléger les notations dans la suite du chapitre, on notera abusivement pour tout $k \geq 0$, $\mathcal{T}(\omega_{\lambda_k}) = \mathcal{T}^{(k)}$ et $s_{\omega_{\lambda_k}} = s^{(k)}$ la règle de prédiction associée à cet arbre d'ordonnancement.

Théorème 7 *(Une séquence finie de températures)*
Soit une suite strictement croissante de températures $(\lambda_k)_{0 \leq k \leq K}$, $K \geq 1$, *et* $(\mathcal{T}^{(k)})_k$ *la collection de sous-arbres de* $\mathcal{T}_{max}$ *associée à l'ensemble des séquences admissibles* $(\omega_{\lambda_k})_k$, *où*

$$\forall\, k \in \{0, \cdots, K\},\ \omega_{\lambda_k} \in \arg\max_{\omega \in \Omega} \widehat{\mathrm{ASC}}(s_\omega) - \lambda_k \cdot |\widetilde{\mathcal{T}}(\omega)|.$$

Avec ces notations, pour tout $k \leq K - 1$, *si* $\lambda \in [\lambda_k, \lambda_{k+1}[$, *alors* $\mathcal{T}_{\omega_\lambda} = \mathcal{T}^{(k)}$.

Ce résultat théorique reposant sur la *linéarité* de la pénalité et non pas sur la nature du risque considéré dans l'algorithme, sa démonstration est identique à celle proposée dans

[Breiman *et al.* 1984] pour les arbres de classification CART. Nous renvoyons donc à cette référence pour le détail de la preuve.

Toutefois, en règle générale, le nombre de sous-arbres de $\mathcal{T}_{max}$ à considérer est encore trop conséquent et cette approche se révèle trop coûteuse en temps de calcul. C'est là que réside l'intérêt de la procédure d'élagage proposée dans [Breiman *et al.* 1984], car elle permet de construire rapidement une suite de sous-arbres de $\mathcal{T}_{max}$ *emboîtés*, contenant toute l'information nécessaire pour résoudre le problème (3.9). En effet, on déduit de la Proposition 15 ci-dessous (cf Théorème 3.10 dans [Breiman *et al.* 1984]), qu'en résolvant ce problème d'optimisation pour K valeurs croissantes de la température λ, où $K \geq 1$, on obtient une suite de K sous-arbres $(\mathcal{T}^{(k)})_{0\leq k\leq K}$, élagués de $\mathcal{T}_{max}$. De même que précédemment, nous renvoyons à [Breiman *et al.* 1984] pour le détail de la preuve de ce résultat, qui repose sur la linéarité de la pénalité et sur la structure arborescente de la règle de prédiction.

Proposition 15 *(Sous-arbres imbriqués)*
Soient λ_1 et λ_2 deux réels positifs, on note $\mathcal{T}^{(1)}$ et $\mathcal{T}^{(2)}$ les arbres associés aux séquences admissibles ω_{λ_1} et ω_{λ_2} telles que :

$$\forall\, k \in \{1,2\},\ \omega_{\lambda_k} \in \arg\max_{\omega\in\Omega} \widehat{\mathrm{ASC}}(s_\omega) - \lambda_k \cdot |\widetilde{\mathcal{T}}(\omega)|.$$

Avec ces notations, on peut écrire que

$$\lambda_1 \leq \lambda_2 \Rightarrow \mathcal{T}^{(2)} \subseteq \mathcal{T}^{(1)}.$$

Le processus d'élagage, dont les étapes sont détaillées dans la Partie 3.3.2 suivante, repose sur les deux résultats théoriques présentés ci-dessous. Le Lemme 8, analogue à la Proposition 3.8 de [Breiman *et al.* 1984], stipule que l'ASC calculée localement sur un noeud quelconque $C_{d,k}$ de $\mathcal{T}_{max}$, scindé en deux sous-ensembles non vides $C_{d,k} = C_{d+1,2k} \cup C_{d+1,2k+1}$, est toujours inférieure ou égale à la somme des ASC calculées localement sur les enfants $C_{d+1,2k}$ et $C_{d+1,2k+1}$. Ce résultat découle logiquement du fait que le partitionnement du noeud $C_{d,k}$ a pour but de maximiser l'ASC locale : $\mathrm{ASC}_{\mathrm{C_{d,k}}}$ (cf Partie 2.1.1 Chapitre 2).

Lemme 8 *Soit s_{max} la fonction de score définie sur $\mathcal{X}$, associée à l'arbre maximal $\mathcal{T}_{max}$. Supposons que l'on scinde un noeud quelconque $C_{d,k}$ de $\mathcal{T}_{max}$, où $d \in \{0,\cdots,D-1\}$ et $k \in \{0,\cdots,2^d-1\}$, en deux sous-ensembles non vides : $C_{d,k} = C_{d+1,2k} \cup C_{d+1,2k+1}$. Alors on peut écrire*

$$\mathrm{ASC}_{\mathrm{C_{d,k}}}(\mathrm{s_{max}}) \leq \mathrm{ASC}_{\mathrm{C_{d+1,2k}}}(\mathrm{s_{max}}) + \mathrm{ASC}_{\mathrm{C_{d+1,2k+1}}}(\mathrm{s_{max}}), \tag{3.10}$$

où pour un sous-ensemble C quelconque de $\mathcal{X}$, on note $\mathrm{ASC}_{\mathrm{C}}(\mathrm{s_{max}})$ l'aire sous la courbe $\mathrm{COR}(\mathrm{s_{max}})$ calculée localement sur C (cf Définition 9).

Remarque 11 *(*ASC *associée à une branche)*
*Soulignons qu'avec la Définition 9 de l'*ASC *restreinte, introduite dans le Chapitre 2, la somme des termes de droite de l'inégalité (3.10) représente l'*ASC *associée à la branche $\mathcal{T}_{max}(C_{d,k})$ de l'arbre maximal. L'analogie avec la Proposition* 3.8 *de [Breiman et al. 1984] devient alors plus évidente.*

En se basant sur ce résultat et sur la linéarité du critère ASCP, la Proposition 16 ci-dessous montre que, pour toute température λ, on peut trouver un *plus petit sous-arbre optimal*. Ce résultat garantit non seulement l'existence mais aussi l'unicité d'un tel sous-arbre.

Proposition 16 *(Proposition* 3.7 *dans [Breiman et al. 1984])*
Pour tout $\lambda \in \mathbb{R}_+$ il existe un unique sous-arbre $\mathcal{T}(\omega_\lambda)$ élagué de $\mathcal{T}_{max}$ satisfaisant les deux conditions suivantes

(i) $\mathcal{T}(\omega_\lambda) \in \arg\max_{\omega\in\Omega} \widehat{\mathrm{ASCP}}_\lambda(s_\omega)$,

(ii) Si $\exists\, \omega \in \Omega$ *tel que* $\widehat{\mathrm{ASCP}}_\lambda(s_\omega) = \widehat{\mathrm{ASCP}}_\lambda(s_{\omega_\lambda})$, *alors* $\mathcal{T}(\omega_\lambda) \subseteq \mathcal{T}(\omega)$.

Maintenant que les fondements théoriques de la procédure d'élagage ont été présentés, nous pouvons détailler les différentes étapes de la construction de la suite imbriquée de sous-arbres élagués de $\mathcal{T}_{max}$.

3.3.2 Construction d'une suite de sous-arbres optimaux

On initialise le processus en remarquant simplement que, par construction, l'arbre $\mathcal{T}_{max}$ maximise le critère $\widehat{\mathrm{ASCP}}_{\lambda_0}$ défini par

$$\widehat{\mathrm{ASCP}}_0(s) = \widehat{\mathrm{ASC}}(s) - 0 \cdot |\widetilde{\mathcal{T}}|,$$

pour toute fonction de score $s \in \mathcal{S}$ associée à un arbre d'ordonnancement $\mathcal{T}$. La première itération consiste alors à trouver le plus petit sous-arbre $\mathcal{T}(\omega_{\lambda_0})$, élagué de $\mathcal{T}_{max}$, maximisant le critère $\widehat{\mathrm{ASCP}}_{\lambda_0}$.

Une fois que l'on a obtenu le premier sous-arbre $\mathcal{T}(\omega_{\lambda_0}) = \mathcal{T}^{(0)}$ de la suite, on poursuit en élaguant ses branches tout en faisant croître la température λ pour trouver le plus petit sous-arbre $\mathcal{T}^{(1)}$ optimal, associé à la température λ_1. On note $s^{(0)} \in \mathcal{S}_N$, où $N = |\widetilde{\mathcal{T}}^{(0)}|$, la fonction de score associée à l'arbre $\mathcal{T}^{(0)}$, de profondeur $D \geq 1$. Le Lemme 8 nous donne alors que, pour tout noeud interne $C_{d,k}$ de $\mathcal{T}^{(0)}$, où $d \in \{0, \cdots, D-1\}$ et $k \in \{0, \cdots, 2^d - 1\}$

$$\mathrm{ASC}_{\mathrm{C_{d,k}}}(\mathrm{s}^{(0)}) \leq \mathrm{ASC}_{\mathrm{C_{d+1,2k}}}(\mathrm{s}^{(0)}) + \mathrm{ASC}_{\mathrm{C_{d+1,2k+1}}}(\mathrm{s}^{(0)}).$$

Il apparaît clairement que si l'on pénalise l'ASC empirique par une température λ trop faible, l'inégalité

$$\mathrm{ASC}_{\mathrm{C_{d,k}}}(\mathrm{s}^{(0)}) \leq \mathrm{ASC}_{\mathrm{C_{d+1,2k}}}(\mathrm{s}^{(0)}) + \mathrm{ASC}_{\mathrm{C_{d+1,2k+1}}}(\mathrm{s}^{(0)}) - \lambda(|\widetilde{\mathcal{T}}^{(0)}|)$$

reste valide. L'idée est donc de trouver la température seuil λ_1, permettant de *renverser* l'inégalité pour au moins un noeud $C_{d,k} \in \mathcal{T}^{(0)}$. En d'autres termes, on veut trouver la température λ_1 qui conduira à couper au moins une branche de $\mathcal{T}^{(0)}$, sans altérer la performance de la fonction de score finale, *i.e.* l'ASC empirique. On notera $\mathcal{T}^{(1)}$ l'arbre obtenu en élaguant les branches $\mathcal{T}^{(0)}(C_{d,k})$, qui vérifient l'inégalité suivante

$$\mathrm{ASC}_{\mathrm{C_{d,k}}}(\mathrm{s}^{(0)}) \geq \mathrm{ASC}_{\mathrm{C_{d+1,2k}}}(\mathrm{s}^{(0)}) + \mathrm{ASC}_{\mathrm{C_{d+1,2k+1}}}(\mathrm{s}^{(0)}) - \lambda_1(|\widetilde{\mathcal{T}}^{(0)}|).$$

On itère ensuite ce processus jusqu'à sélectionner le dernier arbre de la suite, correspondant à la racine de $\mathcal{T}_{max}$, que l'on notera $\mathcal{T}^{(K)} = C_{0,0}$. Finalement, on obtient une suite

$$C_{0,0} = \mathcal{T}^{(K)} \subseteq \cdots \subseteq \mathcal{T}^{(1)} \subseteq \mathcal{T}^{(0)},$$

de sous-arbres *optimaux* pour la séquence croissante de températures $\{\lambda_0, \cdots, \lambda_K\}$, au sens où, pour tout $k \in \{0, \cdots, K\}$, $\mathcal{T}^{(k)}$ est le plus petit sous-arbre de $\mathcal{T}_{max}$ solution du problème (3.9) pour la température λ_k. En effet, de la même façon que dans le contexte de la classification, on peut montrer que ces sous-arbres vérifient le Théorème 9 suivant, analogue au Théorème 3.10 de [Breiman *et al.* 1984].

Théorème 9 *Soit $\mathcal{T}$ un arbre d'ordonnancement de racine $C_{0,0}$, il existe une suite finie et croissante de constantes $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_K = \infty$ telles que*

$$C_{0,0} = \mathcal{T}^{(K)} \subseteq \ldots \subseteq \mathcal{T}^{(1)} \subseteq \mathcal{T}^{(0)} \subseteq \mathcal{T},$$

et pour tout $k \leq K-1$, on ait

$$\forall\, \lambda \in [\lambda_k, \lambda_{k+1}[,\ \mathcal{T}(\omega_\lambda) = \mathcal{T}^{(k)},$$

où ω_λ est la séquence admissible définie par

$$\omega_\lambda \in \arg\max_{\omega \in \Omega} \widehat{\mathrm{ASC}}(s_\omega) - \lambda \cdot |\widetilde{\mathcal{T}}(\omega)|.$$

De plus, il découle de ce théorème le Corollaire 10 suivant, qui montre que la suite de sous-arbres obtenue par la procédure d'élagage est suffisante pour représenter toute l'information de l'arbre $\mathcal{T}_{max}$ et que l'on peut donc choisir le meilleur modèle parmi cette collection.

Corollaire 10 *Soit $(\lambda_k)_{0 \leq k \leq K}$, $K \geq 1$, une suite de réels positifs strictement croissante associée à la collection de sous-arbres imbriqués $(\mathcal{T}^{(k)})_{0 \leq k \leq K}$. Les deux propriétés suivantes sont vérifiées :*

(i) $\forall\, \lambda \in \mathbb{R}_+,\ \exists\, k \in \{0, \cdots, K\}$ *tel que* $\mathcal{T}(\omega_\lambda) = \mathcal{T}^{(k)}$,

(ii) $\forall\, k \in \{0, \cdots, K\}$, *si* $|\widetilde{\mathcal{T}}^{(k)}| = n_k$ *alors*

$$\mathcal{T}^{(k)} \in \arg\max_{\omega \in \Omega,\, |\widetilde{\mathcal{T}}(\omega)| = n_k} \widehat{\mathrm{ASC}}.$$

3.3.3 Sélection du sous-arbre optimal par validation

Une fois que l'on dispose de la suite $(\mathcal{T}^{(k)})_{0 \leq k \leq K}$, $K \geq 1$, de sous-arbres optimaux de $\mathcal{T}_{max}$, il ne reste plus qu'à sélectionner le meilleur sous-arbre parmi cette collection. Deux approches sont considérées dans [Breiman *et al.* 1984] : l'utilisation d'un échantillon de validation, indépendant des données d'apprentissage ayant servi à la construction de la suite $(\mathcal{T}^{(k)})_k$, ou la mise en oeuvre d'une procédure de validation croisée par blocs.

Si on dispose de suffisamment de données, on peut effectivement décider de diviser l'échantillon $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$ initial en deux sous-échantillons $\mathcal{D}_n^a$, servant à construire la suite $(\mathcal{T}_k)_{1 \leq k \leq K}$, et $\mathcal{D}_n^v$, utilisé pour sélectionner le meilleur sous-arbre parmi cette collection, au sens des performances en prédiction des règles de score associées. Pour ce faire, on ordonne les observations de $\mathcal{D}_n^v$ selon chaque sous-arbre de la suite et on identifie l'arbre $\mathcal{T}_{\omega^*}$ satisfaisant la condition suivante

$$\omega^* \in \arg\max_{\omega \in (\omega_{\lambda_k})_k} \widehat{\mathrm{ASC}}^v(s_\omega),$$

où $\widehat{\mathrm{ASC}}^v$ est l'ASC empirique de la règle s_ω, calculée sur l'échantillon de validation $\mathcal{D}_n^v$.

Si par contre, on ne dispose pas de suffisamment de données, on procède différemment. Tout d'abord, on divise l'échantillon $\mathcal{D}_n$ en V blocs $(\mathcal{D}_n^v)_{1\leq v\leq V}$, $V \geq 2$, de tailles équivalentes. Puis, on calcule, sur chaque échantillon $\mathcal{D}_n^v$, l'ASC empirique de validation $(\widehat{\mathrm{ASC}}^v)$ associée à chacun des sous-arbres de la suite $(\mathcal{T}^{(k)})_{0\leq k\leq K}$. On sélectionne finalement le modèle ω^* tel que

$$\omega^* \in \underset{\omega\in(\omega_{\lambda_k})_k}{\arg\max} \frac{1}{V}\sum_{v=1}^{V} \widehat{\mathrm{ASC}}^v(s_\omega).$$

Toutefois, pour que cette procédure soit cohérente, il faut reconstituer la suite $(\mathcal{T}^{(k)})_{0\leq k\leq K}$ pour chaque échantillon d'apprentissage $\mathcal{D}_n - \mathcal{D}_n^v$, où $v \in \{1, \cdots, V\}$. En effet, si l'on ne procède pas de la sorte, les échantillons $\mathcal{D}_n^v$ ne peuvent pas être considérés comme des échantillons de validation pour la suite de sous-arbres.

Dans ce cas, la procédure consiste donc à construire V suites $(\mathcal{T}^{(k,v)})_{k,v}$, où $k \in \{0, \cdots, K\}$ et $v \in \{1, \cdots, V\}$, afin d'estimer les ASC de validation des K sous-arbres par une approche de type Monte-Carlo. Autrement dit, l'objectif est de calculer la moyenne empirique des ASC de validation associées aux V estimateurs $(\mathcal{T}^{(k,v)})_v$ de chaque sous-arbres $\mathcal{T}^{(k)}$, $k \in \{0, \cdots, K\}$. Pour cela il faut toutefois s'assurer que les estimateurs $(\mathcal{T}^{(k,v)})_v$ soient suffisamment *proches* des sous-arbres $\mathcal{T}^{(k)}$ de $\mathcal{T}_{max}$. Une solution à ce problème consiste à procéder par étapes, comme suit : on construit tout d'abord la suite $(\mathcal{T}^{(k)})_{0\leq k\leq K}$ de sous-arbres optimaux à partir de l'échantillon $\mathcal{D}_n$, selon la procédure décrite précédemment. Puis, une fois que la séquence de températures $(\lambda_k)_k$ est identifiée, on reconstitue une suite $(\mathcal{T}^{(k,v)})_k$ à partir de chaque échantillon $\mathcal{D}_n - \mathcal{D}_n^v$, $v \in \{1, \cdots, V\}$, en imposant les valeurs de la séquence de températures $(\lambda_k^v)_k$, de sorte à ne pas sélectionner de sous-arbres $(\mathcal{T}^{(k,v)})_v$ trop différents des $\mathcal{T}^{(k)}$ d'origine, appris sur $\mathcal{D}_n$. Plus précisément, on fixe les valeurs $(\lambda_k^v)_k$ suivantes

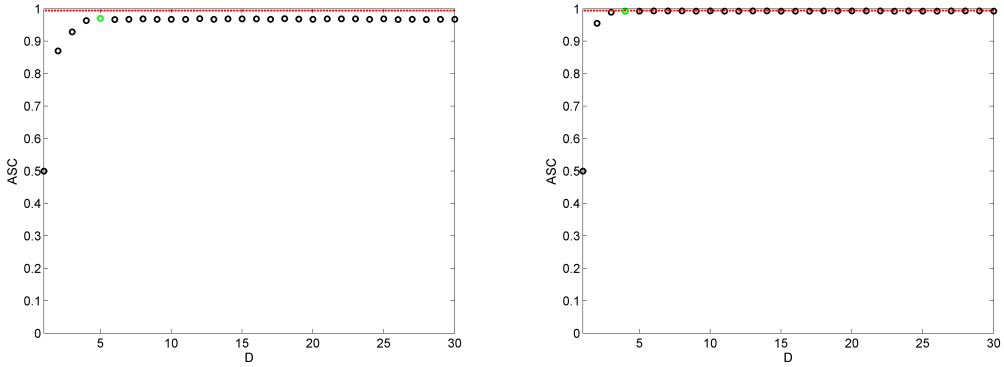
$$\forall\, k \in \{0, \cdots, K\},\ \lambda_k^v = \sqrt{\lambda_k \lambda_{k+1}},$$

qui reviennent à prendre comme température λ_k^v la moyenne géométrique des valeurs λ_k et $\lambda_k + 1$, obtenues en élaguant $\mathcal{T}_{max}$ sur l'échantillon d'apprentissage $\mathcal{D}_n$.

La principale limite de la méthode que nous venons de décrire réside dans l'absence de résultats théoriques, concernant notamment la consistance des règles de prédiction sélectionnées. On peut noter tout de même que des résultats, relatifs à l'utilisation de cette procédure d'élagage dans le contexte de la classification binaire, ont pu être établis dans [Nobel 2002], pour une pénalité de la forme $\sqrt{|\widetilde{\mathcal{T}}|}$. Par ailleurs, une étude des propriétés théoriques de cette approche est aussi proposée dans la contribution [Gey 2002], qui valide notamment le choix de la pénalité $|\widetilde{\mathcal{T}}|$ dans le cadre de la régression.

Néanmoins, cette procédure reste très intéressante car elle présente le grand avantage de pouvoir être implémentée facilement. Aussi, nous avons pu étudier ses performances empiriques, en la mettant en oeuvre pour sélectionner la profondeur des arbres d'ordonnancement, obtenus au moyen des diverses versions de l'heuristique TreeRank. Pour illustrer notre propos, reprenons l'exemple *GaussCroix2d* introduit précédemment. De même que dans la Partie 2.3.3.2 du Chapitre 2, les graphes a et b de la Figure 3.6 suivante, montrent l'évolution de l'ASC empirique calculée sur l'échantillon test, en fonction de la profondeur $D \geq 1$ des procédures d'apprentissage $\mathrm{TRK}_{\mathrm{CART}}$ et $\mathrm{TRK}_{\mathrm{SVM}}$ respectivement. Sur chacun

de ces graphes, nous avons représenté en vert l'ASC empirique de test associée à la règle de score sélectionnée par la procédure d'élagage que nous venons de décrire. On constate que sur ces données, cette procédure permet d'établir un bon compromis entre performances et complexité de la règle de prédiction.



a. Heuristique TRK_{CART} et élagage. b. Heuristique TRK_{SVM} et élagage.

Figure 3.6 – Ex. *GaussCroix2d* - Evolution de l'ASC empirique avec la profondeur D.

De manière générale, les règles de score sélectionnées par cette procédure présentent de bonnes performances en termes de prédictions, et une complexité *raisonnable*. On peut cependant se poser la question de la pertinence de la forme linéaire de la pénalisation. En effet, la représentation arborescente de la règle de prédiction implique que plus la profondeur D augmente, plus le nombre d'observations dans les cellules de l'arbre diminue. Or, la forme de la pénalité ne permet pas de s'adapter à ce phénomène, inhérent à la structure hiérarchique de la procédure d'apprentissage. Aussi, nous considérons dans la partie suivante, des pénalités non linéaires s'écrivant comme des fonctions de la complexité du modèle et du nombre d'observations.

3.4 Elagage par optimisation de l'ASC structurelle

Dans cette partie, nous considérons le problème de la sélection de modèle du point de vue de l'optimisation de l'ASC *structurelle*, au sens de l'approche proposée dans [Vapnik 1982]. Les résultats présentés dans ce chapitre reposent sur l'écriture de l'ASC sous la forme d'une U-statistique (cf Partie 1.2.2.1 du Chapitre 1) et sur l'application des résultats de la théorie des U-processus ([Peña & Giné 1999], [Serfling 1980]). Ainsi, en nous appuyant sur les résultats, introduits dans [Clémençon *et al.* 2005b] et [Clémençon *et al.* 2008], relatifs au contrôle de la déviation uniforme entre le critère ASC et sa contrepartie empirique, nous proposons deux pénalités, pour deux configurations de l'heuristique TreeRank, conduisant à la sélection de règles de prédictions asymptotiquement *optimales*. Nous montrons de plus que ces deux pénalités sont *optimales* d'un point de vue non-asymptotique, en établissant une *inégalité oracle* pour les fonctions de score obtenues via cette procédure de sélection.

3.4.1 Une pénalisation non-linéaire

De même que précédemment, on considère un arbre d'ordonnancement $\mathcal{T}_{max}$, de profondeur $D \geq 1$, appris sur un échantillon $\mathcal{D}_{=}\{(X_i, Y_i),\, 1 \leq i \leq n\}$ d'observations de $\mathcal{X} \times \{-1, +1\}$, définissant une fonction de score s_{max} sur $\mathcal{X}$. On note Ω la collection de séquences admissibles engendrées par $\mathcal{T}_{max}$. Avec ces notations, le problème de l'optimisation de l'ASC structurelle revient à sélectionner la séquence admissible $\omega^* \in \Omega$ telle que

$$\begin{aligned} \omega^* &\in \arg\max_{\omega\in\Omega} \widehat{\mathrm{ASCP}}(s_\omega) \\ &\in \arg\max_{\omega\in\Omega} \widehat{\mathrm{ASC}}(s_\omega) - \mathrm{pen}(|\widetilde{\mathcal{T}}(\omega)|, n), \end{aligned} \tag{3.11}$$

où la pénalité $\mathrm{pen}(K, n)$ est une fonction de la complexité $K = |\widetilde{\mathcal{T}}(\omega)| \geq 1$ de la règle de prédiction s_ω et du nombre n d'observations. L'intérêt de cette nouvelle approche est qu'à la différence de la pénalisation linéaire considérée précédemment, qui dépendait d'un paramètre de régularisation $\lambda \in \mathbb{R}_+$, la quantité $\mathrm{pen}(K, n)$ est fixée et peut être explicitée, si bien qu'aucune procédure de ré-échantillonnage ou de validation n'est nécessaire pour sélectionner le meilleur modèle, comme c'était le cas dans la procédure d'élagage.

Dans ce chapitre, nous proposons d'estimer deux pénalités indépendantes de la distribution des observations de $\mathcal{X} \times \{-1, +1\}$, en calculant une borne supérieure pour la quantité suivante :

$$\mathbb{E}\left[\sup_{\omega\in\Omega:\, |\widetilde{\mathcal{T}}(\omega)|=K} |\widehat{\mathrm{ASC}}(s_\omega) - \mathrm{ASC}(\mathrm{s}_\omega)|\right],$$

où $K \in \{1, \ldots, 2^D\}$.

3.4.2 Deux exemples de pénalités

Nous allons considérer deux configurations de l'algorithme TreeRank, associées à la mise en oeuvre des deux heuristiques LeafRank suivantes, pour le partitionnement des noeuds de $\mathcal{T}_{max}$ à chaque itération :

- $(\mathcal{O}_1)$: la scission d'un noeud $C_{d,k}$ de $\mathcal{T}_{max}$ en deux sous-ensembles non vides est obtenue par la mise en oeuvre d'une version pondérée de l'algorithme CART, comme décrit dans la Partie 2.3.1 du Chapitre 2, avec au plus $\kappa \geq 1$ partitions perpendiculaires aux axes de l'espace d'entrée $\mathcal{X}$,

- $(\mathcal{O}_2)$: en considérant l'espace d'entrée $\mathcal{X} = [0, 1]^q$, la scission d'un noeud $C_{d,k}$ de $\mathcal{T}_{max}$ repose sur la définition à priori d'une partition de $C_{d,k}$ (cf Partie 2.2.1 du Chapitre 2), constituée de cubes dyadiques de la forme

$$\prod_{l=1}^{q} \left[\frac{k_l}{2^m}, \frac{k_l + 1}{2^m}\right[, \text{ où } 0 \leq k_l < 2^m, \text{ pour tout } l \in \{1, \ldots, q\}.$$

Dans la Proposition 17 suivante, nous proposons une pénalité pour les deux configurations pré-citées, qui permettent d'établir une inégalité oracle pour les règles de prédictions s_{ω^*} solution du problème (3.11).

Proposition 17 *(Inégalités oracle)*
Supposons que le taux théorique p de positifs appartienne à un intervalle $[\underline{p}, \bar{p}]$, *où* $0 < \underline{p} < \bar{p} < 1$. *Pour tout* $K \in \{1, \ldots, 2^D\}$ *et* $n \geq 1$, *on choisit le terme de pénalité comme suit, en fonction de la stratégie utilisée pour résoudre l'étape d'optimisation de l'algorithme* TreeRank.

(*i*) *Si les scissions sont optimisées selon la règle* ($\mathcal{O}_1$), *on pose :* $\forall\, \kappa \in \mathbb{N}^*$,

$$pen(K,n) \;=\; \frac{1}{\underline{p}(1-\bar{p})}\sqrt{32 \cdot \frac{\log\left(16((n+1)q)^{2K\kappa}\right) + K}{n}}.$$

(*ii*) *Si les scissions sont optimisées selon la règle* ($\mathcal{O}_2$), *on pose :* $\forall\, m \in \mathbb{N}^*$,

$$pen(K,n) \;=\; \frac{1}{\underline{p}(1-\bar{p})}\sqrt{\frac{\log\left(4K^{2mq}\right) + K}{2n}}.$$

Pour ces deux pénalités, il existe une constante positive C telle que, l'espérance du déficit en ASC, *associé au sous-arbre* $\mathcal{T}(\omega^*)$, *où la séquence* ω^* *maximise l'*ASC *empirique pénalisée, est bornée comme suit :*

$$\mathrm{ASC}^* - \mathbb{E}\left[\mathrm{ASC}(\mathrm{s}_{\omega^*})\right] \leq \inf_{1 \leq \mathrm{K} \leq 2^{\mathrm{D}}} \left\{ \mathrm{C} \cdot pen(\mathrm{K},\mathrm{n}) + \left\{ \mathrm{ASC}^* - \sup_{\omega \in \Omega:\, |\widetilde{\mathcal{T}}(\omega)| = \mathrm{K}} \mathrm{ASC}(\mathrm{s}_\omega) \right\} \right\}. \tag{3.12}$$

Afin de justifier le choix de ces deux pénalités, nous allons montrer que les règles de score auxquelles elles conduisent vérifient bien l'inégalité oracle (3.12). Pour établir cette inégalité, nous introduisons tout d'abord le Lemme 11 suivant, que nous démontrons ci-après.

Soit $K \geq 1$, on note $\mathbf{P}_K(\mathcal{T})$ la collection des partitions induites sur l'espace d'entrée $\mathcal{X} \subset \mathbb{R}^q$ par des arbres d'ordonnancement, constituées de $K \geq 1$ cellules non vides, et $\mathcal{S}_K = \bigcup_{\mathcal{P} \in \mathbf{P}_K(\mathcal{T})} \mathcal{S}_{\mathcal{P}}$ l'ensemble des fonctions de score constantes par morceaux associées à celles-ci. On introduit aussi la règle de score $s_{\omega^*(K)} \in \mathcal{S}_K$ d'ASC empirique maximale, apprise sur un échantillon d'apprentissage $\mathcal{D}_n = \{(X_i, Y_i),\, 1 \leq i \leq, n\}$:

$$s_{\omega^*(K)} \in \operatorname*{arg\,max}_{s_{\omega(K)} \in \mathcal{S}_K} \widehat{\mathrm{ASC}}(s_{\omega(K)}).$$

Lemme 11 *Supposons que les hypothèses de la Proposition 17 soient satisfaites.*

(*i*) *Si les scissions sont optimisées selon la règle* ($\mathcal{O}_1$) *et que la pénalité est choisie en conséquence, alors :* $\forall (K,\kappa) \in \mathbb{N}^{*2}$,

$$\begin{aligned}\mathbb{P}\left\{ \sup_{s_{\omega(K)} \in \mathcal{S}_K} \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})}) - \mathrm{ASC}(\mathrm{s}_{\omega^*}) \geq \epsilon \right\} \;&\leq\; 16\left((n+1)q\right)^{2K\kappa} e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/512} \\ &+\; e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128}.\end{aligned}$$

(*ii*) *Si les scissions sont optimisées selon la règle* ($\mathcal{O}_2$) *et que la pénalité est choisie en conséquence, alors :* $\forall (K,m) \in \mathbb{N}^{*2}$,

$$\mathbb{P}\left\{\sup_{s_{\omega(K)}\in\mathcal{S}_K}\mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})})-\mathrm{ASC}(\mathrm{s}_{\omega^*})\geq\epsilon\right\} \leq 4K^{2^{mq}}e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/8} + e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/2}.$$

La démonstration de ce lemme s'inspire de la démarche proposée dans [Lugosi & Zeger 1996] (voir aussi la Section 18.1 de [L.Devroye *et al.* 1996]). Elle repose sur le contrôle de la complexité de la collection de modèles Ω au moyen de la dimension VC et sur celui de la déviation entre l'ASC et sa contrepartie empirique, grâce aux résultats de la théorie des U-processus.

Preuve 3 *(Preuve du Lemme 11) Pour tout $\epsilon>0$ et tout $K\geq 1$, on a*

$$\mathbb{P}\{\sup_{s_{\omega(K)}\in\mathcal{S}_K}\mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})})-\mathrm{ASC}(\mathrm{s}_{\omega^*}) \geq \epsilon\} \leq \mathbb{P}\{\sup_{l\geq 1}\widehat{\mathrm{ASCP}}(s_{\omega^*(l)}-\mathrm{ASC}(\mathrm{s}_{\omega^*})\geq\frac{\epsilon}{2}\} + \mathbb{P}\{\sup_{s_{\omega(K)}\in\mathcal{S}_K}\mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})})-\sup_{\mathrm{l}\geq 1}\widehat{\mathrm{ASCP}}(\mathrm{s}_{\omega^*(\mathrm{l})}\geq\frac{\epsilon}{2}\}.$$

On peut reformuler et borner le premier terme du membre de droite de cette inégalité comme suit :

$$\begin{aligned}\mathbb{P}\{\widehat{\mathrm{ASCP}}(s_{\omega^*})-\mathrm{ASC}\quad(s_{\omega^*}) &\geq \frac{\epsilon}{2}\}\\ &\leq \mathbb{P}\{\inf_{l\geq 1}\{\widehat{\mathrm{ASCP}}(\widehat{s}_l^*)-\mathrm{ASC}(\widehat{\mathrm{s}}_\mathrm{l}^*)\}\geq\frac{\epsilon}{2}\}\\ &\leq \sum_{l\geq 1}\mathbb{P}\{\left|\mathrm{ASC}(\widehat{\mathrm{s}}_\mathrm{l}^*)-\widehat{\mathrm{ASC}}(\widehat{\mathrm{s}}_\mathrm{l}^*)\right|\geq\frac{\epsilon}{2}+pen(l,n)\}\\ &\leq \sum_{l\geq 1}\mathbb{P}\{\sup_{s\in\mathcal{S}_K}\left|\mathrm{ASC}(\mathrm{s})-\widehat{\mathrm{ASC}}(\mathrm{s})\right|\geq\frac{\epsilon}{2}+pen(l,n)\}. \quad (3.13)\end{aligned}$$

Considérons maintenant le second terme, on remarque que

$$\begin{aligned}\mathbb{P}\{\sup_{s\in\mathcal{S}_K}\mathrm{ASC}(\mathrm{s})-\sup_{\mathrm{l}\geq 1}\widehat{\mathrm{ASCP}}(\widehat{\mathrm{s}}_\mathrm{l}^*) &\geq \frac{\epsilon}{2}\}\\ &\leq \mathbb{P}\{\sup_{s\in\mathcal{S}_K}\mathrm{ASC}(\mathrm{s})-\widehat{\mathrm{ASCP}}(\widehat{\mathrm{s}}_\mathrm{K}^*)\geq\frac{\epsilon}{2}\}\\ &\leq \mathbb{P}\{\sup_{s\in\mathcal{S}_K}\mathrm{ASC}(\mathrm{s})-\widehat{\mathrm{ASC}}(\widehat{\mathrm{s}}_\mathrm{K}^*)\geq\frac{\epsilon}{4}\}\\ &\leq \mathbb{P}\{\sup_{s\in\mathcal{S}_K}\left|\widehat{\mathrm{ASC}}(s)-\mathrm{ASC}(\mathrm{s})\right|\geq\frac{\epsilon}{4}\}, \quad (3.14)\end{aligned}$$

en supposant que $pen(K,n)\leq\epsilon/4$. En s'appuyant sur ces deux résultats, on va pouvoir établir une borne plus précise pour la queue de probabilité de la quantité

$$\sup_{s_{\omega(K)}\in\mathcal{S}_K}\left|\widehat{\mathrm{ASC}}(s_{\omega(K)})-\mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})})\right|,$$

pour les deux configurations ($\mathcal{O}_1$) et ($\mathcal{O}_2$).

Plaçons-nous tout d'abord dans le cas où les scissions sont optimisées selon la règle ($\mathcal{O}_1$), i.e. au moyen d'une version pondérée de l'algorithme CART*, avec au plus κ partitions perpendiculaires aux axes de $\mathcal{X}$. On va considérer l'approche développée dans le*

contexte de la minimisation du risque empirique d'ordonnancement, introduite dans [Clémençon et al. 2008]. Rappelons notamment le résultat suivant, basé sur la représentation de Hoeffding pour les U-statistiques ([Hoeffding 1948]).

Lemme 12 *(Lemme A1 de [Clémençon et al. 2008])*
Soit une famille de fonctions à valeurs réelles, $h_\tau : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, indexées par $\tau \in T$, où T est un ensemble donné. Soient $(X_1, \ldots, X_n)$, n copies i.i.d. d'une variable aléatoire $X \in \mathcal{X}$. Pour toute fonction convexe ψ croissante, on a

$$\mathbb{E}\left[\psi\left(\sup_{\tau\in T}\frac{1}{n(n-1)}\sum_{i\neq j}h_\tau(X_i,X_j)\right)\right] \leq \mathbb{E}\left[\psi\left(\sup_{\tau\in T}\frac{1}{\lfloor n/2\rfloor}\sum_{i=1}^{\lfloor n/2\rfloor}h_\tau(X_i,X_{\lfloor n/2\rfloor+i})\right)\right],$$

à condition que les suprema soient mesurables et que les espérances existent, avec $\lfloor n/2\rfloor$ la partie entière de $n/2$.

Considérons maintenant le $n^{ème}$ coefficient de pulvérisation de la classe de sous-ensembles de $\mathcal{X} \times \mathcal{X}$

$$\mathcal{A} = \bigcup_{\mathcal{P}\in\mathbf{P}_K}\{A \times B : (A, B) \in \mathcal{P}^2\},$$

qui s'écrit

$$\begin{aligned}\mathrm{S}(\mathcal{A}, n) &= \max_{\{X_1,\cdots,X_n\}} \mathcal{N}_{\mathcal{A}},\\ &= \max_{\{X_1,\cdots,X_n\}} \#\{\{X_1, \cdots, X_n\} \cap C \,:\, C \in \mathcal{A}\}.\end{aligned}$$

Ce coefficient peut être borné comme suit :

$$\mathrm{S}(\mathcal{A}, n) \leq ((n+1)q)^{2K\kappa}. \tag{3.15}$$

Considérons de plus une fonction de score $s_{\omega(K)} \in \mathcal{S}_K$ et le noyau $h_{s_{\omega(K)}} - U(s_{\omega(K)})$ où

$$\begin{aligned}U(s_{\omega(K)}) &= 2p(1-p)\mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})}),\\ &= \mathbb{E}\left[\widehat{U}_n(s_{\omega(K)})\right],\end{aligned}$$

est l'espérance de la U-statistique d'ordre 2 donnée par

$$\widehat{U}_n(s_{\omega(K)}) = \frac{2}{n(n-1)}\sum_{1\leq i<j\leq n} h_{s_{\omega(K)}}((X_i, Y_i), (X_j, Y_j)),$$

et $h_{s_{\omega(K)}}$ est le noyau symétrique borné défini par

$$\begin{aligned}h_{s_{\omega(K)}}((X,Y),(X'Y')) &= \mathbb{I}\{(Y - Y')(s_{\omega(K)}(X) - s_{\omega(K)}(X') > 0)\}\\ &+ \frac{1}{2}\mathbb{I}\{(s_{\omega(K)}(X) = s_{\omega(K)}(X'), Y \neq Y')\},\end{aligned}$$

pour tous couples (X, Y) et (X', Y') i.i.d..

En appliquant le Lemme 12 à la collection de noyaux $\{h_{s_{\omega(K)}} - U(s_{\omega(K)})\}_{s_{\omega(K)}\in\mathcal{S}_K}$ et en

combinant le résultat ainsi obtenu avec l'inégalité de Vapnik-Chervonenkis ([Vapnik 1998]) et la borne (3.15) du coefficient de pulvérisation, on obtient : $\forall\epsilon$, $\forall n \geq 1$,

$$\mathbb{P}\left\{\sup_{s_{\omega(K)}\in\mathcal{S}_K} |\widehat{U}_n(s_{\omega(K)}) - U(s_{\omega(K)})| \geq \epsilon\right\} \leq 8\left((n+1)q\right)^{2K\kappa} e^{-n\epsilon^2/32}.$$

Ainsi, pour une valeur de n suffisamment grande, on a

$$\mathbb{P}\left\{\sup_{s_{\omega(K)}\in\mathcal{S}_K} |\widehat{\mathrm{ASC}}(s_{\omega(K)}) - \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})})| \geq \epsilon\right\} \leq 16\left((n+1)q\right)^{2K\kappa} e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/32}, \quad (3.16)$$

le facteur multiplicatif additionnel dans la borne ci-dessus s'expliquant par la prise en compte des fluctuations du taux empirique de positifs dans l'échantillon autour de la proportion p, pour n assez grand. En combinant ce résultat avec l'inégalité (3.13), on obtient

$$\begin{aligned}
\mathbb{P}\{\widehat{\mathrm{ASCP}}(s_{\omega^*}) - \mathrm{ASC}(\mathrm{s}_{\omega^*}) &\geq \frac{\epsilon}{2}\} \\
&\leq \sum_{l=1}^{2^D} 16((n+1)q)^{2K\kappa} e^{-n\underline{p}^2(1-\bar{p})^2(\frac{\epsilon}{2}+pen(K,n))^2/32}, \\
&\leq e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128} \sum_{l=1}^{2^D} 16((n+1)q)^{2\kappa} e^{-n\underline{p}^2(1-\bar{p})^2 pen(K,n)^2/32}, \\
&\leq e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128} \sum_{l\geq 1} e^{-K}, \\
&\leq e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128}, \qquad (3.17)
\end{aligned}$$

où pen(K,n) *est remplacée par son expression explicite dans la configuration* $(\mathcal{O}_1)$*. Par ailleurs, en combinant les inégalités (3.16) et (3.14), on peut écrire que*

$$\begin{aligned}
\mathbb{P}\{\sup_{s_{\omega(K)}\in\mathcal{S}_K} \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})}) &- \sup_{l\geq 1} \widehat{\mathrm{ASCP}}(s_{\omega^*(l)}) \geq \frac{\epsilon}{2}\} \\
&\leq 16((n+1)q)^{2K\kappa} e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/512}. \qquad (3.18)
\end{aligned}$$

Finalement, la combinaison des deux inégalités (3.17) et (3.18) prouve la première assertion du Lemme 11.

Supposons maintenant que $\mathcal{X} = [0,1]^q$ *et plaçons-nous dans la configuration* $(\mathcal{O}_2)$*, dans laquelle les cellules de la partition de* $\mathcal{X}$ *sont obtenues par réunion de cubes dyadiques de côté* 2^{-m}*,* $m \in \mathbb{N}$*. Dans ce cas précis, en combinant une version de l'inégalité de Hoeffding pour les U-statistiques (cf Théorème A de la Section* 5.6 *de [Serfling 1980]) avec la borne d'union et le fait que* $\#\{h_{s_{\omega(K)}} : s_{\omega(K)} \in \mathcal{S}_K\} \leq K^{2^{mq}}$*, on obtient,* $\forall\epsilon$, $\forall n \geq 1$,

$$\mathbb{P}\left\{\sup_{s_{\omega(K)}\in\mathcal{S}_K} \left|\widehat{U}_n(s_{\omega(K)}) - U(s_{\omega(K)})\right| \geq \epsilon\right\} \leq 2K^{2^{mq}} e^{-2n\epsilon^2},$$

et pour n assez grand on a :

$$\mathbb{P}\left\{\sup_{s_{\omega(K)}\in\mathcal{S}_K} \left|\widehat{\mathrm{ASC}}(s_{\omega(K)}) - \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})})\right| \geq \epsilon\right\} \leq 4K^{2^{mq}} e^{-2n\underline{p}^2(1-\bar{p})^2\epsilon^2}.$$

En réitérant le même cheminement que précédemment, on prouve la deuxième assertion du lemme. Nous ne détaillons pas les calculs, qui sont en tout point similaires à ceux effectués pour la configuration ($\mathcal{O}_1$).

Maintenant que les assertions du Lemme 11 ont été établies, on peut revenir à la démonstration de l'inégalité oracle de la Proposition 17.

Preuve 4 *(Preuve de la Proposition 17)*
On peut écrire assez simplement que

$$\begin{aligned} \mathrm{ASC}^* - \mathbb{E}\left[\mathrm{ASC}(\mathrm{s}_{\omega^*})\right] &= \inf_{K\geq 1}\{(\mathrm{ASC}^* - \sup_{\mathrm{s}_{\omega(\mathrm{K})}\in\mathcal{S}_{\mathrm{K}}} \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})})) \\ &+ (\sup_{s_{\omega(K)}\in\mathcal{S}_K} \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})}) - \mathbb{E}\left[\mathrm{ASC}(\mathrm{s}_{\omega^*})\right])\}. \end{aligned}$$

Il en découle que

$$\begin{aligned} (\sup_{s_{\omega(K)}\in\mathcal{S}_K} \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})}) &- \mathbb{E}\left[\mathrm{ASC}(\mathrm{s}_{\omega^*})\right])^2 \\ &\leq u + \int_{t=u}^{\infty} \mathbb{P}\{\left(\sup_{s_{\omega(K)}\in\mathcal{S}_K} \mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{K})}) - \mathrm{ASC}(\mathrm{s}_{\omega^*})\right)^2 > t\}dt. \end{aligned}$$

Finalement, l'inégalité oracle (3.12) est obtenue en intégrant les bornes données dans le Lemme 11 et en posant $u = C\left(pen(K,n)\right)^2$, où la pénalité est explicitée selon la configuration considérée.

3.4.3 Règles de score consistantes

Le résultat suivant découle directement de la Proposition 17 précédente. Il montre la consistance des règles de score solutions du problème (3.11) pour les deux pénalités proposées, sous réserve que certaines hypothèses soient satisfaites.

Corollaire 13 *Supposons que les hypothèses de la Proposition 17 soient satisfaites et qu'il existe une séquence $(\mathcal{T}(\omega(n)))_{\omega(n)}$ de sous-arbres de $\mathcal{T}_{max}$, obtenus par le biais de l'algorithme* TreeRank *sur un échantillon d'apprentissage $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$ et tels que*

$$\mathbb{E}\left[\mathrm{ASC}(\mathrm{s}_{\omega(\mathrm{n})})\right] \to_{n\to\infty} \mathrm{ASC}^*.$$

Supposons de plus que :

(i) $\mathcal{T}_{max}$ est construit selon la règle $\mathcal{O}_1$, avec $\kappa = \kappa(n)$ scissions perpendiculaires, alors

$$\kappa(n) \cdot \mathbb{E}\left[|\widetilde{\mathcal{T}}(\omega(n))|\right] = o(\frac{n}{\log n}) \text{ quand } n \to \infty,$$

(ii) $\mathcal{T}_{max}$ est construit selon la règle $\mathcal{O}_2$, basée sur des hypercubes dyadiques de côté 2^{-m} avec $m = m(n)$, alors

$$\mathbb{E}\left[|\widetilde{\mathcal{T}}(\omega(n))|\right] = o(n) \text{ et } m(n) = o(\frac{n}{\log n}) \text{ quand } n \to \infty.$$

*Sous ces conditions, la règle de score $s_{\omega^*(n)}$ basée sur la maximisation de l'*ASC *structurelle (3.11) est consistante au sens où :*

$$\lim_{n\to\infty} \mathbb{E}\left[\mathrm{ASC}(\mathrm{s}_{\omega^*(\mathrm{n})})\right] = \mathrm{ASC}^*.$$

Ainsi, nous avons proposé deux pénalités permettant de sélectionner des règles de score asymptotiquement optimales et satisfaisant une inégalité oracle, lorsque l'*étape d'optimisation* de la procédure TreeRank est résolue au moyen d'une version pondérée de l'algorithme CART (configuration ($\mathcal{O}_1$)) ou à partir d'une partition uniforme et dyadique, fixée à priori, (configuration ($\mathcal{O}_2$). Toutefois, l'approche que nous avons proposée présente quelques limites.

Tout d'abord, nous n'avons pas considéré, par exemple, le cas où l'heuristique TreeRank repose sur la mise en oeuvre de Machines à Vecteurs Supports. Or dans cette configuration précise, la dimension VC peut s'avérer être infinie et n'est donc pas forcément la mesure de complexité la plus pertinente à prendre en compte. Il conviendrait dans ce cas, d'évaluer autrement la complexité de la collection de modèles générés par l'arbre maximal d'ordonnancement, au moyen par exemple des *moyennes conditionnelles de Rademacher* citées précédemment. Soulignons d'ailleurs que, pour cette mesure de complexité, des pénalités explicites peuvent être déduites de la borne généralisée pour le supremum de U-processus établie dans [Clémençon *et al.* 2008]. Notons de plus que l'utilisation de ces quantités présenterait l'avantage supplémentaire de conduire à des pénalités plus *fines*, calculées en fonction des observations (voir par exemple [Arlot 2007]). En effet, on peut remarquer que les deux pénalités proposées dans ce chapitre sont indépendantes de la distribution des observations. Ces pénalités pouvant être utilisées pour n'importe quelle loi de distribution, elles sont forcément adaptées au *pire des cas* et donc très pessimistes, ce qui les rend conservatives, au sens où elles ont tendance à sélectionner des modèles de petite taille. Pour aller encore plus loin, il pourrait aussi être intéressant de considérer des pénalités ré-échantillonnées, comme par exemple les pénalités *bootstrap* ou *V-fold*, citées dans la première partie de ce chapitre.

Une autre limite importante, commune à toutes les méthodes de pénalisation, découle du fait que les pénalités sont définies à une constante près seulement, qui dans notre cas apparaît notamment dans l'inégalité oracle de la Proposition 17. Or le problème de l'estimation de cette constante rend difficile la mise en pratique de ces procédures de sélection de modèle. Notons cependant que divers travaux sont menés dans le but de rendre ces approches plus *accessibles*. On peut notamment citer les récentes contributions [Birgé & Massart 2006] et [Massart 2007b], dans lesquelles les auteurs proposent une heuristique permettant de *calibrer* la valeur de la constante dans le contexte spécifique de la régression sur un design fixe avec un bruit homoscédastique. Afin d'exposer le principe de cette approche, considérons la décomposition suivante de la pénalité idéale :

$$\begin{aligned}
\mathrm{pen}_{id}(s_\omega) &= \widehat{\mathrm{ASC}}(s_\omega) - \mathrm{ASC}(\mathrm{s}_\omega), \\
&\approx \mathrm{pen}_1(s_\omega) + \mathrm{pen}_2(s_\omega), \\
&= (\mathrm{ASC}(\mathrm{s}_\omega) - \mathrm{ASC}(\mathrm{s}^*)) + (\widehat{\mathrm{ASC}}(\mathrm{s}^*) - \widehat{\mathrm{ASC}}(\mathrm{s}_\omega)),
\end{aligned}$$

où s^* est la fonction de score optimale, solution du problème (3.3). Dans [Birgé & Massart 2006], en se fondant sur l'*heuristique de pente*, les auteurs proposent une pénalité optimale de la forme

$$\mathrm{pen}_{opt}(s_\omega) = 2 \cdot \mathrm{pen}_{min}(s_\omega),$$

où la pénalité *minimale* $\text{pen}_{min}(s_\omega) = \text{pen}_2(s_\omega)$ peut être estimée à partir des observations. Partant de ce constat, ils proposent finalement d'estimer la constante multiplicative C en trois étapes, comme suit :

(i) calculer $s_{\omega_C^*} \in \arg\max_{\omega\in\Omega} \widehat{\text{ASC}}(s_\omega) + 2C \cdot \text{pen}_{min}(s_\omega)$ pour tout $C > 0$,

(ii) identifier $C_{min} = \widehat{C}$ telle que la complexité $\widetilde{\mathcal{T}}(\omega_C^*)$ de la règle $s_{\omega_C^*}$ est *grande* si $C < C_{min}$ et devient *raisonnable* quand $C > C_{min}$,

(iii) sélectionner le modèle $s_{\omega^*} = s_{\omega^*_{C_{min}}}$.

Dans [Arlot 2007], cette heuristique de *calibration* est étendue au problème de la régression sur design fixe avec un bruit hétéroscédastique. Cependant, la validité de cette approche n'a pas encore pu être établie dans le contexte de la classification binaire et donc a fortiori dans le contexte de l'ordonnancement binaire.

3.5 Conclusion et perspectives

Dans ce chapitre, nous avons considéré le problème de la sélection de la profondeur $D \geq 1$ optimale d'un arbre d'ordonnancement, garantissant la bonne généralisation de sa règle de prédiction associée. Pour ce faire, nous avons introduit une procédure en deux étapes consistant tout d'abord à apprendre un arbre aussi *grand* que possible, puis à élaguer ses branches de sorte à optimiser l'ASC empirique pénalisée par un terme tenant compte de la complexité des règles de prédiction générées. Nous avons notamment proposé deux méthodes de pénalisation différentes.

La première est une méthode dite de *régularisation*, au sens où l'ASC est pénalisée linéairement par la complexité du modèle. Plus précisément, nous avons transposé à la problématique d'ordonnancement, la procédure d'*élagage*, introduite dans [Breiman *et al.* 1984] pour les arbres de classifications de type CART, dans laquelle la complexité du modèle est évaluée par le nombre de feuilles terminales de l'arbre. Si nous n'avons pas pu établir de résultats théoriques sur la consistance de cette heuristique, nous avons toutefois pu vérifier ses performances d'un point de vue empirique. Cependant, la forme linéaire de la pénalité n'est pas forcément la plus pertinente pour sélectionner des modèles structurés de façon hiérarchique, comme ceux obtenus via l'heuristique TreeRank.

Nous avons donc considéré une deuxième approche reposant sur des pénalités non-linéaires, fonctions de la complexité du modèle. Nous avons ainsi calculé, pour deux configurations spécifiques de l'algorithme TreeRank, deux pénalités conduisant à des règles de score asymptotiquement optimales et pour lesquelles nous avons établi une inégalité oracle. Ces résultats ont été obtenus en bornant la déviation du critère ASC par rapport à sa contrepartie empirique et en contrôlant la complexité de la collection de règles de prédiction, engendrée par l'arbre maximal, au moyen de la dimension VC. Malheureusement, ces pénalités n'ont pu être définies qu'à une constante près, ce qui rend leur mise en oeuvre délicate. Une solution pourrait consister à calibrer ces constantes en se fondant sur l'*heuristique de pente* présentée dans [Birgé & Massart 2006]. Toutefois, nous ne disposons pas encore, à ce jour, de résultats théoriques justifiant la validité de cette approche dans le contexte de

l'ordonnancement binaire.

Naturellement, ces résultats pourraient être élargis, en proposant par exemple des pénalités pour d'autres configurations de l'algorithme TreeRank. Notons qu'il serait alors fort probable que la collection de modèles, engendrée par l'arbre maximal, soit bien plus importante que dans les deux configurations considérées dans ce chapitre. Dans ce cas, il pourrait s'avérer nécessaire de considérer une nouvelle mesure pour évaluer la complexité de cette collection. Par ailleurs, la considération de pénalités ré-échantillonnées permettrait de calculer des pénalités, en fonction des observations, moins pessimistes que celles obtenues en bornant la déviation de l'ASC.

Enfin, une dernière approche possible consisterait à construire une version *ré-échantillonnée* d'une règle de score, en mettant en oeuvre une procédure de *boosting*, qui est parfois vue comme une alternative intéressante aux méthodes de sélection de modèle citées dans ce chapitre (voir par exemple [Nemirovski 2000] et [Lecué 2007]). Cette approche permettrait par ailleurs de réduire l'*instabilité* de la procédure d'apprentissage, inhérente à la structure hiérarchique de l'algorithme TreeRank, problème que nous abordons dans le chapitre suivant. Cependant, l'application d'une telle procédure à des fonctions de score constantes par morceaux prenant plus de 2 valeurs distinctes est un problème difficile. Se pose notamment la question du calcul de la *pondération* des différentes règles de score pour la définition d'une règle de prédiction *moyennée*. Aussi, nous considérons, dans le Chapitre 4, d'autres méthodes, reposant sur le ré-échantillonnage des observations et/ou la *randomisation* des prédicteurs.

Chapitre 4

Ré-échantillonnage, *randomisation* et agrégation

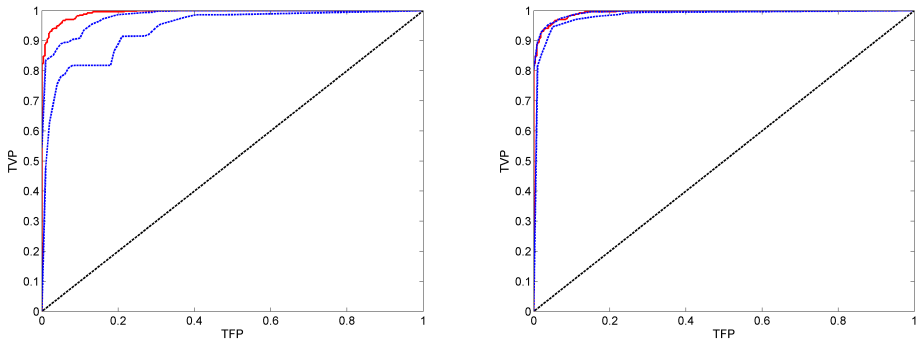
Si l'heuristique TreeRank présente les nombreux avantages d'une méthode de modélisation par arbres de décision, elle hérite aussi de ses points faibles, résidant principalement dans le manque de *régularité* de la règle de prédiction estimée, constante par morceaux par construction, et surtout dans l'*instabilité* de la procédure d'apprentissage. Ainsi, de légères modifications dans les données d'apprentissage peuvent conduire à des règles d'ordonnancement très différentes et de performances variables.

Afin d'illustrer ce phénomène, nous présentons l'expérience suivante, menée sur les données de l'exemple *GaussCroix2d*. Une collection de $N = 30$ échantillons *i.i.d.* a été générée à partir du mélange de lois explicité précédemment dans le Chapitre 1. Les versions $\mathrm{TRK}_{\mathrm{CART}}$ et $\mathrm{TRK}_{\mathrm{SVM}}$ de la méthode de scoring TreeRank (cf Chapitre 2) ont été mises en oeuvre sur chacun d'entre eux, produisant ainsi une collection de 30 règles de score constantes par morceaux. Pour pouvoir visualiser l'*instabilité* de ces deux procédures d'apprentissage, nous avons généré un échantillon de validation sur lequel nous avons calculé les courbes $\widehat{\mathrm{COR}}_v$ associées aux 30 règles de prédiction estimées. Les enveloppes (en norme $\mathcal{L}_\infty$) de ces courbes sont représentées en pointillés bleus sur la Figure 4.1, la courbe optimale COR^* étant tracée en rouge.

L'aire de l'enveloppe des courbes COR obtenues permet de visualiser la variabilité de la performance d'une règle de score produite par l'algorithme TreeRank. On remarque toutefois que celle-ci est beaucoup moins importante lorsque la procédure LeafRank repose sur la mise en oeuvre de SVM. Cependant, si la flexibilité de cette approche permet de réduire l'instabilité de la procédure d'apprentissage, elle ne peut le faire que dans une certaine mesure. En effet, ce phénomène d'instabilité, étudié dans [Breiman 1996c], est intrinsèquement lié à la structure hiérarchique du processus de partitionnement et touche ainsi toutes les méthodes reposant sur ce principe, comme par exemple l'algorithme de classification CART introduit dans [Breiman *et al.* 1984].

Diverses procédures ont été proposées pour *corriger* ce problème, basées sur le *ré-échantillonnage* des observations de l'échantillon d'apprentissage. Dans le contexte de la classification, celles-ci permettent de définir une règle de prédiction plus *robuste* à la présence de bruit dans les données, mais aussi plus *lisse* et plus *performante*, en *agrégeant*[1]

1. selon un processus à définir



a. Heuristique $\mathrm{TRK}_{\mathrm{CART}}$: courbe COR^* (en rouge) et $\widehat{\mathrm{COR}}_v$ (en pointillés bleus).

b. Heuristique $\mathrm{TRK}_{\mathrm{SVM}}$: courbe COR^* (en rouge) et $\widehat{\mathrm{COR}}_v$ (en pointillés bleus).

FIGURE 4.1 – Exemple *GaussCroix2d* : enveloppes en norme $\mathcal{L}_\infty$ des courbes COR obtenues sur $N = 30$ échantillons *i.i.d.*.

une collection de classifieurs appris sur des *ré-échantillons* des données d'apprentissage $\mathcal{D}_n = \{(X_i, Y_i),\, 1 \leq i \leq n\}$, où pour tout $i \in \{1, \cdots, n\}$, $(X_i, Y_i) \in \mathcal{X} \times \{-1, +1\}$.

Parmi ces méthodes on peut distinguer deux grandes approches, reposant sur deux schémas de *ré-échantillonnage* différents. La première consiste à agréger, selon un processus de *vote à la majorité*, une *forêt* d'arbres de décision construits sur des réplications *bootstrap* de l'échantillon d'apprentissage $\mathcal{D}_n$. La seconde repose quant à elle, sur le *ré-échantillonnage adaptatif* des observations de $\mathcal{D}_n$: à chaque itération, un nouveau classifieur est produit et les observations de $\mathcal{D}_n$ sont pondérées selon les performances de cette règle de prédiction ; finalement, la règle de classification *agrégée* est définie, comme précédemment, selon un processus de *vote à la majorité*, éventuellement pondéré.

Nous décrivons plus en détail ces deux grandes approches dans la première partie de ce chapitre, en nous focalisant sur les quatre procédures les plus utilisées en pratique : la procédure de *bagging*, introduite dans [Breiman 1996b], et les *forêts aléatoires*, proposées dans [Breiman 2001], pour la première approche et la procédure de *boosting*, proposée dans [Freund & Schapire 1999], et les classifieurs *arcing*, introduits dans [Breiman 1996a], pour la seconde. Bien que cette dernière ait pu être mise en oeuvre dans le contexte de l'ordonnancement pour agréger des règles de score binaires (voir [Freund *et al.* 2003]), son application devient moins évidente dès lors que les règles de prédictions prennent plus de deux valeurs. Aussi dans ce chapitre, nous proposons plutôt d'estimer une règle de score *moyenne* en agrégeant une *forêt* d'arbres d'ordonnancement au sens des méthodes proposées dans [Breiman 1996b] et [Breiman 2001]. Cependant, s'il est facile d'agréger des classifieurs, l'agrégation de relations d'ordre est une tâche beaucoup moins évidente. En effet, nous allons voir dans la deuxième partie de ce chapitre, qu'un pré-ordre défini selon une procédure de *vote à la majorité* ne permet pas d'établir un *bon consensus*. Nous proposons donc de considérer le problème de l'agrégation de pré-ordres du point de vue de l'*approche métrique*, dans laquelle établir un *consensus* revient à calculer un pré-ordre *médian*[2]. Enfin, après avoir montré la consistance des règles de prédiction agrégées selon ce procédé, nous proposons d'adapter la procédure de *bagging* et les *forêts aléatoires* au contexte de

2. au sens d'une pseudo-métrique entre pré-ordres à définir

l'ordonnancement binaire. Nous présentons alors deux versions *ré-échantillonnées* de l'heuristique TreeRank, dont nous illustrons les performances, en termes de *prédiction* et de *stabilité*, au moyen d'une étude empirique sur des jeux de données simulées.

4.1 Procédures de ré-échantillonnage

Comme nous venons de le souligner, une des principales limites des méthodes de classification par *arbres de décision*, dont la méthode CART ([Breiman *et al.* 1984]) fait partie, réside dans l'*instabilité* de la règle de prédiction, très sensible à la présence de bruit dans les données d'apprentissage. Aussi, diverses procédures ont été proposées afin de définir des règles de prédiction plus *robustes*. D'une manière générale, ces méthodes consistent à *apprendre* une collection de classifieurs à partir de plusieurs *ré-échantillons* des données d'apprentissage $\mathcal{D}_n$ et à les agréger, selon un processus de *vote à la majorité*. Nous décrivons ci-dessous les deux principales approches, reposant sur deux schémas de *ré-échantillonnage* différents.

4.1.1 Forêts d'arbres de classification

Dans cette partie nous présentons deux procédures de ré-échantillonnage : le *bagging* (contraction de *bootstrap aggregating*) et les *forêts aléatoires*, respectivement introduites dans [Breiman 1996b] et [Breiman 2001]. Toutes deux reposent sur la mise en oeuvre de l'heuristique *bootstrap* ([Efron 1979]), qui permet de construire des *réplications bootstrap* $\mathcal{D}_n^{(1)}, \cdots, \mathcal{D}_n^{(B)}$, $B \geq 1$, d'un échantillon $\mathcal{D}_n$, constituées de n observations tirées aléatoirement avec remise dans cet échantillon (cf Figure 3.3 du Chapitre 3).

En s'appuyant sur ce principe, l'auteur des deux contributions pré-citées introduit la notion de *forêt* d'arbres de classification, pour désigner une collection d'arbres de classification $\mathcal{T}^{\mathbf{B}} = \{\mathcal{T}^{(1)}, \cdots, \mathcal{T}^{(B)}\}$, $B \geq 1$, appris sur B réplications bootstrap de l'échantillon d'apprentissage $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$, où $(X_i, Y_i) \in \mathcal{X} \times \{-1, +1\}$ pour tout $i \in \{1, \cdots, n\}$. Dans la continuité des travaux présentés dans [Kwok & Carter 1990] (voir aussi [Heath *et al.* 1993] et [Dietterich & Bakiri 1991]), il propose alors une première procédure, appelée *bagging* ([Breiman 1996b]), qui consiste simplement à agréger les arbres d'une forêt $\mathcal{T}^{\mathbf{B}}$ selon un processus de *vote à la majorité* : pour chaque observation X'_j d'un échantillon $\mathbf{X^m} = \{X'_1, \cdots, X'_m\} \subset \mathcal{X}$ indépendant de $\mathcal{D}_n$, le classifieur *agrégé* prédit la classe *majoritaire* définie par

$$\forall\, j \in \{1, \cdots, m\},\ \widehat{Y'}_j = \underset{k \in \{-1,+1\}}{\arg\max}\ \#\{\mathcal{I} \subset \{1, \cdots, B\} \,:\, \forall\, b \in \mathcal{I},\ \widehat{Y'}_j^{(b)} = k\},$$

où $\widehat{Y'}_j^{(b)}$ est la prédiction de l'arbre $\mathcal{T}^{(b)} \in \mathcal{T}^{\mathbf{B}}$ pour l'observation $X'_j \in \mathbf{X^m}$.

L'efficacité de cette approche réside entièrement dans l'instabilité des classifieurs de la forêt $\mathcal{T}^{\mathbf{B}}$. En effet, il semble évident que l'agrégation de règles de prédictions très *similaires* conduit à définir un classifieur *proche* de celles-ci tandis qu'au contraire, agréger des règles très différentes permet de *moyenner* leurs prédictions et ainsi de réduire l'instabilité de la procédure d'apprentissage. Soulignons que dans le même temps, la règle de prédiction agrégée est plus *lisse*, quoique toujours constante par morceaux, et de plus, il a été

montré dans [Breiman 1996b] que ses performances supplantent généralement celles des classifieurs de la forêt pris individuellement. Naturellement, pour que cette procédure soit efficace il faut agréger un nombre *suffisamment* important d'arbres de classification. Dans [Breiman 1996b], l'auteur évalue empiriquement l'impact de la taille de la forêt sur les performances du classifieur agrégé sur un exemple pour lequel il montre que l'agrégation d'une douzaine d'arbres permet d'optimiser les performances en termes de prédictions, la prise en compte d'une forêt plus grande ne faisant qu'accroître le coût de calcul sans apporter de gain significatif. Toutefois, en pratique, on préfère agréger un nombre plus important de classifieurs, de $B = 50$ à $B = 1000$ selon les données considérées, d'autant plus que l'algorithme de classification CART présente un coût raisonnable, en termes de temps de calcul. Notons par ailleurs que l'auteur suggère d'agréger des arbres de taille relativement petite et fixée a priori, sans avoir recours à la procédure d'élagage proposée dans [Breiman *et al.* 1984].

Les *forêts aléatoires*, introduites dans [Breiman 2001] et inspirées des travaux présentés dans [Dietterich 1998] et [Amit & Geman 1997], peuvent être vues comme une variante de la procédure de *bagging*. Cette méthode repose en effet sur l'association de cette procédure de ré-échantillonnage et d'une étape de *randomisation* des prédicteurs : à chaque itération de la procédure d'apprentissage mise en oeuvre pour construire la forêt $\mathcal{T}^{\mathbf{B}}$, un noeud d'un arbre de classification est scindé en deux à partir d'un sous-ensemble des variables du problème, sélectionnées de manière aléatoire. Diverses études empiriques ont montré que l'introduction de cette étape supplémentaire de *randomisation* permet d'améliorer les performances du classifieur agrégé (voir par exemple [Dietterich 1998] et [Breiman 2001]). Cependant, en procédant de la sorte, cette approche induit une variabilité supplémentaire qui vient s'ajouter à l'instabilité de la procédure d'apprentissage. Il convient donc en pratique de construire des forêts plus grandes que dans la procédure de *bagging*. Notons toutefois que ceci n'implique pas nécessairement une augmentation significative du temps de calcul puisque la mise en oeuvre de la procédure de *randomisation* permet de réduire le coût de l'estimation de chaque classifieur.

Une autre question se pose lors de la mise en pratique des *forêts aléatoires* : celle du choix du nombre de variables à sélectionner pour scinder les noeuds d'un arbre de classification. Une étude empirique proposée dans [Breiman 2001] montre que les performances de la procédure de *ré-échantillonnage* ne dépendent que faiblement de ce paramètre et que l'on peut se contenter d'un nombre *relativement faible* (dans la limite du raisonnable toutefois) de prédicteurs.

De nombreuses études empiriques ont montré la supériorité de cette approche par rapport à la procédure de *bagging*, notamment du point de vue de sa résistance à la présence de bruit dans les données et de valeurs aberrantes (voir par exemple [Svetnik *et al.* 2003], [Diaz-Uriarte & de Andrés 2006] et [Dietterich 1998]). Notons par ailleurs que la consistance des règles de prédictions caractérisées par des *forêts aléatoires* a été étudiée dans [Biau *et al.* 2008].

Remarque 12 *(Estimation « out-of-bag »)*
Quelle que soit la procédure de ré-échantillonnage utilisée, la mise en oeuvre de l'heuristique bootstrap apporte une plus value intéressante en permettant d'estimer des versions généralisées de grandeurs d'intérêt, comme le risque du classifieur agrégé par exemple, à partir d'observations non utilisées pour l'apprentissage des arbres de la forêt : on parle alors d'un estimateur « out-of-bag ». Cette approche, introduite dans [Breiman 1996d],

repose sur le fait que les réplications $(\mathcal{D}_n^{(b)})_{1\leq b\leq B}$ *sont tirées avec remise et qu'il existe donc, pour tout* $b \in \{1, \cdots, B\}$*, une partie des observations de* $\mathcal{D}_n$ *qui ne sont pas utilisées pour l'apprentissage du modèle* $\mathcal{T}^{(b)}$ *et qui peuvent donc servir d'échantillon de validation pour le classifieur associé. On peut ainsi définir un estimateur sans biais du risque de la règle de prédiction finalement obtenue, sans pour autant sacrifier une partie des données d'apprentissage.*

4.1.2 Une procédure de *ré-échantillonnage adaptatif*

La deuxième approche que nous considérons ici, repose sur le *ré-échantillonnage adaptatif* des observations de $\mathcal{D}_n$, selon le principe du *boosting* introduit dans la contribution [Freund & Schapire 1999], qui consiste à pondérer les observations de l'échantillon d'apprentissage en fonction des performances de la règle de prédiction estimée. Cette méthode procède de manière itérative : un nouveau modèle est estimé à chaque itération et le *ré-échantillonnage* des données d'apprentissage est réalisé en fonction de ses performances, de telle sorte que la pondération des observations mal classées augmente tandis que celle des observations bien classées diminue. De la même façon que précédemment, la règle de prédiction agrégée est définie selon un processus de *vote à la majorité*.

Il existe différentes méthodes reposant sur ce même principe général, mais qui diffèrent sur quelques points. La plus utilisée en pratique est la procédure AdaBoost, proposée dans [Freund & Schapire 1999], dans laquelle la règle de prédiction est définie comme la moyenne pondérée des classifieurs estimés à chaque itération. On peut citer aussi les classifieurs *Arcing*, introduits dans [Breiman 1996a], qui reposent sur une pondération légèrement différente et qui sont agrégés selon un processus classique de *vote à la majorité*. Les études empiriques présentées dans [Maclin & Opitz 1997], [Bauer & Kohavi 1998] et [Dietterich 1998] par exemple, montrent que la méthode AdaBoost est nettement supérieure en comparaison avec la procédure de *bagging*, mais que par contre, les *forêts aléatoires* semblent être un sérieux compétiteur.

Il est intéressant de souligner que le principe du *boosting* a d'ores et déjà été étendu au cadre de l'ordonnancement binaire. Ainsi, la méthode RankBoost, proposée dans [Freund *et al.* 2003], repose sur l'agrégation de fonctions de score binaires, optimales au sens de l'ASC, apprises sur des *ré-échantillons* des données d'apprentissage obtenus via la procédure de *boosting* décrite précédemment. Cependant, l'adaptation de cette approche pour l'agrégation de fonctions de score prenant plus de deux valeurs distinctes est loin d'être évidente et pose notamment le problème de la définition d'une *pondération* pertinente pour les observations et pour les règles de score à agréger.

Dans le cadre de cette thèse, nous nous sommes surtout intéressés au problème de la construction de forêts (éventuellement aléatoires) d'arbres d'ordonnancement. Cependant, comme nous allons le voir dans la partie suivante, l'agrégation de relations d'ordres est un problème complexe, auquel les procédures de *vote à la majorité* n'apportent pas de solution satisfaisante.

4.2 Agrégation de pré-ordres : une approche *métrique*

Agréger des pré-ordres dans le but d'établir un consensus est une tâche difficile et la littérature consacrée à ce sujet est particulièrement vaste. Les toutes premières contributions remontent à la deuxième moitié du $XVII^{\text{ème}}$ siècle avec le développement de la *Théorie du Choix* dans le domaine des sciences sociales ([Fishburn 1973]). A cette époque, avec l'avènement des démocraties et la mise en place des systèmes électoraux, de nombreuses procédures de *vote à la majorité* ont été proposées, afin de définir un consensus à partir des *préférences* exprimées par une population sur une collection d'*alternatives*. Certaines reposent sur des séries de duels -de comparaisons par paires-, comme les procédures proposées dans [Condorcet 1785], d'autres encore, comme la *règle de Borda* par exemple ([Borda 1781]), associent un score à chaque alternative selon son rang dans les préférences que l'on cherche à agréger, on parle alors de méthodes *positionnelles*.

Cependant, dans [Condorcet 1785], l'auteur met alors en lumière un défaut majeur de ces procédures de votes : la présence de *cycles* dans le consensus, dès que le nombre d'alternatives est supérieur à deux. Ainsi, lors d'une élection portant sur au moins trois alternatives (A, B et C), la règle de préférence *majoritaire* n'est pas nécessairement *transitive*, autrement dit, il est tout à fait possible que A soit majoritairement préféré à B, que B soit majoritairement préféré à C mais que dans le même temps C soit majoritairement préféré à A et qu'il n'y ait donc pas de *vainqueur*. Ce constat amène l'auteur à formuler ce que l'on appelle le *paradoxe de Condorcet*, selon lequel il n'existe pas de règle de décision collective *simple* cohérente avec le choix d'un individu rationnel, qui sera confirmé mathématiquement dans [Arrow 1951] par le *Théorème d'impossibilité d'Arrow*, aussi appelé *paradoxe d'Arrow*.

Toutefois, dans [Condorcet 1785], l'auteur précise une condition à satisfaire pour lever son paradoxe. Il énonce ainsi le désormais célèbre *critère de Condorcet*, stipulant que *si une alternative est classée première par une majorité absolue de votants, alors elle doit l'être aussi par le consensus*. Cette contribution majeure fait encore référence aujourd'hui et une attention particulière est apportée à la définition de pré-ordres consensus satisfaisant le critère de Condorcet, ou sa variante, le *critère étendu de Condorcet* introduite dans [Truchon 1998]. Il a notamment été montré que les procédures de vote à la majorité ne satisfont pas ce critère (voir par exemple [Dwork *et al.* 2001] et [Young 1974] pour le cas particulier de la règle de Borda).

Depuis, de nombreux travaux ont été menés, que ce soit pour généraliser les concepts mathématiques introduits dans le contexte de la *Théorie du Choix* (voir notamment la référence [Barthélémy & Montjardet 1981]) ou pour définir des consensus pertinents entre pré-ordres. Une contribution majeure dans ce domaine est introduite dans [Kemeny 1959], où l'auteur propose de définir le consensus comme le *minimiseur du désaccord* entre les préférences. Cette approche satisfait notamment le critère de Condorcet et permet ainsi d'éviter la présence de cycles mais elle présente aussi un inconvénient majeur. En effet, il a été montré que le calcul du *consensus optimal de Kemeny* nécessite de résoudre un problème $NP-$complet (voir par exemple [Cohen *et al.* 1999], [Dwork *et al.* 2001], ainsi que les contributions [Hudry 2004], [Hudry 2008] ou encore [Wakabayashi 1998]).

Plus récemment, la problématique de l'agrégation de pré-ordres -ou de préférences- a connu un regain d'intérêt notable, en particulier auprès de la communauté de l'apprentissage

automatique, suite à l'émergence de nouvelles applications dans le contexte du développement du réseau internet, comme par exemple, le design de *méta-moteurs* de recherche, le *filtrage collaboratif* ou encore la mise au point de filtres *anti-spam* (voir notamment [Pennock *et al.* 2000], [Dwork *et al.* 2001], [Fagin *et al.* 2003] et [Ilyas *et al.* 2002]). Aussi, de nombreux travaux ont été menés afin de développer des procédures efficaces pour le calcul de pré-ordres consensus *optimaux*, au sens de Kemeny notamment (voir par exemple les contributions [Betzler *et al.* 2008], [Mandhani & Meila 2009], [Meila *et al.* 2007], ou encore [Dwork *et al.* 2001], [Fagin *et al.* 2003] et [Conitzer 2006]).

Dans ce chapitre, nous considérons le problème de l'agrégation de pré-ordres selon une approche métrique. De ce point de vue, établir un consensus consiste à calculer un pré-ordre *médian*, au sens d'une mesure de la distance -ou au contraire de l'*adéquation*- entre relations d'ordres. Aussi, nous explicitons cette notion de pré-ordre *médian* dans la Partie 4.2.1 suivante, avant de présenter trois métriques, permettant d'évaluer la *similarité* entre deux pré-ordres définis sur un ensemble fini. Puis, nous revenons à la problématique initiale de l'agrégation de pré-ordres induits par des fonctions de score constantes par morceaux. Nous soulignons notamment que de telles fonctions permettent d'induire deux types de pré-ordres de nature différente et que l'on peut donc considérer deux stratégies pour l'implémentation pratique d'un consensus. Nous en privilégierons une des deux, pour laquelle nous présentons, dans la Partie 4.2.4, des résultats théoriques relatifs à la consistance, au sens de l'ASC, du consensus obtenu. Notons que dans cette partie, nous confondrons abusivement la notion de pré-ordre et de règle de score.

4.2.1 Notion de pré-ordre médian

Considérons une collection $\Pi_{\mathcal{Z}}^B$ de B de pré-ordres $\Pi_{\mathcal{Z}} = \{\preceq_1, \cdots, \preceq_B\}$, $B \geq 1$, définis sur un ensemble $\mathcal{Z}$. Par analogie avec le contexte de la *Théorie du Choix*, on parlera du *profil* $\Pi_{\mathcal{Z}}^B$. L'objectif est de trouver un pré-ordre « *central* », qui établisse le *consensus* entre les pré-ordres du profil $\Pi_{\mathcal{Z}}^B$. En présence de données scalaires, les notions de *moyenne* ou de *médiane* pourvoient ce genre de résultat. Cependant dans le contexte de l'ordonnancement, la notion de *consensus* est plus *floue* et peut être interprétée de différentes façons. Ici, nous considérons le problème de l'agrégation de pré-ordres du point de vue de l'approche *métrique*, en nous basant sur la procédure dite *médiane*, introduite dans [Barthélémy & Montjardet 1981], dans laquelle le *consensus* est défini par un pré-ordre *médian*, au sens d'une mesure de la distance entre pré-ordres à définir.

Considérons, dans un premier temps, un profil de pré-ordres définis sur un ensemble $\mathcal{Z}$ de cardinal $N < \infty$. On peut noter que, dans ce cas spécifique, on a

$$\#\Pi_{\mathcal{Z}} = \sum_{k=1}^{N} (-1)^k \sum_{m=1}^{k} (-1)^m \left(\begin{array}{c} k \\ m \end{array} \right) m^N$$

pré-ordres possibles sur $\mathcal{Z}$, où

$$(-1)^k \sum_{m=0}^{k} (-1)^m \left(\begin{array}{c} k \\ m \end{array} \right) m^N$$

est le nombre de surjections de $\{1, \ldots, N\}$ dans $\{1, \ldots, k\}$, pour $1 \leq k \leq N$. Dans ce contexte, on peut définir la notion de pré-ordre *consensus* comme suit.

Définition 12 *(Pré-ordre médian)*
Soit $\Pi_{\mathcal{Z}}$ l'ensemble des pré-ordres définis sur un ensemble $\mathcal{Z}$ de cardinal fini, et un profil $\Pi_{\mathcal{Z}}^B = \{\preceq_1, \cdots, \preceq_B\}$, $B \geq 1$, tel que $\Pi_{\mathcal{Z}}^B \subset \Pi_{\mathcal{Z}}$. On note δ une mesure de la distance entre pré-ordres. Le pré-ordre $\preceq^* \in \Pi_{\mathcal{Z}}$ établissant le consensus parmi le profil $\Pi_{\mathcal{Z}}^B$ est un pré-ordre médian, défini par

$$\preceq^* \in \arg\min_{\preceq \in \Pi_{\mathcal{Z}}} \sum_{b=1}^{B} \delta(\preceq, \preceq_b). \tag{4.1}$$

Remarque 13 *(Unicité)*
Notons qu'en règle générale, il n'y a pas un unique pré-ordre médian solution du problème (4.1). On peut en effet vérifier que tout pré-ordre défini sur $Z = \{1,\ 2\}$ est une médiane au sens de la métrique du τ de Kendall dénombrant le taux de paires discordantes, définie ci-après dans la Partie 4.2.2.

Considérons maintenant un profil de pré-ordres, $\Pi_{\mathcal{X}}^B$, induits sur un espace $\mathcal{X}$ de cardinal infini par une collection de fonctions de score $(s^{(b)})_{1 \leq b \leq B}$, $B \leq 1$, définies sur $\mathcal{X}$ et munissons-nous d'une pseudo-métrique $\tilde{\delta}$ sur l'ensemble de ces pré-ordres. Il est important de noter que dans ce cas, l'existence d'une règle de score *médiane* $\tilde{s} \in \mathcal{S}$, telle que

$$\sum_{b=1}^{B} \delta(\tilde{s}, s^{(b)}) = \inf_{s \in \mathcal{S}} \sum_{b=1}^{B} \delta(s, s^{(b)}), \tag{4.2}$$

n'est en aucun cas garantie dans le cas général.

Par contre, ce problème ne se pose pas si l'on se restreint à des fonctions de score constantes par morceaux. En effet, on peut alors considérer qu'une règle de score définit un pré-ordre, que l'on notera $\preceq_{S_N}$, sur les cellules de la partition $\mathcal{P}$ finie de $\mathcal{X}$ induite par S_N et se ramener au cas de pré-ordres définis sur un ensemble de cardinal fini. Dans ce cas précis, la distance minimale définie par (4.2) est donc effectivement atteinte par une fonction de score *médiane*, constante sur les cellules de la partition $\mathcal{P}$.

4.2.2 Mesures d'adéquation entre deux pré-ordres

On peut considérer diverses métriques pour évaluer la *similarité* ou la *dissimilarité* entre pré-ordres définis sur un même ensemble fini $\mathcal{Z}$. Beaucoup ont été proposées dans la littérature (voir par exemple [Diaconis 1988], [Marden 1995] et [Critchlow 1985]) et l'on pourrait naturellement en définir de nouvelles, en étendant par exemple les métriques définies sur les groupes symétriques sur des ensembles finis (cf [Howie 2000] et [Deza & Deza 2009]).

Dans cette partie, nous allons nous intéresser à trois mesures de similarité issues de la théorie des tests statistiques non-paramétriques (voir notamment [Fagin *et al.* 2003] et le Chapitre 4 de [Mielke & Berry 2001]). Plus précisémment, nous considérons des extensions de ces métriques classiques. En effet, dans la plupart des études théoriques dédiées au problème de l'agrégation de pré-ordres, les rangs définis sur un ensemble fini $\mathcal{Z}$ sont décrits par des permutations sur l'ensemble des indices $\{1, \ldots, \#(\mathcal{Z})\}$, sans laisser la possibilité de la présence d'ex-aequo dans l'ordonnancement des observations. Or, dans ce manuscrit, nous considérons des règles de score caractérisées par des arbres, qui sont donc

constantes par morceaux par construction et prédisent des ex-aequo sur l'espace $\mathcal{X}$. Dans la littérature, on parle parfois de règles d'ordonnancement *partielles* (voir par exemple [Bansal & Fernández-Baca 2009], [Fagin *et al.* 2003] et [Fagin *et al.* 2006]). Les mesures d'adéquation introduites ci-après ont donc été adaptées, selon l'approche proposée dans [Kendall 1945], afin de tenir compte de la présence d'ex-aequo dans les ordonnancements (voir aussi [Mielke & Berry 2001]).

Dans toute cette partie, on considère deux pré-ordres $\preceq_1$ et $\preceq_2$, définis sur un ensemble fini $\mathcal{Z} = \{z_1, \ldots, z_N\}$, $N \geq 1$. Pour tout $b \in \{1, 2\}$ et tout $z_i \in \mathcal{Z}$, on note $\mathcal{R}_{\preceq_b}(z_i)$ le rang attribué à l'observation z_i par le pré-ordre $\preceq_b$.

La mesure de similarité la plus couramment citée et étudiée dans la littérature repose sur la distance de Kendall ([Kendall 1945]), notée δ_τ, qui dénombre les *paires d'observations discordantes*, *i.e.* ordonnées différemment par les pré-ordres considérés. En effet, minimiser cette distance sur l'ensemble $\Pi_{\mathcal{Z}}$ des pré-ordres définis sur l'ensemble $\mathcal{Z}$, revient à calculer le *consensus optimal de Kemeny* (voir [Kemeny 1959]). Nous définissons ci-dessous une extension de la distance classique de Kendall (voir par exemple [Fagin *et al.* 2006]), permettant de tenir compte de la présence d'ex-aequo, où le nombre de paires de $\mathcal{Z}$ discordantes selon $\preceq_1$ et $\preceq_2$ est évalué en attribuant par convention le poids $1/2$ quand deux observations de $\mathcal{Z}$ sont classées ex-aequo par l'un des pré-ordre uniquement.

Définition 13 *(Le τ de Kendall)*
La distance de Kendall entre deux pré-ordres $\preceq_1$ et $\preceq_2$, définis sur $\mathcal{Z}$, est la U-statistique d'ordre 2 donnée par :

$$\begin{aligned}
\delta_\tau(\preceq_1, \preceq_2) &= \sum_{1 \leq i < j \leq N} U_{i,j}(\preceq_1, \preceq_2) \\
&= \sum_{1 \leq i < j \leq N} \big(\mathbb{I}\{(\mathcal{R}_{\preceq_1}(z_i) - \mathcal{R}_{\preceq_1}(z_j))(\mathcal{R}_{\preceq_2}(z_i) - \mathcal{R}_{\preceq_2}(z_j)) < 0\} \\
&+ \frac{1}{2}\mathbb{I}\{\mathcal{R}_{\preceq_1}(z_i) = \mathcal{R}_{\preceq_1}(z_j),\ \mathcal{R}_{\preceq_2}(z_i) \neq \mathcal{R}_{\preceq_2}(z_j)\} \\
&+ \frac{1}{2}\mathbb{I}\{\mathcal{R}_{\preceq_2}(z_i) = \mathcal{R}_{\preceq_2}(z_j),\ \mathcal{R}_{\preceq_1}(z_i) \neq \mathcal{R}_{\preceq_1}(z_j)\} \big). \qquad (4.3)
\end{aligned}$$

La mesure de similarité entre pré-ordres, appelée τ de Kendall, est alors définie par la normalisation suivante :

$$\widehat{\tau}(\preceq_1, \preceq_2) = 1 - \frac{4}{N(N-1)}\delta_\tau(\preceq_1, \preceq_2). \qquad (4.4)$$

La statistique $\widehat{\tau}(\preceq_1, \preceq_2)$ est à valeurs dans $[-1, +1]$, où la valeur -1 correspond au cas d'inadéquation totale entre les pré-ordres et la valeur 1 au cas d'adéquation parfaite. Par ailleurs, comme cela est souligné dans [Conitzer 2006] par exemple, le consensus de Kemeny, obtenu par la maximisation de cette mesure de similarité, peut être interprété en termes de maximum de vraisemblance. En effet, si on note $\preceq^*$ le *vrai* pré-ordre médian sur $\mathcal{Z}$ et que l'on considère une séquence de pré-ordres $\{\preceq_1, \cdots, \preceq_B\}$, $B \geq 1$, définis comme des versions *bruitées* de $\preceq^*$, alors le pré-ordre médian au sens du τ de Kendall maximise la vraisemblance de produire le consensus $\preceq^*$ (voir par exemple [Conitzer 2006]).

En ce qui concerne l'implémentation de cette mesure de similarité, on peut remarquer que dans le cas particulier de deux pré-ordres $\preceq_1$ et $\preceq_2$ ne contenant aucun ex-aequo, calculer la quantité (4.4) revient à calculer la distance « bubble sort » entre ces pré-ordres,

i.e. à évaluer le nombre de transpositions nécessaires entre paires adjacentes pour passer du pré-ordre $\preceq_1$ au pré-ordre $\preceq_2$. Ainsi, un calcul *naïf* de cette distance induirait un coût non négligeable d'ordre $O(N^2)$. Mais, dans [Bansal & Fernández-Baca 2009], les auteurs proposent une heuristique permettant de réduire ce coût et de calculer le τ de Kendall entre deux pré-ordres en $O(N \log N / \log\log N)$.

Cependant, comme cela a été démontré dans [Wakabayashi 1998], [Hudry 2004] et [Hudry 2008] par exemple, trouver un pré-ordre $\preceq$ sur $\mathcal{Z}$ maximisant la similarité, au sens du τ de Kendall, avec l'ensemble d'un profil $\Pi_{\mathcal{Z}}^B$, $B \geq 1$, est un problème $NP-$complet. Aussi, de nombreux travaux de recherche ont été menés et des procédures d'approximation efficaces ont pu être proposées, mettant en oeuvre des méta-heuristiques (probabilistes) telles que le recuit simulé, des algorithmes génétiques ou la recherche *tabu* (voir notamment [Spall 2003], [Laguna *et al.* 1999], [Charon & Hudry 1998] et [Betzler *et al.* 2008]). Toutefois, étant donné le temps de calcul requis par ces différentes approches, nous proposons de considérer deux autres mesures de similarité, plus faciles à calculer, la règle de Spearman et le coefficient de corrélation ρ de Spearman, que nous introduisons ci-dessous.

La règle de Spearman permet de mesurer l'adéquation entre deux pré-ordres $\preceq_1$ et $\preceq_2$, en calculant la distance $\mathcal{L}_1$ entre leurs prédictions sur les observations de $\mathcal{Z}$. De même que le τ de Kendall défini précédemment, cette mesure de similarité est à valeurs dans $[-1, +1]$.

Définition 14 *(La règle de Spearman)*
La règle de Spearman pour deux pré-ordres $\preceq_1$ et $\preceq_2$, définis sur l'ensemble $\mathcal{Z}$, est définie par :

$$\widehat{F}(\preceq_1, \preceq_2) = 1 - \frac{3 \cdot \sum_{i=1}^{N} |\mathcal{R}_{\preceq_1}(z_i) - \mathcal{R}_{\preceq_2}(z_i)|}{N^2 - 1}. \tag{4.5}$$

Au lieu de considérer la distance $\mathcal{L}_1$, le coefficient de corrélation ρ de Spearman est basé sur la distance quadratique entre les rangs des observations de $\mathcal{Z}$. Encore une fois, on obtient une mesure de similarité définie sur $[-1, +1]$.

Définition 15 *(Le ρ de Spearman)*
Le coefficient ρ de Spearman pour deux pré-ordres $\preceq_1$ et $\preceq_2$, définis sur l'ensemble $\mathcal{Z}$, est donné par :

$$\widehat{\rho}(\preceq_1, \preceq_2) = 1 - \frac{6 \cdot \sum_{i=1}^{N} (\mathcal{R}_{\preceq_1}(z_i) - \mathcal{R}_{\preceq_2}(z_i))^2}{N(N^2 - 1)}. \tag{4.6}$$

L'utilisation de l'une de ces deux mesures, en remplacement du τ de Kendall, nous permet de calculer des pré-ordres *médians* de manière efficace. En effet, il est facile de voir qu'un pré-ordre établissant le consensus parmi un profil $\Pi_{\mathcal{Z}}^B$, au sens de la règle et du coefficient ρ de Spearman respectivement, correspond simplement à une règle d'ordonnancement affectant à chaque observation de $\mathcal{Z}$, respectivement, la médiane et la moyenne des prédictions des pré-ordres du profil $\Pi_{\mathcal{Z}}^B$. Par ailleurs, le Théorème 13 de [Fagin *et al.* 2006] apporte une justification théorique à l'utilisation de ces mesures alternatives. En effet, dans cette contribution, les auteurs montrent que les trois mesures présentées précédemment sont *équivalentes*, au sens où , pour tout pré-ordre $\preceq_1$ et $\preceq_2$ sur $\mathcal{Z}$, on a

$$c_1 \left(1 - \widehat{\rho}(\preceq_1, \preceq_2)\right) \leq \left(1 - \widehat{F}(\preceq_1, \preceq_2)\right)^2 \leq c_2 \left(1 - \widehat{\rho}(\preceq_1, \preceq_2)\right), \tag{4.7}$$

$$c_3 \left(1 - \widehat{\tau}(\preceq_1, \preceq_2)\right) \leq 1 - \widehat{F}(\preceq_1, \preceq_2) \leq c_4 \left(1 - \widehat{\tau}(\preceq_1, \preceq_2)\right), \tag{4.8}$$

avec $c_2 = N^2/(2(N^2-1)) = Nc_1$ et $c_4 = 3N/(2(N+1)) = 2c_3$.

4.2.3 Agrégation de règles de score constantes par morceaux

Dans la suite de cette partie, nous allons nous intéresser plus particulièrement au problème de l'agrégation de pré-ordres induits par des fonctions de score constantes par morceaux. Soit $\mathcal{T}^{\mathbf{B}} = \{\mathcal{T}^{(1)}, \cdots, \mathcal{T}^{(B)}\}$, $B \geq 1$, une séquence d'arbres d'ordonnancement obtenue par le biais de l'heuristique TreeRank, représentant une collection de B fonctions de score constantes par morceaux $\mathcal{S}^{\mathbf{B}} = \{s^{(1)}, \cdots, s^{(B)}\}$ définies sur l'espace $\mathcal{X}$. Notre objectif est de définir une règle de score constante par morceaux $\tilde{s} \in \mathcal{S}_N$, $N \geq 1$, sur $\mathcal{X}$, qui établisse un consensus parmi les fonctions de $\mathcal{S}^{\mathbf{B}}$. Pour ce faire, on va pouvoir considérer deux stratégies d'agrégation différentes.

En effet, considérons une fonction de score $s \in \mathcal{S}_{2^D}$ définie sur l'espace probabiliste $\mathcal{X}$ et représentée par un arbre d'ordonnancement $\mathcal{T}_D$, de profondeur $D \geq 1$. On note $\mathcal{P}$ la partition finie et ordonnée de $\mathcal{X}$ caractérisée par l'arbre $\mathcal{T}_D$, constituée de 2^D cellules non vides $C_{D,k}$, où $k \in \{0, \cdots, 2^D - 1\}$. Comme nous l'avons déjà souligné, on peut considérer que la fonction $s \in \mathcal{S}_{2^D}$ ainsi définie permet d'induire deux types de pré-ordres de nature différente, respectivement sur l'ensemble fini $\mathcal{P} = \{C_{D,k},\, 0 \leq k \leq 2^D - 1\}$ et sur l'espace $\mathcal{X}$.

On note $\preceq_s$ le pré-ordre induit par s sur $\mathcal{P}$ et pour toute cellule $C_{D,k} \in \mathcal{P}$, $\mathcal{R}_{\preceq_s}(C_{D,k})$ désigne son rang selon $\preceq_s$. Rappelons que pour tout $(k, k') \in \{0, \cdots, 2^D - 1\}^2$, on a

$$\mathcal{R}_{\preceq_s}(C_{D,k}) < \mathcal{R}_{\preceq_s}(C_{D,k'}) \text{ si et seulement si } s(C_{D,k}) < s(C_{D,k'}),$$

où $s(C_{D,k})$ est le score attribué par la fonction s à la cellule $C_{D,k}$. On note dans ce cas $C_{D,k} \preceq_s C_{D,k'}$.

La fonction s prenant des valeurs distinctes sur chaque cellule de $\mathcal{P}$, $\preceq_s$ ne prédit pas d'ex-aequo, à la différence du pré-ordre induit sur $\mathcal{X}$, qui est noté $\preccurlyeq_s$. En effet, par construction, la fonction s est constante sur les cellules de $\mathcal{P}$ et attribue donc le même rang, $\mathcal{R}_{\preccurlyeq_s}(x)$, à toutes les observations $x \in C_{D,k}$ appartenant à une même cellule. Dans ce cas particulier, il n'est pas évident de déterminer la valeur du rang qu'il faut attribuer à ces observations. Diverses conventions ont été proposées dans la littérature, afin de calculer le rang des observations d'un ensemble en présence d'ex-aequo (cf Chapitre 13 de [van der Vaart 1998]). Ici, nous utilisons la convention *midrank* -du *rang moyen*-, selon laquelle le rang d'une observation $x \in \mathcal{X}$ est donné par :

$$\mathcal{R}_{\preccurlyeq_s}(x) = \sum_{x' \in \mathcal{X}} \mathbb{I}\{x \succ x'\} + \frac{1}{2}\left(1 + \mathbb{I}\{x \asymp x'\}\right),$$

où $x \asymp x'$ si et seulement si $s(x) = s(x')$ et $x \succ x'$ si et seulement si $s(x') < s(x)$. Finalement, pour tout $(x, x') \in \mathcal{X}^2$, on note $x \preccurlyeq_s x'$ si et seulement si $\mathcal{R}_{\preccurlyeq_s}(x) \leq \mathcal{R}_{\preccurlyeq_s}(x')$.

Notons $\Pi^B_{\mathcal{P}^*} = (\preceq_{s^{(b)}})_{1 \leq b \leq B}$ et $\Pi^B_{\mathcal{X}} = (\preccurlyeq_{s^{(b)}})_{1 \leq b \leq B}$ les profils de pré-ordres induits par les fonctions de la collection $\mathcal{S}^B$ sur une partition $\mathcal{P}^*$ de $\mathcal{X}$ et sur $\mathcal{X}$ respectivement. Afin d'alléger les notations, on posera, pour tout $b \in \{1, \cdots, B\}$, $\preceq_b = \preceq_{s^{(b)}}$ et $\preccurlyeq_b = \preccurlyeq_{s^{(b)}}$. On peut donc envisager deux démarches pour définir une règle de score médiane. Une première approche consisterait à définir un consensus parmi les pré-ordres du profil $\Pi^B_{\mathcal{P}^*}$, ce qui revient à considérer le problème de l'agrégation de pré-ordres définis sur un ensemble

fini, de la même façon que dans la Partie 4.2.1. Une seconde approche reposerait, quant à elle, sur le calcul d'un pré-ordre *médian* pour le profil $\Pi_{\mathcal{X}}^{B}$, au sens d'une pseudo-métrique $\tilde{\delta}$ définie sur l'ensemble $\Pi_{\mathcal{X}}$ des pré-ordres définis sur $\mathcal{X}$.

Comme nous allons le voir dans la Partie 4.2.3.3, ces deux stratégies sont sensiblement différentes. Cependant, afin de pouvoir discuter de leurs avantages et inconvénients respectifs, il nous faut auparavant introduire quelques concepts supplémentaires. Aussi, dans la Partie 4.2.3.1 suivante, nous explicitons la forme de la partition finie $\mathcal{P}^*$ de $\mathcal{X}$ sur laquelle nous calculerons le consensus pour le profil $\Pi_{\mathcal{P}^*}^{B}$. Puis, nous proposons, dans la Partie 4.2.3.2, des mesures permettant d'évaluer la similarité entre les pré-ordres du profil $\Pi_{\mathcal{X}}^{B}$, sans faire référence à la partition sous-jacente $\mathcal{P}^*$ de $\mathcal{X}$.

4.2.3.1 Pré-ordres induits sur les cellules d'une partition $\mathcal{P}^*$ de $\mathcal{X}$

Dans cette partie, on considère le problème de l'agrégation des pré-ordres du profil $\Pi_{\mathcal{P}^*}^{B}$, induits par la collection de fonctions de score $\mathcal{S}^{\mathbf{B}}$ et nous allons voir que ceci revient à agréger des pré-ordres définis sur un ensemble de cardinal fini. En effet, chaque fonction $s^{(b)} \in \mathcal{S}^{\mathbf{B}}$, pour tout $b \in \{1, \cdots, B\}$, définit une partition finie et ordonnée sur $\mathcal{X}$, que l'on notera $\mathcal{P}^{(b)}$. Par définition, chaque pré-ordre $\preceq_b \in \Pi_{\mathcal{P}^*}^{B}$ permet d'ordonner les cellules de la partition associée $\mathcal{P}^{(b)}$. Il en découle donc que tous les pré-ordres du profil permettent d'ordonner les cellules de la partition de $\mathcal{X}$ *commune* à toutes les fonctions de score de la collection $\mathcal{S}^{\mathbf{B}}$.

Afin de définir cette partition, nous rappelons la notion de *sous-partition* introduite dans la Définition 7 du Chapitre 1. Etant donné deux partitions $\mathcal{P}$ et $\mathcal{P}'$ de l'espace d'entrée $\mathcal{X}$, $\mathcal{P}'$ est une sous-partition de $\mathcal{P}$, notée $\mathcal{P}' \subset \mathcal{P}$, si toute cellule non vide $\mathcal{C}$ de $\mathcal{P}$ peut être obtenue par réunion des cellules de $\mathcal{P}'$.

Considérons maintenant la partition $\mathcal{P}^*$ de $\mathcal{X}$, constituée de sous-ensembles non vides $\mathcal{C} \subset \mathcal{X}$ satisfaisant les deux contraintes suivantes :

(i) $\exists\, (\mathcal{C}_1, \ldots, \mathcal{C}_B) \in \mathcal{P}^{(1)} \times \cdots \times \mathcal{P}^{(B)}$ telle que $\mathcal{C} = \bigcap_{b=1}^{B} \mathcal{C}_b$,

(ii) $\forall\, \mathcal{C}' \in \mathcal{P}^{(b)}$, $b \in \{1, \ldots, B\}$, $\mathcal{C}' \subset \mathcal{C} \Rightarrow \mathcal{C}' = \mathcal{C}$.

On notera $\mathcal{P}^* = \bigcap_{b \leq B} \mathcal{P}^{(b)}$ cette partition définie comme une sous-partition de toutes les partitions $\mathcal{P}^{(b)}$, $b \in \{1, \ldots, B\}$. En d'autres termes, il s'agit de la sous-partition la plus *fine*, au sens où, pour tout $b \in \{1, \ldots, B\}$, la sous-partition $\mathcal{P} \subset \mathcal{P}^{(b)}$ est une sous-partition de $\mathcal{P}^*$. Ainsi, $\mathcal{P}^*$ est bien *commune* à l'ensemble des fonctions de $\mathcal{S}^{\mathbf{B}}$: quel que soit $b \in \{1, \ldots, B\}$, $s^{(b)}$ induit un pré-ordre $\preceq_b$ sur les cellules de $\mathcal{P}^*$. Notons d'ailleurs que sur cette partition, les pré-ordres du profil $\Pi_{\mathcal{P}^*}^{B}$ peuvent induire des ex-aequo.

On peut remarquer que, du point de vue de l'implémentation pratique, la représentation arborescente des partitions $\mathcal{P}^{(b)}$ de $\mathcal{X}$, pour $b \in \{1, \cdots, B\}$, facilite considérablement l'obtention de $\mathcal{P}^*$. En effet, la partition la plus fine peut être obtenue selon le procédé schématisé sur la Figure 4.2 ci-dessous.

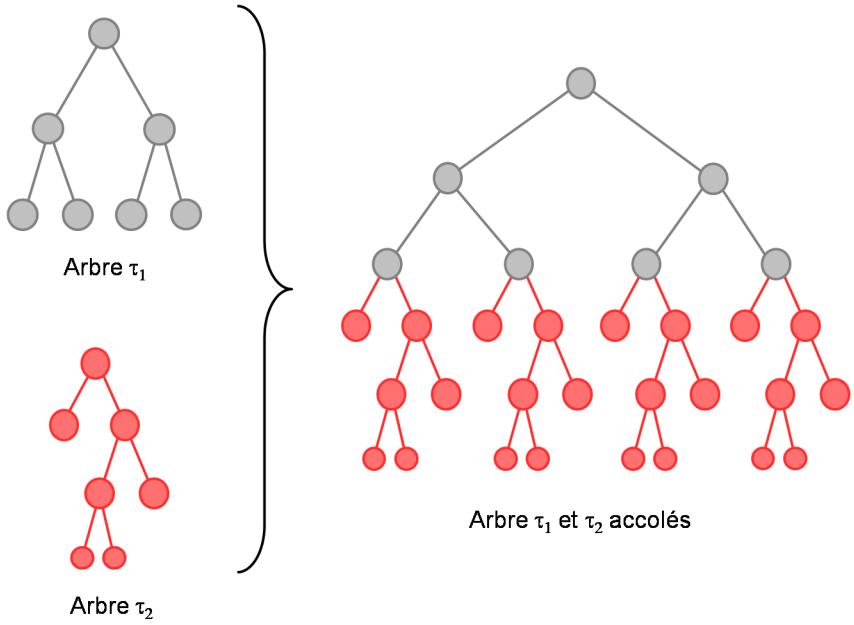


FIGURE 4.2 – Caractérisation graphique de la sous-partition la plus fine associée à une forêt constituée de deux arbres.

Soient $\mathcal{P}^{(1)} = \{\mathcal{C}_{D_1,k}\}_{0\leq k\leq 2^{D_1}-1}$ et $\mathcal{P}^{(2)} = \{\mathcal{C}_{D_2,k'}\}_{0\leq k'\leq 2^{D_2}-1}$ les deux partitions de $\mathcal{X}$ définies par les arbres $\mathcal{T}^{(1)}$ et $\mathcal{T}(2)$, de profondeurs respectives $D_1 \geq 1$ et $D_2 \geq 1$. Pour tout couple (k, k'), la collection des sous-ensembles de la forme $\mathcal{C}_{D_1,k} \cap \mathcal{C}_{D_2,k'}$ peut être obtenue en *accolant* les structures de $\mathcal{T}^{(1)}$ et de $\mathcal{T}^{(2)}$ comme suit : à chaque feuille $\mathcal{C}_{D_1,k}$ de $\mathcal{T}^{(1)}$, on accole le sous-arbre $\mathcal{T}^{(2)}$, dont la racine deviendra $\mathcal{C}_{D_1,k}$, comme cela est schématisé sur la Figure 4.2. Les feuilles terminales résultant de cette procédure forment ainsi les cellules de la sous-partition la plus fine de $\mathcal{P}^{(1)}$ et $\mathcal{P}^{(2)}$.

Pour construire la sous-partition la plus fine $\mathcal{P}^*$ associée à la collection $\mathcal{T}^{\mathbf{B}}$, il suffit d'itérer ce processus pour chaque arbre $\mathcal{T}^{(b)}$. Pour faciliter l'implémentation, l'idéal est d'initier la procédure avec l'arbre le plus complexe et d'accoler, de manière itérative, des arbres de plus en plus petits. Notons que certaines cellules de la sous-partition la plus fine, obtenue par ce procédé, peuvent être vides par construction.

4.2.3.2 Pré-ordres induits sur les observations de $\mathcal{X}$

Afin de pouvoir considérer le problème de la définition d'un consensus pour le profil $\Pi_{\mathcal{X}}^B$, indépendamment de la partition $\mathcal{P}^*$ de $\mathcal{X}$, il nous faut définir des métriques sur l'ensemble $\Pi_{\mathcal{X}}$ des pré-ordres induits sur un espace de cardinal infini, par des fonctions de score constantes par morceaux. En effet, de la même façon que dans le cas de pré-ordres définis sur un ensemble fini, on souhaite pouvoir quantifier la distance entre deux pré-ordres $\preccurlyeq_1$ et $\preccurlyeq_2$ définis sur $\mathcal{X}$, au sens d'une pseudo-métrique $\tilde{\delta}$ telle que

$$\tilde{\delta}(\preccurlyeq_1, \preccurlyeq_2) \overset{def}{=} \delta(\preceq_1, \preceq_1),$$

où δ évalue la distance entre pré-ordres définis sur une partition finie $\mathcal{P}$ de $\mathcal{X}$ (cf Partie 4.2.1). Pour ce faire, nous allons nous appuyer sur le fait que les mesures d'adéquation introduites dans la Partie 4.2.2 précédente, impliquent des quantités ayant des contreparties

probabilistes bien identifiées.

Considérons par exemple le cas particulier de la mesure du τ de Kendall (cf Définition 13). On peut introduire le τ théorique de Kendall $\widetilde{\tau}(X_1, X_2)$, associé à un couple de variables aléatoires $(X_1, X_2) \in \mathcal{X}^2$, défini par

$$\widetilde{\tau}(X_1, X_2) = 1 - 2\tilde{\delta}_\tau(X_1, X_2),$$

avec

$$\begin{aligned}\tilde{\delta}_\tau(X_1, X_2) = \mathbb{P}\{(X_1 - X_1') \cdot (X_2 - X_2') < 0\} &+ \frac{1}{2}\mathbb{P}\{X_1 = X_1', X_2 \neq X_2'\} \\ &+ \frac{1}{2}\mathbb{P}\{X_1 \neq X_1', X_2 = X_2'\}, \quad (4.9)\end{aligned}$$

où (X_1', X_2') est une copie *i.i.d.* du couple (X_1, X_2). Un moyen naturel de mesurer la similarité entre les pré-ordres $\preccurlyeq_1$ et $\preccurlyeq_2$, induits sur $\mathcal{X}$ par deux fonctions de score constantes par morceaux $s^{(1)}$ et $s^{(2)}$, consiste donc à calculer la quantité

$$\tau_X(\preccurlyeq_1, \preccurlyeq_2) = \widetilde{\tau}(s^{(1)}(X), s^{(2)}(X)),$$

évaluant la probabilité que les fonctions $s^{(1)}$ et $s^{(2)}$ ordonnent deux observations indépendantes, X et X' de $\mathcal{X}$, dans le même sens. Afin d'éviter toute confusion, la quantité τ_X sera désignée comme le τ *probabiliste* de Kendall. Dans le but de définir des notations cohérentes, on posera aussi $\tilde{\delta}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2) = \tilde{\delta}_\tau(s_1(X), s_2(X))$.

Un simple calcul, que nous ne reproduisons pas ici, nous permet d'établir le lien suivant entre la distance $\tilde{\delta}_{\tau_X}$ et la U-statistique permettant d'évaluer l'adéquation entre deux pré-ordres $\preceq_1$ et $\preceq_2$, introduite dans la Définition 13 du τ de Kendall.

Lemme 14 *Soient $s^{(1)}$ et $s^{(2)}$ deux fonctions de score constantes par morceaux, induisant les pré-ordres $\preceq_1$ et $\preceq_2$ sur une partition $\mathcal{P}$ de $\mathcal{X}$, constituée de N cellules non vides $(C_i)_{1 \leq i \leq N}$. Avec ces notations, on a*

$$\tilde{\delta}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2) = 2 \sum_{1 \leq i < j \leq N} \mu(\mathcal{C}_i)\mu(\mathcal{C}_j) \cdot U_{i,j}(\preceq_1, \preceq_2), \quad (4.10)$$

où on rappelle que μ est la loi marginale des observations de $\mathcal{X}$ et $U_{i,j}$ est définie, pour tout $1 \leq i < j \leq N$, par

$$\begin{aligned}U_{i,j} &= \mathbb{I}\{(\mathcal{R}_{\preceq_1}(C_i) - \mathcal{R}_{\preceq_1}(C_j))(\mathcal{R}_{\preceq_2}(C_i) - \mathcal{R}_{\preceq_2}(C_j)) < 0\} \\ &+ \mathbb{I}\{\mathcal{R}_{\preceq_1}(C_i) = \mathcal{R}_{\preceq_1}(C_j),\ \mathcal{R}_{\preceq_2}(C_i) \neq \mathcal{R}_{\preceq_2}(C_j)\} \\ &+ \mathbb{I}\{\mathcal{R}_{\preceq_1}(C_i) \neq \mathcal{R}_{\preceq_1}(C_j),\ \mathcal{R}_{\preceq_2}(C_i) = \mathcal{R}_{\preceq_2}(C_j)\}.\end{aligned}$$

La mesure d'adéquation $\tau_X(\preccurlyeq_1, \preccurlyeq_2)$ peut donc être vue comme une version pondérée du taux de paires concordantes évalué par la quantité $\tau(\preceq_1, \preceq_2)$. De plus, quand toutes les cellules ont le même poids par rapport à μ, *i.e.* quand pour tout $(i, j) \in \{1, \ldots, N\}^2$, $\mu(\mathcal{C}_i) = \mu(\mathcal{C}_j) = 1/N$, on peut écrire que

$$\tilde{\delta}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2) = 2 \cdot \delta_\tau(\preceq_1, \preceq_2)/N^2.$$

Finalement, on obtient une version statistique du τ de Kendall entre deux pré-ordres

$\preccurlyeq_1$ et $\preccurlyeq_2$, en remplaçant simplement dans (4.10), les quantités théoriques $\mu(\mathcal{C}_i)$, avec $i \in \{1, \cdots, N\}$, par leurs contreparties empiriques calculées sur un échantillon $\mathcal{D}_n$ constitué de n observations *i.i.d.* de $\mathcal{X} \times \{-1, +1\}$. On la notera

$$\widehat{\tau}_X(\preccurlyeq_1, \preccurlyeq_2) = 1 - 2\widehat{\tilde{\delta}}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2), \tag{4.11}$$

où la distance

$$\widehat{\tilde{\delta}}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \mathbf{K}(X_i, X_j)$$

est une U-statistique de degré 2, de noyau symétrique $\mathbf{K}(.,.)$, défini pour tout couple $(x, x') \in \mathcal{X}^2$, par

$$\begin{aligned} \mathbf{K}(x, x') &= \mathbb{I}\{(s^{(1)}(x) - s^{(1)}(x')) \cdot (s^{(2)}(x) - s^{(2)}(x')) < 0\} \\ &+ \frac{1}{2}\mathbb{I}\{s_1(x) = s_1(x'),\ s_2(x) \neq s_2(x')\} \\ &+ \frac{1}{2}\mathbb{I}\{s_1(x) \neq s_1(x'),\ s_2(x) = s_2(x')\}. \end{aligned}$$

Comme nous l'avons déjà indiqué, l'optimisation du τ de Kendall nécessite de résoudre un problème NP-complet, particulièrement coûteux en temps de calcul, et il est donc préférable de travailler avec d'autres mesures. En particulier, on peut tout à fait envisager de mesurer la similarité entre deux pré-ordres $\preccurlyeq_1$ et $\preccurlyeq_2$ définis sur $\mathcal{X}$, en considérant le coefficient de corrélation *théorique* de Spearman, $\tilde{\rho}(s^{(1)}, s^{(2)})$, qui correspond au coefficient de corrélation linéaire entre les variables aléatoires $F_1(s^{(1)}(X))$ et $F_2(s^{(2)}(X))$, où pour tout $b \in \{1, 2\}$ et $X \in \mathcal{X}$, F_b est la fonction de répartition de la variable aléatoire $s^{(b)}(X)$.

En se basant sur un échantillon $(X_1, \ \ldots, \ X_n)$ de copies *i.i.d.* de la variable aléatoire X, la contrepartie empirique de ce coefficient est simplement la corrélation linéaire empirique entre les vecteurs de rangs $(s^{(1)}(X_1), \ \ldots, \ s^{(1)}(X_n))$ et $(s^{(2)}(X_1), \ \ldots, \ s^{(2)}(X_n))$. Notons que, dans le cas particulier où les fonctions $s^{(1)}$ et $s^{(2)}$ sont constantes sur les cellules d'une partition $\mathcal{P} = \{\mathcal{C}_i\}_{1 \leq i \leq N}$ de $\mathcal{X}$, définie telle que $\mu(C_i) = 1/N$ pour tout $i \in \{1 \cdots, N\}$, on a $\tilde{\rho}(s^{(1)}, s^{(2)}) = \rho(\preceq_1, \preceq_2)$.

4.2.3.3 Deux stratégies d'agrégation

Soit $\mathcal{D}_n$ un échantillon d'apprentissage constitué de n copies *i.i.d.* du couple de variables aléatoires $(X, Y) \in \mathcal{X} \times \{-1, +1\}$. En conservant les notations introduites précédemment, on pose $\mathcal{T}^{\mathbf{B}}$ la séquence de $B \geq 1$ arbres d'ordonnancement, appris sur B réplications bootstrap de l'échantillon $\mathcal{D}_n$. La collection $\mathcal{S}^{\mathbf{B}} = \{\hat{s}^{(1)}, \cdots, \hat{s}^{(B)}\}$, de fonctions de score constantes par morceaux associée à $\mathcal{T}^{\mathbf{B}}$, définit les profils de pré-ordres, $\Pi^B_{\mathcal{P}^*} = \{\preceq_1, \cdots, \preceq_B\}$, où $\mathcal{P}^*$ est la sous-partition la plus fine de l'ensemble de partitions $(\mathcal{P}^{(b)})_{1 \leq b \leq B}$ de $\mathcal{X}$, et $\Pi^B_{\mathcal{X}} = \{\preccurlyeq_1, \cdots, \preccurlyeq_B\}$.

Soit $\mathbf{X^m} = \{X'_1, \cdots, X'_m\}$ un échantillon de $m \geq 1$ copies *i.i.d.* de la variable aléatoire $X \in \mathcal{X}$, que l'on souhaite ordonner. Précédemment, nous avons défini deux types de métriques, permettant d'évaluer l'adéquation entre les pré-ordres des profils $\Pi^B_{\mathcal{P}^*}$ et $\Pi^B_{\mathcal{X}}$. On peut donc considérer deux stratégies pour définir une règle de score *médiane*, permettant d'ordonner les observations de $\mathbf{X^m}$, consistant respectivement à :

(i) définir un pré-ordre médian $\preceq^*$ sur les cellules de la sous-partition la plus fine $\mathcal{P}^*$ de $\mathcal{X}$

(ii) définir un pré-ordre médian $\preccurlyeq^*$ sur $\mathcal{X}$

Dans le premier cas, on ordonnera les observations de $\mathbf{X^m}$ en identifiant à quelle cellule C de la partition $\mathcal{P}^*$ elles appartiennent et en leur attribuant son rang $\mathcal{R}_{\preceq^*}(C)$ selon le consensus $\preceq^*$. Dans le second cas, les prédictions $\mathcal{R}_{\preccurlyeq^*}(X'_i)$, pour tout $i \in \{1, \cdots, m\}$, du pré-ordre médian $\preccurlyeq^*$ permettront d'ordonner directement les observations de $\mathbf{X^m}$. D'un point de vue empirique, en utilisant la version statistique du coefficient $\tilde{\rho}$ de Spearman, ces deux stratégies reposent sur le calcul de la moyenne des rangs prédits, respectivement, par le profil $\Pi^B_{\mathcal{P}^*}$ sur les cellules de $\mathcal{P}^*$, et par le profil $\Pi^B_{\mathcal{X}}$ sur l'échantillon $\mathbf{X^m}$.

Ces deux stratégies ne sont certes pas *équivalentes*, mais on s'attend à ce qu'elles génèrent des résultats similaires, en particulier quand n est suffisamment grand. En effet, si l'on considère par exemple la notion de pré-ordre *médian* au sens du τ de Kendall probabiliste, la distance δ_τ entre les pré-ordres du profil $\Pi^B_{\mathcal{P}^*}$, calculée sur l'échantillon $\mathbf{X}^m$, peut être vue comme l'estimateur empirique de la distance $\tilde{\delta}_{\tau_X}$ entre les pré-ordres du profil $\Pi^B_{\mathcal{X}}$, basé sur l'échantillon $\mathbf{X}^m$. Ces deux procédures conduisent donc à estimer des quantités analogues. Elles diffèrent cependant en ce sens que, dans la stratégie (ii), l'estimation du pré-ordre médian repose sur les prédictions du profil $\Pi^B_{\mathcal{X}}$ sur les observations $(X'_i)_{1\leq i\leq m}$, alors que dans la stratégie (i), elle repose sur les observations de l'échantillon d'apprentissage $\mathcal{D}_n$, puisque dans ce cas on procède au calcul des rangs médians des cellules de $\mathcal{P}^*$ et non pas des observations de $\mathbf{X}^m$.

Pour l'implémentation pratique de la procédure d'agrégation appliquée à l'algorithme TreeRank, nous avons privilégié la deuxième option, moins coûteuse en temps de calcul. En effet, la première stratégie nécessite d'identifier la sous-partition la plus fine $\mathcal{P}^*$, qui contient un très grand nombre de cellules, dès que B devient grand ou que la taille des arbres d'ordonnancement de $\mathcal{T}^{\mathbf{B}}$ est trop importante. Dans ces situations, le calcul du pré-ordre consensus $\widehat{\preceq^*}$ devient alors très coûteux, même en évaluant les distances entre pré-ordres via les mesures d'adéquation de Spearman, dont le calcul est *quasi-immédiat*.

Toutefois, nous avons observé qu'une grande majorité de ces cellules étaient vides, par construction, du moins lorsque la partition $\mathcal{P}^*$ est obtenue via le schéma proposé dans la Partie 4.2.3.1. Il conviendrait donc d'identifier ces cellules afin de les retirer. Malheureusement, dès que la règle de partitionnement utilisée pour la procédure TreeRank devient trop complexe, la détermination explicite de la frontière des cellules de $\mathcal{P}^*$, et donc l'identification de ses cellules vides, devient difficile d'un point de vue algorithmique. On pourrait envisager une approche statistique, à condition de disposer d'un nombre suffisant de données, qui consisterait à estimer le *poids* $\widehat{\mu}(C)$ des cellules $C \in \mathcal{P}^*_B$ et à ne conserver que les cellules telles que $\widehat{\mu}(C) > 0$. En observant le Lemme 14 précédent, on constate que cette démarche se rapproche alors de la stratégie (ii).

Avant de passer à la description des deux heuristiques de ré-échantillonnage mises en oeuvre pour réduire l'instabilité de la procédure TreeRank, nous présentons ci-dessous un certain nombre de résultats théoriques, relatifs notamment à la consistance des règles de score *médianes* obtenue au moyen de la stratégie (ii) retenue.

4.2.4 Résultats théoriques

Les résultats théoriques, que nous présentons ici, ont été établis pour la mesure du τ de Kendall probabiliste. En effet, nous allons voir que cette quantité est étroitement liée au critère de l'ASC. En nous appuyant sur cette relation, nous avons pu établir des résultats théoriques relatifs à la consistance des règles de score médianes. Toutefois, du fait de la relation d'équivalence (4.7), l'ensemble de ces résultats reste valable pour les deux mesures de Spearman, que nous privilégieront pour l'implémentation pratique de règles de score médianes.

En écrivant l'ASC en fonction du risque d'ordonnancement, de la même façon que dans la Partie 1.2.2.1 du Chapitre 1, on peut établir facilement un lien avec le τ théorique de Kendall.

Proposition 18 *Le τ théorique de Kendall du couple de variables aléatoires $(s(X), Y)$, où $X \in \mathcal{X}$, $Y \in \{-1, +1\}$ et $s : \mathcal{X} \to \mathbb{R}$, est relié à l'*ASC *de la fonction de score s comme suit :*

$$\frac{1}{2}\left(1 - \widetilde{\tau}(s(X), Y)\right) = 2p(1-p)\left(1 - \mathrm{ASC(s)}\right) + \frac{1}{2}\mathbb{P}\{s(X) \neq s(X'),\ Y = Y'\}, \quad (4.12)$$

où $(s(X'), Y')$ est une copie i.i.d. du couple $(s(X), Y)$.

Preuve 5 *(Preuve de la Proposition 18)*
*La démonstration de ce résultat repose simplement sur la ré-écriture de l'*ASC(s) *en fonction du risque d'ordonnancement $\mathcal{L}(s)$, défini précédemment dans le Chapitre 1. En effet, pour tout $s \in \mathcal{S}$ et tous couples (X, Y) et (X', Y') i.i.d. de $\mathcal{X} \times \{-1, +1\}$, on peut écrire que*

$$\begin{aligned}
\mathrm{ASC(s)} &= 1 - \frac{\mathcal{L}(s)}{2p(1-p)} - \frac{1}{2}\mathbb{P}\{s(X) = s(X') \mid (Y, Y') = (+1, -1)\}, \\
&= 1 - \frac{\mathbb{P}\{(s(X) - s(X'))(Y - Y') < 0\}}{2p(1-p)} - \frac{1}{2}\mathbb{P}\{s(X) = s(X') \mid (Y, Y') = (+1, -1)\}.
\end{aligned}$$

On a donc

$$\begin{aligned}
2p(1-p)(1 - \mathrm{ASC(s)}) &= \mathbb{P}\{(s(X) - s(X'))(Y - Y') < 0\} \\
&+ p(1-p) \cdot \mathbb{P}\{s(X) = s(X') \mid (Y, Y') = (+1, -1)\}, \\
&= \mathbb{P}\{(s(X) - s(X'))(Y - Y') < 0\} \\
&+ \mathbb{P}\{s(X) = s(X'), (Y, Y') = (+1, -1)\}, \\
&= \mathbb{P}\{(s(X) - s(X'))(Y - Y') < 0\} + \frac{1}{2}\mathbb{P}\{s(X) = s(X'), Y \neq Y'\}.
\end{aligned}$$

Finalement, il vient que

$$\begin{aligned}
\frac{1}{2}(1 - \tilde{\tau}(s(X), Y)) &= \mathbb{P}\{(s(X) - s(X'))(Y - Y') < 0\} + \frac{1}{2}\mathbb{P}\{s(X) = s(X'), Y \neq Y'\} \\
&+ \frac{1}{2}\mathbb{P}\{s(X) \neq s(X'), Y = Y'\}, \\
&= 2p(1-p)(1 - \mathrm{ASC(s)}) + \frac{1}{2}\mathbb{P}\{\mathrm{s(X) \neq s(X'), Y = Y'}\}.
\end{aligned}$$

Les résultats suivants découlent naturellement de la relation (4.12), mise en évidence dans la Proposition 18 précédente. On montre notamment, que la déviation en termes d'ASC entre deux fonctions de score $s^{(1)}$ et $s^{(2)}$ peut être contrôlée de façon très simple par le τ probabiliste de Kendall. C'est essentiellement pour cette raison que le critère du τ de Kendall joue un si grand rôle dans l'analyse théorique présentée dans la suite de cette partie.

Proposition 19 *(*ASC *et τ de Kendall probabiliste)*
Soit $p = \mathbb{P}\{Y = +1\} \in]0,1[$. Pour toutes fonctions de score $s^{(1)}$ et $s^{(2)}$ définies sur $\mathcal{X}$, on peut écrire :

$$|\mathrm{ASC}(\mathrm{s}^{(1)}) - \mathrm{ASC}(\mathrm{s}^{(2)})| \leq \frac{1 - \tau_{\mathrm{X}}(\preccurlyeq_1, \preccurlyeq_2)}{4\mathrm{p}(1-\mathrm{p})}.$$

Preuve 6 *(Preuve de la Proposition 19)*
Rappelons tout d'abord que $\tau_X(\preccurlyeq_1, \preccurlyeq_2) = 1 - 2\tilde{\delta}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2)$, où la distance $\tilde{\delta}_{\tau_X}$ est définie telle que :

$$\begin{aligned}
\tilde{\delta}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2) &= \mathbb{P}\{(s^{(1)}(X) - s^{(1)}(X')) \cdot (s^{(2)}(X) - s^{(2)}X')) < 0\} \\
&+ \frac{1}{2}\mathbb{P}\{s^{(1)}(X) = s^{(1)}(X'),\ s^{(2)}(X) \neq s^{(2)}(X')\} \\
&+ \frac{1}{2}\mathbb{P}\{s^{(1)}(X) \neq s^{(1)}(X'),\ s^{(2)}(X) = s^{(2)}(X')\}.
\end{aligned}$$

*De plus, de même que précédemment, on peut écrire l'*ASC(s) *comme suit, pour toute fonction $s \in \mathcal{S}$:*

$$\mathrm{ASC}(\mathrm{s}) = \frac{1}{2\mathrm{p}(1-\mathrm{p})}\mathbb{P}\{(\mathrm{s}(\mathrm{X}) - \mathrm{s}(\mathrm{X}')) \cdot (\mathrm{Y} - \mathrm{Y}') > 0\} + \frac{1}{4\mathrm{p}(1-\mathrm{p})}\mathbb{P}\{\mathrm{s}(\mathrm{X}) = \mathrm{s}(\mathrm{X}'),\ \mathrm{Y} \neq \mathrm{Y}'\}.$$

Ainsi, en utilisant l'inégalité de Jensen, on peut facilement borner la quantité

$$2p(1-p)|\mathrm{ASC}(\mathrm{s}_1) - \mathrm{ASC}(\mathrm{s}_2)|$$

par l'espérance d'une variable aléatoire de la forme :

$$\begin{aligned}
h(X, X') &= \mathbb{I}\{(s^{(1)}(X) - s^{(1)}(X')) \cdot (s^{(2)}(X) - s^{(2)}(X')) > 0\} \\
&+ \frac{1}{2}\mathbb{I}\{s^{(1)}(X) = s^{(1)}(X')\} \cdot \mathbb{I}\{s^{(2)}(X) \neq s^{(2)}(X')\} \\
&= \frac{1}{2}\mathbb{I}\{s^{(1)}(X) \neq s^{(1)}(X')\} \cdot \mathbb{I}\{s^{(2)}(X) = s^{(2)}(X')\},
\end{aligned}$$

qui correspond à la distance $\tilde{\delta}_{\tau_X}(\preccurlyeq_1, \preccurlyeq_2) = (1 - \tau_X(\preccurlyeq_1, \preccurlyeq_1))/2$.

Il est généralement vain de chercher un contrôle dans l'autre sens, des fonctions de score induisant des pré-ordres différents sur $\mathcal{X}$ pouvant avoir exactement la même ASC. Cependant, sous l'hypothèse supplémentaire d'une condition de bruit (cf [Clémençon *et al.* 2008]), le résultat suivant garantit qu'une fonction de score d'ASC quasi-optimale sera *proche* de la fonction de score s^* optimale, au sens du τ de Kendall.

Proposition 20 *(*ASC *et τ probabiliste de Kendall (bis))*
Supposons que la variable aléatoire $\eta(X)$ soit continue et qu'il existe $\epsilon \in]0, 1/2[$ tel que

$\epsilon \leq \eta(X) \leq 1-\epsilon$ *avec probabilité* 1. *Supposons de plus, qu'il existe une constante* $c < \infty$ *et* $a \in]0,1[$ *tels que la condition de bruit suivante soit satisfaite*

$$\forall x \in \mathcal{X}, \;\; \mathbb{E}\left[|\eta(X) - \eta(x)|^{-a}\right] \leq c. \tag{4.13}$$

Alors, pour tout $(s, s^*) \in \mathcal{S} \times \mathcal{S}^*$, *on a*

$$1 - \tau_X(\preccurlyeq_{s^*}, \preccurlyeq_s) \leq C \cdot (\mathrm{ASC}^* - \mathrm{ASC(s)})^{a/(1+a)},$$

où $C = 2 \cdot \max\{c^{1/(1+a)},\ p(1-p)/\epsilon^2\}$.

Preuve 7 *(Preuve de la Proposition 20)*
*En s'appuyant sur l'expression (1.31) de l'*ASC *optimale donnée dans la Proposition 6 du Chapitre 1, on peut formuler le déficit en* ASC, *associé à une fonction* $s \in \mathcal{S}$, *comme suit (cf Exemple 1 dans [Clémençon et al. 2008]) :*

$$\begin{aligned} 2p(1-p)\{\mathrm{ASC}^* - \mathrm{ASC(s)}\} &= \mathbb{E}\left[|\eta(X) - \eta(X')| \cdot \mathbb{I}\{(X,X') \in \Gamma_s\}\right] \\ &+ \mathbb{P}\{s(X) = s(X'),\ (Y,Y') = (-1,+1)\}, \end{aligned}$$

où

$$\Gamma_s = \{(X,X') \in \mathcal{X}^2 :\ (s(X) - s(X')) \cdot (\eta(X) - \eta(X')) < 0\}.$$

En combinant l'inégalité de Hölder avec la condition de bruit définie par (4.13), on peut borner la probabilité $\mathbb{P}\{(X,X') \in \Gamma_s\}$ *comme suit :*

$$\mathbb{P}\{(X,X') \in \Gamma_s\} \leq \left(\mathbb{E}\left[|\eta(X) - \eta(X')| \cdot \mathbb{I}\{(X,X') \in \Gamma_s\}\right]\right)^{a/(1+a)} \times c^{1/(1+a)}. \tag{4.14}$$

De plus, pour tout $s^* \in \mathcal{S}^*$ *on a :*

$$\tilde{\delta}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_{s^*}) = \mathbb{P}\{(X,X') \in \Gamma_s\} + \frac{1}{2}\mathbb{P}\{s(X) = s(X')\}.$$

Par ailleurs, on peut écrire naturellement que

$$p(1-p)\mathbb{P}\{s(X) = s(X') \mid (Y,Y') = (-1,+1)\} = \mathbb{E}[\mathbb{I}\{s(X) = s(X')\} \cdot \eta(X')(1-\eta(X))],$$

où cette espérance est, par hypothèse, supérieure à $\epsilon^2 \cdot \mathbb{P}\{s(X) = s(X')\}$. *En utilisant la propriété de concavité de la fonction* $t \geq 0 \mapsto t^{a/(1+a)}$ *et la borne (4.14) établie précédemment, on obtient le résultat attendu.*

Remarque 14 *(A propos de la condition de bruit) Soulignons que la condition (4.13) introduite dans [Clémençon et al. 2008], est relativement faible, au sens où elle est satisfaite pour tout* $a \in]0,1[$, *dès que la densité de* $\eta(X)$ *est bornée (cf Corollaire* 8 *dans [Clémençon et al. 2008]). Notons de plus que la contrainte imposée sur le domaine de définition de* $\eta(X)$ *garantit que le rapport de vraisemblance des distributions* $\phi(X) = dG/dH(X)$ *est borné. En effet, on rappelle que* $\phi(X) = ((1-p)/p) \cdot (\eta(X)/(1-\eta(X)))$. *Ceci implique notamment que la pente de la tangente à l'origine de la courbe optimale* COR^* *n'est pas infinie.*

Maintenant que le lien entre le critère ASC et le τ théorique de Kendall a été mis en évidence, nous nous basons sur cette mesure de similarité pour établir un certain nombre de résultats théoriques relatifs à la procédure d'agrégation. Plus précisément, nous étudions la consistance, en termes d'ASC, de règles de score *médianes* calculées à partir d'un

profil de pré-ordres, pouvant être vus comme des versions *ré-échantillonnées* d'une même relation d'ordre, consistants au sens de la distance du τ probabiliste de Kendall. Notons que par la suite, on confondra *abusivement* les notions de pré-ordre et de fonction de score sur l'espace $\mathcal{X}$.

Afin de formaliser le problème de l'agrégation, nous introduisons la notion de fonction de score *ré-échantillonnée*, que l'on notera $\mathbf{s}_{\mathcal{D}_n}(.,Z)$, où l'échantillon d'apprentissage $\mathcal{D}_n$ est constitué de n copies *i.i.d.* du couple $(X,Y) \in \mathcal{X} \times \{-1,+1\}$ et Z est une variable aléatoire, à valeurs dans un espace mesurable $\mathcal{Z}$, décrivant le mécanisme de *ré-échantillonnage* appliqué aux observations de $\mathcal{D}_n$. L'ASC généralisée d'une telle fonction est donnée par :

$$\begin{aligned} \mathrm{ASC}(\mathbf{s}_{\mathcal{D}_\mathrm{n}}(.,\mathrm{Z})) &= \mathbb{P}\{\mathbf{s}_{\mathcal{D}_n}(X,Z) > \mathbf{s}_{\mathcal{D}_n}(X',Z) \mid (Y,Y') = (+1,-1)\} \\ &+ \frac{1}{2}\mathbb{P}\{\mathbf{s}_{\mathcal{D}_n}(X,Z) = \mathbf{s}_{\mathcal{D}_n}(X',Z) \mid (Y,Y') = (+1,-1)\}, \end{aligned}$$

où (X,Y) et (X',Y') sont des observations *i.i.d.* de $\mathcal{X} \times \{-1,+1\}$, indépendantes de l'échantillon $\mathcal{D}_n$.

On parlera de *consistance* au sens de l'ASC (respectivement de *consistance forte* au sens de l'ASC) dès lors que l'on aura établi une convergence de la forme

$$\mathrm{ASC}(\mathbf{s}_{\mathcal{D}_\mathrm{n}}(.,\mathrm{Z})) \to_{\mathrm{n}\to\infty} \mathrm{ASC}^* \tag{4.15}$$

en *probabilité* (respectivement *presque sûrement*). On dira alors que la fonction de score *ré-échantillonnée* $\mathbf{s}_{\mathcal{D}_n}(.,Z)$ est ASC-consistante (respectivement fortement ASC-consistante).

Considérons l'ensemble $\mathbf{Z^B} = (Z_1, \ldots, Z_B)$, $B \geq 1$, de copies *i.i.d.* de Z tirées conditionnellement à l'échantillon d'apprentissage $\mathcal{D}_n$, définissant la collection $\mathcal{S}^{\mathbf{B}}$ de fonctions de score *ré-échantillonnées* $\mathbf{s}_n(.,Z_b)$, où $1 \leq b \leq B$. Soit $\mathcal{S}_0 \subset \mathcal{S}$ une collection de fonctions de score définies sur $\mathcal{X}$, on notera $\tilde{\mathbf{s}}(.,\mathbf{Z^B})$ la fonction de score *médiane* sur $\mathcal{S}_0$ au sens de la distance du τ probabiliste de Kendall, *i.e.* définie par

$$\tilde{\mathbf{s}}(.,\mathbf{Z^B}) \in \arg\min_{s\in\mathcal{S}_0} \sum_{b=1}^{B} \tilde{\delta}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_{\mathbf{s}_n(.,Z_b)}). \tag{4.16}$$

Le résultat suivant montre qu'une règle d'ordonnancement *médiane* d'une collection $\mathcal{S}^{\mathbf{B}}$ de fonctions de score *ré-échantillonnées* consistantes, définie au sens de l'égalité (4.16), est elle-même consistante à condition toutefois qu'elle existe. Notons de plus que la preuve de ce théorème montre que le taux de convergence est aussi préservé.

Théorème 15 *(Agrégation et* ASC*-consistance)*
Soit une collection de fonctions de score $\mathcal{S}_0 \subset \mathcal{S}$ telle que $\mathcal{S}^* \cap \mathcal{S}_0 \neq \emptyset$ et soit $B \geq 1$. On suppose que les hypothèses de la Proposition 20 sont satisfaites et que la fonction de score ré-échantillonnée $\mathbf{s}_n(.,Z)$ est ASC*-consistante (respectivement fortement* ASC*-consistante). Si pour tout $n \geq 1$, il existe une médiane $\tilde{\mathbf{s}} \in \mathcal{S}_0$, au sens du τ probabiliste de Kendall, de B réplications indépendantes de la fonction de score ré-échantillonnée $\mathbf{s}_n(.,Z)$, alors, cette règle de score agrégée $\tilde{\mathbf{s}}$ est* ASC*-consistante (respectivement, fortement* ASC*-consistante).*

Preuve 8 *(Preuve du Théorème 15)*
D'après la Proposition 19 précédente, on a, pour tout $s^* \in \mathcal{S}^*$:

$$\mathrm{ASC}^* - \mathrm{ASC}(\tilde{\mathbf{s}}) \leq \frac{\tilde{\delta}_{\tau_\mathrm{X}}(\preccurlyeq_{\mathrm{s}^*}, \preccurlyeq_{\tilde{\mathbf{s}}})}{2\mathrm{p}(1-\mathrm{p})}.$$

En appliquant l'inégalité triangulaire, on obtient

$$\tilde{\delta}_{\tau_X}(\preccurlyeq_{s^*}, \preccurlyeq_{\tilde{\mathbf{s}}}) \leq \tilde{\delta}_{\tau_X}(\preccurlyeq_{s^*}, \preccurlyeq_{\mathbf{s}_{\mathcal{D}_n}(.,Z_b)}) + \tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_{\mathcal{D}_n}(.,Z_b)}, \preccurlyeq_{\tilde{\mathbf{s}}}),$$

pour tout $b \in \{1, \ldots, B\}$. *En moyennant sur l'indice* b *et en utilisant le fait qu'en choisissant l'estimateur de* s^* *dans* $\mathcal{S}_0$, *on a*

$$\sum_{b=1}^{B} \tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_{\mathcal{D}_n}(.,Z_b)}, \preccurlyeq_{\tilde{\mathbf{s}}}) \leq \sum_{b=1}^{B} \tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_{\mathcal{D}_n}(.,Z_b)}, \preccurlyeq_{s^*}),$$

on obtient que

$$\tilde{\delta}_{\tau_X}(\preccurlyeq_{s^*}, \preccurlyeq_{\tilde{\mathbf{s}}}) \leq \frac{2}{B} \sum_{b=1}^{B} \tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_{\mathcal{D}_n}(.,Z_b)}, \preccurlyeq_{s^*}).$$

On obtient finalement le résultat attendu en combinant le résultat de la Proposition 20 avec l'hypothèse de consistance des B *fonctions de score ré-échantillonnées.*

En pratique, le calcul d'une fonction de score médiane repose sur la version empirique du τ de Kendall probabiliste, donnée par (4.11). Le résultat suivant montre que la règle de prédiction $\tilde{s} \in \mathcal{S}_0$ ainsi obtenue est asymptotiquement médiane au sens de la distance $\tilde{\delta}_{\tau_X}$, à condition que la classe $\mathcal{S}_0$, sur laquelle elle est calculée, ne soit pas trop complexe.

Théorème 16 *(Médiane empirique)*
Soit $\mathcal{S}^{\mathbf{B}} = \{s^{(1)}, \ldots, s^{(B)}\}$, $B \geq 1$, *une collection de fonctions de score induisant le profil* $\Pi_{\mathcal{X}}^{B} = \{\preccurlyeq_1, \cdots, \preccurlyeq_B\}$ *et* $\mathcal{S}_0 \subset \mathcal{S}$ *une classe de fonctions de score de dimension* VC *finie. Pour tout* $s \in \mathcal{S}$, *on pose*

$$\Delta_{B,n}(s) = \sum_{b=1}^{B} \hat{\tilde{\delta}}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_b),$$

où la distance $\hat{\tilde{\delta}}_{\tau_X}$ *est calculée sur* $n \geq 1$ *copies indépendantes de la variable aléatoire* $X \in \mathcal{X}$. *On suppose de plus que la fonction de score* $\tilde{s}_n \in \mathcal{S}$ *est telle que*

$$\Delta_{B,n}(\tilde{s}_n) = \min_{s \in \mathcal{S}_0} \Delta_{B,n}(s).$$

Avec ces notations, on a, avec probabilité 1,

$$\Delta_B(\tilde{s}_n) \to_{n\to\infty} \min_{s \in \mathcal{S}_0} \Delta_B(s), \tag{4.17}$$

où $\Delta_B(s) = \sum_{b=1}^{B} \tilde{\delta}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_b)$. *De plus, le taux de convergence est de l'ordre de* $O_{\mathbb{P}}(n^{-1/2})$.

Preuve 9 *(Preuve du Théorème 16)*
Remarquons tout d'abord que l'on a :

$$\begin{aligned} \Delta_B(\tilde{s}_n) - \min_{s \in \mathcal{S}_0} \Delta_B(s) &\leq 2 \cdot \sup_{s \in \mathcal{S}_0} |\Delta_{B,n}(s) - \Delta_B(s)| \\ &\leq 2 \sum_{b=1}^{B} \sup_{s \in \mathcal{S}_0} |\hat{\tilde{\delta}}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_b) - \tilde{\delta}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_b)|. \end{aligned}$$

Or, d'après la loi des grands nombres pour les U*-processus, établie dans le Corollaire* 5.2.3 *de [Peña & Giné 1999], pour tout* $b \in \{1, \ldots, B\}$, *on a*

$$\sup_{s \in \mathcal{S}_0} |\hat{\tilde{\delta}}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_b) - \tilde{\delta}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_b)| \to_{N\to\infty} 0. \tag{4.18}$$

Enfin, le taux de convergence d'ordre $O_{\mathbb{P}}(n^{-1/2})$ *découle du théorème central limite pour les* U*-processus (cf Théorème* 5.3.7 *de [Peña & Giné 1999]).*

Finalement, la combinaison des deux théorèmes précédents permet d'établir la consistance d'une règle de score médiane empirique, sous certaines conditions.

Corollaire 17 *(Agrégation et* ASC*-consistance bis)*
Supposons que les hypothèses du Théorème 15 soient satisfaites. Soit $\mathcal{S}_0 \subset \mathcal{S}$ de dimension VC *finie, supposons que la fonction de score randomisée* ASC*-consistante $\mathbf{s}_n(.,Z)$ appartienne à $\mathcal{S}_0$. Soit $B \geq 1$, on suppose de plus que, pour tout $n \geq 1$, il existe des règles de score médianes $\tilde{\mathbf{s}}$ et $\tilde{\mathbf{s}}_n$ dans $\mathcal{S}_0$, de B réplications indépendantes de la fonction de score $\mathbf{s}_n(.,Z)$, au sens du τ de Kendall probabiliste et de sa version empirique respectivement. Sous ces hypothèses, la règle de score empirique agrégée $\tilde{\mathbf{s}}_n$ est* ASC*-consistante. De plus, si le taux de convergence de $\mathbf{s}_n(.,Z)$ est de l'ordre de $O_{\mathbb{P}}(v_n)$, quand v_n décroît vers 0, alors celui de $\tilde{\mathbf{s}}_n$ est de l'ordre de $O_{\mathbb{P}}(\sup\{v_n, 1/\sqrt{n}\})$.*

Preuve 10 *(Preuve du Corollaire 17)*
En reprenant les mêmes arguments que dans la démonstration du Théorème 15, on obtient que

$$\tilde{\delta}_{\tau_X}(\preccurlyeq_{s^*}, \preccurlyeq_{\tilde{\mathbf{s}}_n}) \leq \frac{1}{B}\sum_{b=1}^{B} \tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_n(.,Z_b)}, \preccurlyeq_{s^*}) + \frac{1}{B}\sum_{b=1}^{B} \tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_n(.,Z_b)}, \preccurlyeq_{\tilde{\mathbf{s}}_n}).$$

De plus, de même que dans la preuve du Théorème 16, on a aussi

$$\begin{aligned} \frac{1}{B}\sum_{b=1}^{B}\{\tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_n(.,Z_b)}, \preccurlyeq_{\tilde{\mathbf{s}}_n}) \quad &- \quad \tilde{\delta}_{\tau_X}(\preccurlyeq_{\mathbf{s}_n(.,Z_b)}, \preccurlyeq_{\tilde{\mathbf{s}}_n})\} \\ &\leq \quad 2 \cdot \sup_{(s,s')\in\mathcal{S}_0^2} |\hat{\tilde{\delta}}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_{s'}) - \tilde{\delta}_{\tau_X}(\preccurlyeq_s, \preccurlyeq_{s'})|. \end{aligned} \tag{4.19}$$

En appliquant une fois encore la loi des grands nombres pour les U-processus (Corollaire 5.2.3 *de [Peña & Giné 1999]), on obtient que le terme de droite de la borne (4.19) disparaît quand $n \to \infty$. Finalement, le résultat attendu découle du Théorème 15 précédent.*

Notons que l'on peut bien entendu considérer des hypothèses plus générales sur la complexité de la classe $\mathcal{S}_0$ sur laquelle on estime le pré-ordre consensus (voir notamment les travaux présentés dans [Clémençon *et al.* 2008]). Toutefois, l'hypothèse sur la dimension *VC* est suffisante si l'on considère des arbres d'ordonnancement de profondeur maximale fixée $D \geq 1$ et pour lesquels la procédure LeafRank repose sur la mise en oeuvre d'une version pondérée de l'algorithme CART, par exemple.

Enfin, rappelons encore une fois que les résultats présentés ci-dessus, peuvent naturellement être étendus aux règles de score médianes calculées à partir d'une mesure de dissimilarité $\tilde{\delta}$ équivalente à $\tilde{\delta}_{\tau_X}$, *i.e.* satisfaisant une relation de la forme $c_1 \cdot \tilde{\delta}_{\tau_X}(.,.) \leq \tilde{\delta}(.,.) \leq c_2 \cdot \tilde{\delta}_{\tau_X}(.,.)$, pour deux constantes $0 < c_1 \leq c_2 < \infty$.

4.3 Forêts d'arbres d'ordonnancement

Maintenant que nous disposons d'une procédure d'agrégation de pré-ordres induits par des fonctions de score constantes par morceaux, nous présentons deux heuristiques de ré-échantillonnage, visant à estimer une règle de score médiane à partir d'une *forêt* d'arbres

d'ordonnancement. Plus précisément, nous proposons d'adapter le principe du *bagging* et des *forêts aléatoires*, introduites dans [Breiman 1996b] et [Breiman 2001], aux arbres d'ordonnancement. Après avoir détaillé les différentes étapes de l'application de ces deux heuristiques à la procédure d'apprentissage TreeRank, nous faisons quelques remarques sur leur mise en pratique et présentons un critère permettant d'évaluer l' *instabilité* des versions *ré-échantillonnées* de l'heuristique TreeRank. Enfin, nous illustrons les performances de ces deux approches au moyen d'une étude empirique menée sur plusieurs jeux de données simulées.

4.3.1 Ré-échantillonnage et *randomisation* de l'heuristique TreeRank

De même que dans le contexte de la classification binaire, le principe des procédures de ré-échantillonnage, que nous proposons d'adapter ici, est d'*agréger* une collection de règles de score représentées par des arbres d'ordonnancement, appris sur des réplications *bootstrap* de l'échantillon d'apprentissage $\mathcal{D}_n$.

Soit $B \geq 1$, on considère B réplications bootstrap de l'échantillon d'apprentissage, notées $\mathcal{D}_n^{(b)}$, où $b \in \{1, \cdots, B\}$, obtenues par tirage avec remise de n observations dans l'échantillon originel $\mathcal{D}_n$. En conservant les notations introduites dans la partie précédente, on note $\mathcal{T}^{\mathbf{B}}$ la collection d'arbres d'ordonnancement construits à partir des échantillons $(\mathcal{D}_n^{(b)})_{1 \leq b \leq B}$ et $\mathcal{S}^{\mathbf{B}} = \{\hat{s}^{(1)}, \cdots, \hat{s}^{(B)}\}$ l'ensemble des fonctions de score associées, définies sur $\mathcal{X}$.

Soit $\mathbf{X^m} = \{X'_1, \cdots, X'_m\}$ un échantillon constitué de $m \geq 1$ copies *i.i.d.* de la variable aléatoire $X \in \mathcal{X}$, indépendantes des réalisations de X contenues dans $\mathcal{D}_n$. Pour tout $b \in \{1, \ldots, B\}$, on note $\preccurlyeq_b$ le pré-ordre induit par la règle de score $\hat{s}^{(b)}$ sur les observations de $\mathbf{X^m}$. Comme nous l'avons indiqué précédemment, l'implémentation d'une règle de score *médiane* $\preccurlyeq^*$ repose ici sur l'agrégation des pré-ordres du profil $\Pi^B_{\mathbf{X^m}}$ selon l'approche métrique présentée précédemment. On définit donc le pré-ordre médian

$$\preccurlyeq^* \in \underset{\preccurlyeq \in \Pi_{\mathbf{X^m}}}{\arg\min} \sum_{b=1}^{B} \tilde{\delta}(\preccurlyeq, \preccurlyeq_b), \tag{4.20}$$

où $\tilde{\delta}$ est une mesure de la distance entre pré-ordres définis sur $\mathbf{X^m} \subset \mathcal{X}$.

Dans un premier temps, nous proposons d'adapter la procédure de bagging, introduite dans [Breiman 1996b]. Supposons que les paramètres de l'heuristique TreeRank (procédure LeafRank, profondeur $D \geq 1$ ou procédure d'élagage) soient fixés, de même que la taille N des réplications boostrap de D_n, le nombre $B \geq 1$ de réplications et la métrique $\tilde{\delta}$ sur l'ensemble des pré-ordres définis sur $\mathbf{X^m}$. Nous détaillons dans l'encadré ci-dessous les différentes étapes de la construction d'une règle de score *médiane*.

De la même façon que dans le contexte de la classification, on peut aussi ajouter une étape de *randomisation*, en sélectionnant de manière aléatoire un sous-ensemble de variables à chaque itération de TreeRank, à partir desquelles on procédera à l'optimisation de l'ASC locale. Posons $\mathcal{X} \subset \mathbb{R}^q$, $q \geq 1$, on note $(X^{(1)}, \cdots, X^{(q)})$ l'ensemble des q composantes du problème. On propose d'appliquer la procédure de sélection suivante, à chaque itération de TreeRank :

Procédure de bagging appliquée à l'heuristique TreeRank

- **Ré-échantillonnage.** Conditionnellement à $\mathcal{D}_n$, former des échantillons $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(B)}$ de taille N en effectuant $B \times N$ tirages avec remise dans l'échantillon $\mathcal{D}_n$.
- **Apprentissage.** Pour tout $b \in \{1, \cdots, B\}$, construire un arbre binaire orienté $\mathcal{T}^{(b)}$ via l'algorithme TreeRank, à partir de l'échantillon $\mathcal{D}^{(b)}$.
- **Prédiction.** Pour tout $b \in \{1, \cdots, B\}$, définir le pré-ordre $\preccurlyeq_b$ sur $\mathbf{X^m}$ induit par la règle de score $s^{(b)}$ et construire le profil $\Pi^B_{\mathbf{X^m}}$.
- **Agrégation.** Ordonner les observations de l'échantillon $\mathbf{X^m}$ selon un pré-ordre médian $\preccurlyeq^*_{bagg}$, défini par

$$\preccurlyeq^*_{bagg} \in \operatorname*{arg\,min}_{\preccurlyeq \in \Pi_{\mathbf{X^m}}} \sum_{b=1}^{B} \tilde{\delta}(\preccurlyeq, \preccurlyeq_b).$$

– $\mathcal{RF}_1$: *à chaque noeud $C_{d,k}$, où $d \in \{0, \cdots, D-1\}$ et $k \in \{0, \cdots, 2^d - 1\}$, d'un arbre d'ordonnancement $\mathcal{T}_D$ de profondeur $D \geq 1$, tirer aléatoirement un sous-ensemble $\mathcal{I} \subset \{1, \cdots, q\}$ d'indices, de cardinal $\#(\mathcal{I}) = q_i \leq q$ et mettre en oeuvre la procédure* LeafRank *pour scinder la cellule $C_{d,k}$ à partir du sous-ensemble de prédicteurs $(X^{(i)})_{i \in \mathcal{I}}$.*

Notons que, dans le cas où la procédure LeafRank repose sur la mise en oeuvre d'une version pondérée de l'algorithme CART, le partitionnement de chaque noeud $C_{d,k} \in \mathcal{T}_D$ peut être représenté par un *sous-arbre* de classification binaire. On peut donc envisager une deuxième procédure de *randomisation* :

– $\mathcal{RF}_2$: *à chaque noeud C du sous-arbre de classification décrivant le partitionnement d'une cellule $\mathcal{C}_{d,k} \in \mathcal{T}_D$, tirer aléatoirement un sous-ensemble de $\mathcal{J} \subset \{1, \cdots, q\}$ d'indices, de cardinal $\#(\mathcal{J}) = q_j \leq q$ et scinder la cellule C perpendiculairement à l'une des variables du sous-ensemble $(X^{(j)})_{j \in \mathcal{J}}$.*

Notons que la procédure $\mathcal{RF}_2$ consiste en réalité à résoudre la procédure LeafRank au moyen d'une *forêt aléatoire* ne contenant qu'un unique arbre de classification. Bien entendu, ces deux niveaux de *randomisation* peuvent être cumulés. Dans ce cas, la construction d'un arbre d'ordonnancement $\mathcal{T}_D$ repose sur le tirage aléatoire d'un sous-ensemble $\mathcal{I} \subset \{1, \cdots, q\}$ à chaque noeud $C_{d,k} \in \mathcal{T}_D$, et d'un sous-ensemble $\mathcal{J} \subset \mathcal{I}$ à chaque noeud C du sous-arbre de classification caractérisant le partitionnement de la cellule $C_{d,k}$.

De même que précédemment, on considère l'échantillon d'apprentissage $\mathcal{D}_n$ et l'échantillon d'observations $\mathbf{X^m}$ et on suppose fixés les paramètres de l'heuristique TreeRank, la taille N des réplications boostrap de D_n, le nombre $B \geq 1$ de réplications, la métrique $\tilde{\delta}$ sur l'ensemble des pré-ordres définis sur $\mathbf{X^m}$, ainsi que la stratégie de *randomisation* $\mathcal{RF}$.

Le détail des différentes étapes de la procédure *Ranking Forest* est donné dans l'encadré ci-dessous.

PROCÉDURE « RANKING FOREST »

- **Ré-échantillonnage.** Conditionnellement à $\mathcal{D}_n$, former des échantillons $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(B)}$ de taille N en effectuant $B \times N$ tirages avec remise dans l'échantillon $\mathcal{D}_n$.
- **Apprentissage.** Pour tout $b \in \{1, \cdots, B\}$, construire un arbre binaire orienté $\mathcal{T}^{(b)}$ via l'algorithme TREERANK combiné avec la procédure de *randomisation* des prédicteurs $\mathcal{FR}$, à partir de l'échantillon $\mathcal{D}^{(b)}$.
- **Prédiction.** Pour tout $b \in \{1, \cdots, B\}$, définir le pré-ordre $\preccurlyeq_b$ sur $\mathbf{X^m}$ induit par la règle de score $s^{(b)}$ et construire le profil $\Pi^B_{\mathbf{X^m}}$.
- **Agrégation.** Ordonner les observations de l'échantillon $\mathbf{X^m}$ selon un pré-ordre médian $\preccurlyeq^*_{RF}$, défini par

$$\preccurlyeq^*_{RF} \in \underset{\preccurlyeq \in \Pi_{\mathbf{X^m}}}{\arg\min} \sum_{b=1}^{B} \tilde{\delta}(\preccurlyeq, \preccurlyeq_b).$$

Un certain nombre de questions peuvent se poser quant au paramétrage de ces deux heuristiques. La première concerne la taille N des échantillons bootstrap tirés à partir de $\mathcal{D}_n$. Dans la procédure de bagging, telle qu'elle a été introduite dans [Breiman 1996b], ces échantillons sont pris de la même taille que l'échantillon d'apprentissage initial. Toutefois, dans [Breiman 2001], l'auteur suggère de construire des échantillons bootstrap plus petits, en particulier si le nombre n d'observations initialement disponibles est grand, afin de réduire le temps de calcul de la procédure.

La deuxième question concerne cette fois le nombre B de réplications bootstrap *nécessaires* pour définir une règle de prédiction, suffisamment robuste. Dans les exemples présentés dans [Breiman 1996b], la construction d'un classifieur repose sur l'agrégation de $B = 50$ arbres. Cependant, dans cette référence, l'auteur propose une étude empirique sur un jeu de données réelles, dans laquelle il étudie l'évolution de l'erreur de classification pour des valeurs croissantes de B, allant de 10 à 100. Dans cette expérience, l'erreur minimale est atteinte dès l'agrégation de 25 arbres, la prise en compte de forêts plus grandes induit un coût de calcul supplémentaire sans apporter de gains notables, en termes de performances de la règle de prédiction *moyennée*. Toutefois, le temps de calcul lié à la mise en oeuvre de l'algorithme de classification CART ou de classifieurs SVM étant raisonnable, nous pourrons envisager d'agréger un nombre plus important d'arbres d'ordonnancement.

La troisième question est relative au paramétrage de la stratégie de *randomisation* $\mathcal{RF}$ pour les forêts aléatoires, par le nombre de variables à conserver à chaque itération de la procédure d'apprentissage. Dans [Breiman 2001], l'auteur propose une étude empirique pour évaluer l'impact du nombre de prédicteurs retenus. Il en ressort notamment que le

cardinal du sous-ensemble considéré n'a que peu d'influence sur les performances de la procédure, à condition toutefois qu'il ne soit pas trop petit. Enfin, notons que pour ces deux procédures, l'auteur conseille d'agréger des arbres de profondeur *raisonnable*, fixée à priori, sans recourir à une procédure d'élagage qui augmenterait le temps de calcul sans apporter de gain significatif sur la performance de la règle de prédiction *médiane*.

Par ailleurs, il nous paraît important de souligner que l'interprétabilité du modèle est conservée lorsque ces heuristiques de ré-échantillonnage sont appliquées à la version TRK_{CART} de la procédure TreeRank, qui repose sur l'implémentation récursive de l'algorithme de classification CART. Il est possible par exemple d'évaluer l'impact des prédicteurs sur les performances de la règle médiane $\preccurlyeq^*$ en moyennant leurs impacts respectifs, calculés sur chaque arbre de la collection $\mathcal{T}^{\mathbf{B}}$ (cf Partie 2.3.1.3 du Chapitre 2).

Remarque 15 *(Estimation « out-of-bag »)*
*De la même façon que dans le contexte de la classification binaire, on peut envisager d'utiliser les observations laissées de côté à chaque itération b de la procédure de ré-échantillonnage, comme échantillon de test afin d'estimer, par exemple, l'*ASC *généralisée de la règle de score médiane. Ces données pourraient même être utilisées pour définir une estimation « out-of-bag »de la règle de score agrégée. Cependant, comme nous l'avons déjà souligné, dans le contexte de la classification, cette approche ne semble pas améliorer les performances de la règle de prédiction de manière significative et on peut s'attendre à ce qu'il en soit de même dans le cadre de la problématique d'ordonnancement.*

4.3.2 Mesurer l'instabilité d'un algorithme d'ordonnancement

Comme nous l'avons souligné dans l'introduction de ce chapitre, l'algorithme d'apprentissage TreeRank est instable, au sens où les règles de score produites peuvent être fortement affectées par de petits changements dans l'échantillon d'apprentissage. On attend donc des deux heuristiques de ré-échantillonnage présentées ci-dessus, qu'elles réduisent significativement cette instabilité à la fois au sens de l'ASC, *i.e.* de sorte que de légères modifications des données d'apprentissage n'engendrent pas de fluctuations significatives de l'ASC *test* mais aussi au sens de la *proximité* entre pré-ordres, *i.e.* de sorte que, malgré de telles modifications, elles produisent des ordonnancements « voisins » sur un même échantillon test.

Afin d'évaluer l'impact du bagging et des forêts aléatoires, nous proposons de quantifier l'*instabilité* de la procédure d'apprentissage considérée au moyen de la quantité suivante

$$\mathbf{Instab}(\mathbf{S}) = \mathbb{E}\left[\tilde{\delta}(\preccurlyeq_{s_{\mathcal{D}}}, \preccurlyeq_{s_{\mathcal{D}'}})\right], \tag{4.21}$$

où $\tilde{\delta}$ est une pseudo-métrique sur l'ensemble des pré-ordres induits sur $\mathcal{X}$ et $\mathbf{S} = \{s_{\mathcal{D}}, s_{\mathcal{D}'}\}$ est une collection de règles de score obtenues via l'heuristique TreeRank, appliquée à deux échantillons d'apprentissage indépendants constitués de n copies *i.i.d.* du couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$, sur lesquels on calcule l'espérance.

Notons que les heuristiques de ré-échantillonnage proposées précédemment peuvent être utilisées pour calculer l'instabilité propre à l'algorithme TreeRank. En effet, on peut interpréter la quantité

$$\widehat{\mathbf{Instab}}_n(\mathbf{S}) = \frac{2}{B(B-1)} \sum_{1 \leq b < b' \leq B} \hat{\tilde{\delta}}(\preccurlyeq_{s_{\mathcal{D}^{(b)}}}, \preccurlyeq_{s_{\mathcal{D}^{(b')}}}), \tag{4.22}$$

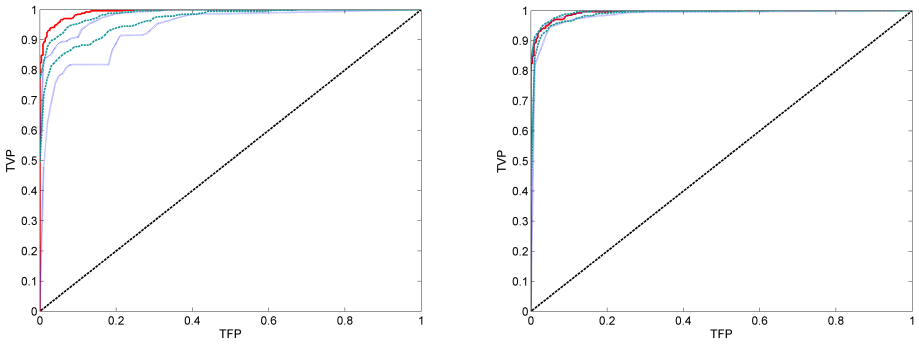
comme un estimateur *bootstrap* de (4.21).

4.3.3 Résultats expérimentaux

Les résultats empiriques suivants, obtenus sur données simulées, ont pour but de montrer l'impact de l'étape de ré-échantillonnage et de *randomisation* sur la stabilité et les performances de l'algorithme TreeRank. Après avoir présenté les résultats obtenus sur l'exemple *GaussCroix2d* cité en introduction de ce chapitre, nous introduisons trois nouveaux exemples en dimension $q = 20$, afin d'étudier l'influence de l'étape de *randomisation*. Pour toutes ces simulations, nous avons fixé le taux théorique d'observations positives p à $1/2$.

4.3.3.1 Exemple *GaussCroix2d*

Reprenons tout d'abord l'exemple *GaussCroix2d* cité en introduction. On rappelle que nous avions généré une collection $\mathcal{S}^{\mathbf{N}}$ de $N = 30$ règles de score à partir des heuristiques $\mathrm{TRK}_{\mathrm{CART}}$ et $\mathrm{TRK}_{\mathrm{SVM}}$ introduites dans le Chapitre 2. Observons maintenant l'impact de l'étape de ré-échantillonnage sur l'instabilité de l'algorithme TreeRank, mesurée au sens de l'ASC. La Figure 4.3 ci-dessous permet de comparer les enveloppes (en norme $\mathcal{L}_\infty$) des courbes $\widehat{\mathrm{COR}}_v$ obtenues via les heuristiques $\mathrm{TRK}_{\mathrm{CART}}$ et $\mathrm{TRK}_{\mathrm{SVM}}$ (tracées en pointillés bleus) et leurs versions ré-échantillonnées, $\mathrm{Bagg}_{\mathrm{CART}}$ et $\mathrm{Bagg}_{\mathrm{SVM}}$ (tracées en pointillés verts), selon la procédure de *bagging* présentée précédemment.



a. Heuristiques $\mathrm{TRK}_{\mathrm{CART}}$ et $\mathrm{Bagg}_{\mathrm{CART}}$: courbe COR^* (en rouge) et $\widehat{\mathrm{COR}}_v$ (en pointillés bleus et verts resp.).

b. Heuristique $\mathrm{TRK}_{\mathrm{SVM}}$ et $\mathrm{Bagg}_{\mathrm{SVM}}$: courbe COR^* (en rouge) et $\widehat{\mathrm{COR}}_v$ (en pointillés bleus et verts resp.).

FIGURE 4.3 – Exemple *GaussCroix2d* : enveloppes en norme $\mathcal{L}_\infty$ des courbes COR obtenues sur $N = 30$ échantillons *i.i.d.*.

On constate que l'étape de ré-échantillonnage améliore nettement la stabilité de l'heuristique $\mathrm{TRK}_{\mathrm{CART}}$, basée sur l'implémentation récursive d'une version pondérée de l'algorithme de classification CART. En effet, dans ce cas précis, la mise en oeuvre de la procédure de *bagging* permet de réduire l'aire de l'enveloppe de 50%. On remarque aussi, que les règles de score agrégées sont plus performantes que celles obtenues via l'heuristique $\mathrm{TRK}_{\mathrm{CART}}$: sur cet exemple, le *bagging* induit un gain moyen en ASC de 1.5%.

L'impact du ré-échantillonnage est beaucoup moins évident lorsque l'heuristique TreeRank est basée sur des classifieurs SVM. On peut toutefois vérifier numériquement que l'heuristique Bagg_{SVM} est plus *stable* que TRK_{SVM}, l'aire de l'enveloppe étant réduite de 40%.

4.3.3.2 Mélanges de gaussiennes en dimension 20

Pour ces trois jeux de données, nous nous sommes inspirés des simulations proposées dans [Gretton *et al.* 2008a] : nous proposons 3 exemples de complexité croissante (*GaussEasy20d*, *GaussMed20d* et *GaussHard20d*), en contrôlant la distance euclidienne $\Delta_{+/-}$ entre les moyennes (non nulles) des distributions des *positifs* et des *négatifs*, de même que l'écart entre leurs matrices de variance-covariance, $\Sigma_+ = I_q$ et $\Sigma_- = \sigma^2 \cdot I_q$, par le biais du coefficient multiplicatif σ^2, où I_q désigne la matrice *identité* dans $\mathbb{R}^q$. Les caractéristiques des trois exemples proposés sont résumées dans le tableau de la Figure 4.4 ci-dessous, et nous avons représenté leurs courbes COR* associées sur le graphe de la même figure, respectivement en bleu pour *GaussEasy20d*, en vert pour *GaussMed20d* et en rouge pour *GaussHard20d*.

Exemple	$\Delta_{+/-}$	$\log_{10}\sigma^2$	ASC*
GaussEasy20d	1	0.1	0.806
GaussMed20d	0.8	0.08	0.742
GaussHard20d	0.4	0.04	0.632

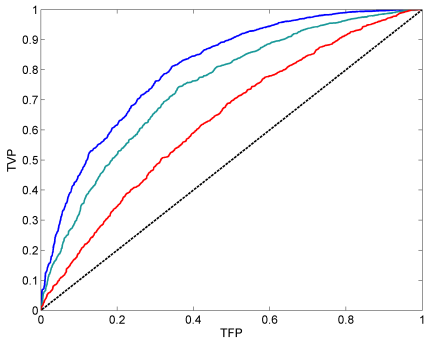


Figure 4.4 – Caractéristiques et courbes COR* des exemples *GaussEasy20d*, *GaussMed20d* et *GaussHard20d*.

Nous nous proposons de comparer 8 procédures d'apprentissage, pour lesquelles un arbre d'ordonnancement $\mathcal{T}$ est obtenu via la mise en oeuvre d'une version de l'heuristique TreeRank, éventuellement *ré-échantillonnée* et *randomisée*, basée sur l'implémentation récursive d'une version pondérée de l'algorithme de classification CART. La procédure d'apprentissage TRK_{CART} servira donc de référence.

Afin d'étudier l'impact d'une étape de *ré-échantillonnage*, nous lui comparerons l'heuristique $\text{Bagg}_{\text{CART}}$, produisant un arbre d'ordonnancement via la procédure de *bagging* décrite précédemment et 6 heuristiques « *Ranking Forest* », RF1_{CART} à RF6_{CART}, obtenues pour 6 paramétrages différents du couple de procédures de *randomisation* $(\mathcal{RF}_1, \mathcal{RF}_2)$. Pour chacune de ces heuristiques, l'agrégation de $B \geq 1$ arbres d'ordonnancement est basée sur l'optimisation du coefficient $\tilde{\rho}$ de Spearman, autrement dit, sur le calcul des *rangs moyens*. Dans la Table 4.1 ci-dessous, nous indiquons le nombre B de réplications bootstrap utilisées pour la définition des pré-ordres *médians*, le nombre $\#(\mathcal{RF}_1)$ de variables

sélectionnées aléatoirement, à chaque itération de TreeRank, pour scinder les noeuds d'un arbre principal d'ordonnancement, et le nombre $\#(\mathcal{RF}_2)$ de variables sélectionnées aléatoirement parmi ces dernières, à chaque itération de CART, pour scinder les noeuds d'un sous-arbre de classification.

Heuristique	B	$\#(\mathcal{RF}_1)$	$\#(\mathcal{RF}_2)$
TRK_{CART}	1	20	20
$\text{Bagg}_{\text{CART}}$	50	20	20
RF1_{CART}	50	5	5
RF2_{CART}	50	1	1
RF3_{CART}	50	20	5
RF4_{CART}	50	20	1
RF5_{CART}	50	10	5
RF6_{CART}	50	5	1

Table 4.1 – Configurations « *Ranking Forest* ».

Pour pouvoir quantifier l'instabilité de ces 8 procédures d'apprentissage, nous avons généré, pour chaque exemple, $N = 30$ échantillons d'apprentissage *i.i.d.*, constitués chacun de $n_a = 2000$ observations, et un unique échantillon test comprenant $n_t = 1000$ observations. Pour chaque procédure testée, nous produisons donc une collection $\mathcal{S}^{\mathbf{N}}$ de 30 règles de score, induisant N ordonnancements sur les observations de l'échantillon test sur lesquels nous estimons l'instabilité $\mathbf{Instab}_n(\mathbf{S^N})$, à partir de la pseudo-métrique $\tilde{\delta}_{\tilde{\rho}}$ de Spearman. De plus, nous évaluons les performances de ces procédures en calculant l'ASC de test *moyennée* sur les N itérations, que nous noterons $\overline{\text{ASC}}^{(t)}$ et indiquons l'écart-type associé $\hat{\sigma}^2$, afin de quantifier leur *instabilité* au sens de l'ASC.

Les résultats de ces expériences sont résumés dans le Tableau 4.2 suivant. Pour chaque procédure et chaque exemple, nous donnons, sur la première ligne, la valeur de l'ASC moyenne $\overline{\text{ASC}}^{(t)}$ accompagnée, entre parenthèses, de son écart-type estimé sur les N itérations et indiquons, sur la seconde ligne, l'instabilité estimée en italique.

Pour faciliter l'interprétation de ces résultats, nous les avons synthétisés sur les graphes *a*, *b* et *c* de la Figure 4.5 ci-dessous. Ceux-ci montrent clairement que l'étape de *ré-échantillonnage* réduit significativement l'instabilité de la procédure d'apprentissage au sens de la proximité entre pré-ordres. Cependant, ce phénomène est moins évident en ce qui concerne l'instabilité au sens de l'ASC. En effet, si l'on observe les écart-types $\hat{\sigma}^2$, indiqués entre parenthèses dans le Tableau 4.2, on constate que seules les heuristiques $\text{Bagg}_{\text{CART}}$ et RF1_{CART} sont systématiquement plus stables que la version de référence TRK_{CART}.

En particulier, on remarque que l'instabilité, au sens de l'ASC, s'accroît lorsque la procédure de *randomisation* $\mathcal{RF}_2$ est mise en oeuvre (RF3_{CART} à RF6_{CART}) et/ou que la *randomisation* est trop importante (RF2_{CART}, RF4_{CART} et RF6_{CART}). Ces résultats montrent donc que même si la mise en oeuvre d'une étape de *ré-échantillonnage* permet de produire

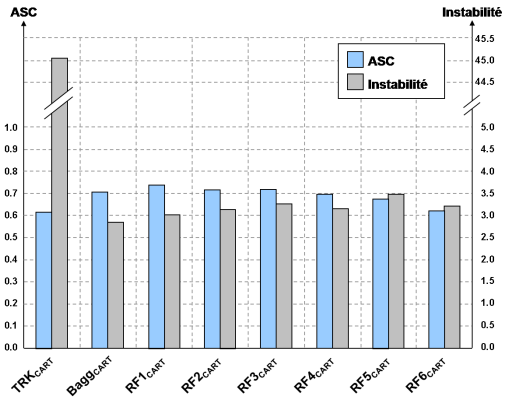
Exemple	*GaussEasy20d*	*GaussMed20d*	*GaussHard20d*
TRK_{CART}	0.620 (±0.010) *45.07*	0.597 (±0.011) *36.93*	0.510 (±0.010) *55,00*
$\text{Bagg}_{\text{CART}}$	0.708 (±0.005) *2.86*	0.673 (±0.006) *3.17*	0.578 (±0.007) *3.49*
RF1_{CART}	0.742 (±0.007) *3.02*	0.700 (±0.009) *2.92*	0.585 (±0.007) *3.58*
RF2_{CART}	0.717 (±0.009) *3.15*	0.672 (±0.006) *2.92*	0.568 (±0.011) *3.35*
RF3_{CART}	0.720 (±0.011) *3.27*	0.661 (±0.015) *3.3*	0.572 (±0.016) *3.4*
RF4_{CART}	0.698 (±0.016) *3.17*	0.637 (±0.024) *3.16*	0.553 (±0.020) *3.17*
RF5_{CART}	0.678 (±0.018) *3.49*	0.630 (±0.022) *3.16*	0.557 (±0.018) *3.24*
RF6_{CART}	0.623 (±0.025) *3.22*	0.584 (±0.028) *3.33*	0.524 (±0.020) *3.23*

Table 4.2 – Résumé des résultats numériques de la comparaison de 8 heuristiques TreeRank sur 3 exemples. Ce Tableau consigne la performance $\overline{\text{ASC}}^{(t)}$ de chaque procédure d'apprentissage, son écart-type $\hat{\sigma}^2$ étant indiqué entre parenthèses, et une évaluation de son instabilité $\widehat{\mathbf{Instab}}_n(\mathbf{S^N})$, en italique.

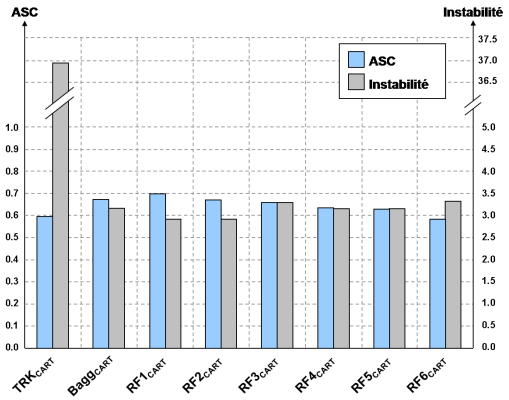
des ordonnancements plus *proches* que celui induit par la version TRK_{CART}, les procédures de *randomisation* $\mathcal{RF}_1$ et $\mathcal{RF}_2$ induisent une variabilité plus importante de l'ASC de test.

Toutefois, comme on peut le voir sur la Figure 4.5, cette *randomisation* permet, dans le même temps, d'améliorer significativement les performances de la procédure d'apprentissage, en termes d'ASC. Sur les 3 exemples traités, la règle de score la plus performante est la règle *médiane* obtenue via l'heuristique RF1_{CART}, qui induit un gain en ASC de 17% en moyenne par rapport à l'heuristique de référence.

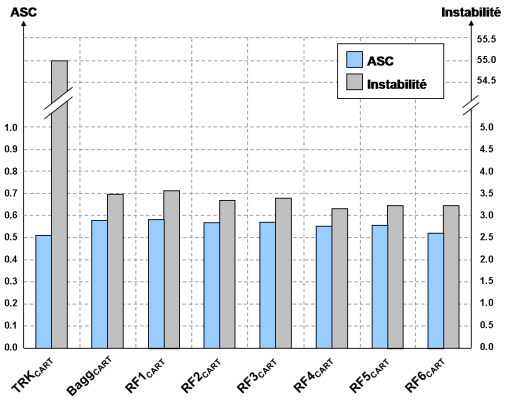
De façon générale, la mise en oeuvre d'une étape de *randomisation* permet d'égaler voire de dépasser les performances de l'heuristique $\text{Bagg}_{\text{CART}}$, induisant un gain moyen en ASC de 13% par rapport aux performances de TRK_{CART}, à condition tout de même de retenir un nombre raisonnable de variables. En effet, on constate notamment que les performances se dégradent lorsque l'on ne retient qu'une seule variable pour scinder les noeuds de l'arbre principal ou des sous-arbres de classification par exemple, comme pour les heuristiques RF2_{CART}, RF4_{CART}, n'induisant respectivement que 13% et 9% de gain moyen en ASC par rapport à TRK_{CART}. Enfin, on peut aussi souligner que, sur ces exemples, la combinaison des deux procédures $\mathcal{RF}_1$ et $\mathcal{RF}_2$ ne donne pas de très bons résultats : les heuristiques RF5_{CART} et RF6_{CART} induisent respectivement un gain moyen en ASC de 8% et 0.5% seulement, les performances de l'heuristique RF6_{CART} sur l'exemple *GaussMed20d* étant même inférieures à celles de la version de référence.



a. Exemple *GaussEasy20d*.



a. Exemple *GaussMed20d*.



a. Exemple *GaussHard20d*.

FIGURE 4.5 – Synthèse graphique de la comparaison de 8 heuristiques TREERANK sur 3 exemples : représentation de la performance $\overline{\mathrm{ASC}}^{(t)}$ et de l'instabilité $\widehat{\mathbf{Instab}}_n(\mathbf{S^N})$.

4.4 Conclusion - Perspectives

Dans ce chapitre, nous nous sommes intéressés au problème de l'*instabilité* de la procédure d'apprentissage TREERANK. Nous avons proposé de mettre en oeuvre des heuristiques de *ré-échantillonnage*, de la même façon que dans le cadre de la classification (voir notamment [Breiman 1996b], [Breiman 2001],[Breiman 1996a] et [Freund & Schapire 1999]) où la définition d'une règle de prédiction *moyenne* repose sur l'agrégation de classifieurs appris sur des réplications bootstrap d'un même échantillon selon un processus de vote à la majorité. Malheureusement, si cette procédure d'agrégation est efficace dans le contexte de la classification, elle ne permet pas d'établir un *consensus* parmi un profil de pré-ordres. En effet, nous avons vu que les règles d'ordonnancement définies par le *vote majoritaire* ne sont pas nécessairement transitives et ne satisfont donc pas le *critère de Condorcet* caractérisant la notion de pré-ordre consensus.

Nous avons donc décidé de considérer le problème de l'agrégation de pré-ordres du point de vue de l'approche métrique, *i.e.* de calculer une règle de prédiction *médiane*, au sens d'une pseudo-métrique définie sur l'ensemble des pré-ordres induits sur $\mathcal{X}$ par une fonction

de score constante par morceaux. Nous avons alors sélectionné trois pseudo-métriques afin d'évaluer la *similarité* entre pré-ordres définis sur les cellules d'une partition finie de $\mathcal{X}$: le τ de Kendall, la règle de Spearman et le coefficient ρ de Spearman, que nous avons ensuite étendu au cas de pré-ordres induits par des fonctions de score constantes par morceaux sur l'espace $\mathcal{X}$ de cardinal infini.

En nous appuyant sur la relation que nous avons pu établir entre le τ de Kendall probabiliste et le critère ASC, nous avons montré la consistance, sous certaines conditions, de la règle de score *médiane* au sens de cette pseudo-métrique. Cependant, nous soulignons que ces résultats restent valables pour les pseudo-métriques de Spearman, étant donné la relation d'équivalence (4.7) établie par le Théorème 13 de [Fagin *et al.* 2006] entre ces trois mesures de similarité.

Nous avons ensuite proposé deux versions *ré-échantillonnées* de l'heuristique TreeRank, en adaptant les procédures de bagging et des forêts aléatoires introduites respectivement dans [Breiman 1996b] et [Breiman 2001] pour l'algorithme de régression et de classification CART. La première consiste simplement à agréger, au sens de l'approche métrique, $B \geq 1$ arbres d'ordonnancement construits à partir de B réplications bootstrap d'un même échantillon d'apprentissage. La seconde repose sur le même principe, mais comprend une étape supplémentaire de *randomisation* des prédicteurs : un sous-ensemble des variables du problème est sélectionné aléatoirement pour procéder à la scission de chaque noeud de l'arbre principal d'ordonnancement et/ou de chaque noeud des sous-arbres de classification, par exemple lorsque l'heuristique TreeRank est vue comme l'implémentation récursive d'une version pondérée de l'algorithme de classification CART.

Enfin, nous avons mené une étude empirique afin d'illustrer le comportement de ces procédures appliquées à deux jeux de données simulées. Avec ces expériences nous avons pu constater que l'étape de *ré-échantillonnage* permettait à la fois de réduire nettement l'instabilité des procédures d'apprentissage, au sens de la distance entre les ordonnancements produits par des versions *ré-échantillonnées* d'une même fonction de score, et d'améliorer les performances au sens de l'ASC. Cette étude nous a aussi permis de mettre en évidence l'intérêt, tout comme les limites, de l'étape de *randomisation* incluse dans la procédure Ranking Forest. En effet, cette approche induit un gain significatif en termes de performances de la règle de score, y compris par rapport à la procédure de bagging, à condition toutefois que le nombre de variables retenues pour la scission des noeuds ne soit pas trop faible, mais dans le même temps, elle à aussi tendance à augmenter l'instabilité au sens de l'ASC.

Pour aller plus loin, il conviendrait d'étudier plus en détail l'influence du nombre de variables retenues par les procédures de *randomisation* sur les performances de la règle de prédiction obtenue. Aussi, nous consacrons le Chapitre 5 à une étude empirique plus détaillée de différentes versions de l'heuristique TreeRank, reposant sur la mise en oeuvre de l'algorithme CART et de SVM, auxquelles nous appliquons les deux procédures de *ré-échantillonnage* que nous venons d'introduire. Nous proposons notamment une comparaison de ces heuristiques entre elles et avec quelques méthodes de l'état-de-l'art, sur un ensemble de données réelles et simulées.

Troisième partie

Applications

Chapitre 5

Applications

Ce chapitre est consacré à l'étude empirique des performances de la méthode TreeRank et se décompose en deux parties bien distinctes. Dans la première, nous présentons deux études empiriques dans lesquelles nous comparons diverses heuristiques d'ordonnancement sur une collection de jeux de données artificielles et réelles. Dans un premier temps, nous proposons de comparer huit versions de l'algorithme TreeRank, dont cinq sont fondées sur l'implémentation récursive d'une version pondérée de l'algorithme de classification CART et trois sur l'utilisation de classifieurs SVM. L'objectif de cette première étude est principalement d'étudier l'impact des procédures de *ré-échantillonnage* et de *randomisation* sur les performances de la méthode TreeRank. Puis, dans un second temps, nous proposons de comparer notre procédure de scoring avec trois de ses concurrentes : la méthode RankBoost, basée sur une procédure de *boosting*, et les méthodes RankSVM et RankRLS, reposant sur la mise en oeuvre d'heuristiques *de type* SVM. Ces deux études empiriques sont donc présentées dans la première partie de ce chapitre, après la description des bases de données utilisées et des critères utilisés pour évaluer les performances des heuristiques considérées.

Dans la deuxième partie de ce chapitre, nous nous intéressons à la problématique industrielle ayant motivé ces travaux de thèse CIFRE : l'*objectivation des prestations*. Après avoir présenté cette problématique de manière générale et la méthodologie actuellement mise en place chez le constructeur automobile Renault, nous détaillons l'application principale de ces travaux de thèse : l'étude de l'*indice brio*, dont l'objectif était de construire un *indice du brio*, afin d'évaluer la *sensation à l'accélération* perçue par les utilisateurs pour un panel de véhicules. Nous décrivons notamment l'ensemble des données disponibles pour cette étude, issues d'une vaste *campagne d'essais*, à partir desquelles nous avons extrait deux bases de données. Nous présentons ensuite les résultats obtenus sur ces bases par trois versions de l'heuristique TreeRank fondées sur l'algorithme CART, que nous comparons également à une méthode de type LASSO proposée dans [Germain 2009] pour l'*objectivation* de l'*agrément de conduite*.

5.1 Etude empirique des performances de la méthode de scoring TreeRank

Dans cette première partie, nous présentons deux études expérimentales permettant d'évaluer les performances de la méthode de scoring TreeRank. Dans la première étude,

présentée dans la Partie 5.1.3, nous proposons de comparer diverses versions de l'heuristique TreeRank, fondées sur la mise en oeuvre d'une version pondérée de l'algorithme de classification CART ou de SVM et éventuellement *ré-échantillonnées*, au moyen des procédures de *bagging* ou de *forêts aléatoires* introduites dans le chapitre précédent. Pour la seconde étude, présentée dans la Partie 5.1.4, nous retenons l'heuristique TreeRank présentant (globalement) les meilleurs résultats sur la collection de jeux de données considérée et nous la comparons à trois méthodes de type ERM : RankBoost ([Freund *et al.* 2003]), RankSVM ([Joachims 2002b]) et RankRLS ([Pahikkala *et al.* 2007]), reposant respectivement sur la mise en oeuvre d'une procédure de type *boosting* ([Freund & Schapire 1999]), de classifieurs SVM et de *moindres carrés régularisés*. Mais avant de présenter ces deux études, nous consacrons les Parties 5.1.1 et 5.1.2 suivantes à la description de la collection des 13 jeux de données (réelles et simulées) sur lesquels nous comparons les différentes heuristiques et des critères utilisés pour évaluer leurs performances.

5.1.1 Description des données

Dans les Parties 5.1.3 et 5.1.4 suivantes, les performances des divers algorithmes de scoring considérés sont évaluées sur une collection de 3 jeux de données simulées et de 10 jeux de données réelles, que nous qualifierons de données *benchmark*, issus de la banque de donnée en ligne de l'UCI (http ://archive.ics.uci.edu/ml/).

Les trois exemples simulés que nous avons retenus ont déjà été introduits dans le chapitre précédent : il s'agit des mélanges de gaussiennes *GaussEasy20d*, *GaussMed20d* et *GaussHard20d*, définis sur l'espace $\mathbb{R}^{20} \times \{-1, +1\}$. Nous rappelons que ces trois exemples sont inspirés des simulations proposées dans [Gretton *et al.* 2008a], au sens où la *complexité* du problème d'ordonnancement est contrôlée par la distance euclidienne $\Delta_{+/-}$ entre les moyennes (non nulles) des distributions des *positifs* et des *négatifs* et par l'écart entre leurs matrices de variance-covariance, $\Sigma_+ = I_{20}$ et $\Sigma_- = \sigma^2 \cdot I_{20}$, par le biais du coefficient multiplicatif σ^2, où I_{20} désigne la matrice *identité* dans $\mathbb{R}^{20}$ (cf Partie 4.3.3.2 du Chapitre 4). Les caractéristiques principales des échantillons générés sont rappelées dans le Tableau 5.1 ci-dessous.

Exemple	Taille de l'échantillon	Nombre de variables	Taux de positifs	Variables catégorielles	Descriptif
GaussEasy20d	2000	20	0.5	non	Mélange de gaussiennes
GaussMed20d	2000	20	0.5	non	Mélange de gaussiennes
GaussHard20d	2000	20	0.5	non	Mélange de gaussiennes

Table 5.1 – Description des données simulées.

Afin d'évaluer les performances de la méthode TreeRank sur des jeux de données plus *réalistes*, nous avons choisi dix exemples dans la banque de données de l'UCI, dont nous résumons les caractéristiques principales dans le Tableau 5.2 suivant. Cette collection contient trois jeux de données relatifs à l'évaluation du risque d'allocation de crédits financiers (*Aus-*

tralian Credit, *German Credit* et *Japanese Credit*), quatre autres relatifs à la détection de diverses pathologies médicales (*Breast Cancer Diagnosis*, *Breast Cancer Original*, *Hepatitis* et *Heart Disease*), l'exemple *Autos MPG* relatif à l'évaluation du niveau de consommation de véhicules automobiles, l'exemple *Congressional Vote* concernant la détermination de la tendance politique (*démocrate* ou *républicaine*) selon les votes au congrès américain et enfin l'exemple *Ionosphere* relatif à la détection de structures spécifiques dans la ionosphère par l'enregistrement de retours de signaux radars.

Neuf de ces jeux de données relèvent d'un problème de classification binaire, où l'objectif est d'expliquer la *sortie binaire* Y au moyen d'un ensemble de variables continues et catégorielles. Dans l'exemple *Autos MPG* par contre, la variable à expliquer Y, qui représente la consommation de véhicules automobiles en *miles per gallon* est continue. Nous nous sommes donc ramenés à un problème de classification binaire en considérant qu'un *bon* véhicule consomme *peu* et en attribuant ainsi à chaque véhicule dont la consommation est inférieure (resp. supérieure) à $\max(Y)/2$ une étiquette *positive* (resp. *négative*). Dans la banque de données de l'UCI, la plupart de ces jeux de données sont divisés en un échantillon d'apprentissage et un échantillon de validation. Ici, nous avons regroupé l'ensemble des observations en un seul échantillon dans le but de mettre en oeuvre une procédure de *validation croisée par blocs* (cf Chapitre 3) et de produire des estimateurs *moyennés* et plus *robustes* des performances des heuristiques. Enfin, nous avons choisi de ne pas faire de *complétion* et de simplement supprimer les observations contenant des valeurs manquantes pour certaines variables.

5.1.2 Critères d'évaluation des performances

Dans les deux études présentées, nous comparons les performances des diverses heuristiques mises en oeuvre sur ces données en termes d'ASC. Pour chaque exemple, nous proposons d'estimer l'ASC par une procédure de validation croisée par blocs. Aussi, en notant $\mathcal{D}_n$ l'ensemble des observations d'apprentissage, nous constituons une collection de $V = 10$ sous-échantillons, notés $\{\mathcal{D}^{(1)}, \cdots, \mathcal{D}^{(V)}\}$, afin d'estimer l'ASC empirique de test *moyenne* $\overline{\mathrm{ASC}}^{(t)}$ donnée par :

$$\overline{\mathrm{ASC}}^{(t)} = \frac{1}{V} \sum_{v=1}^{V} \widehat{\mathrm{ASC}}^{(v)},$$

où pour tout $v \in \{1, \cdots, V\}$, $\widehat{\mathrm{ASC}}^{(v)}$ est l'ASC empirique associée à la règle de score $s^{(-v)}$, apprise sur l'échantillon d'apprentissage $\mathcal{D}^{(-v)} = \mathcal{D}_n \setminus \mathcal{D}^{(v)}$ et calculée sur l'échantillon $\mathcal{D}^{(v)}$. On notera $\hat{\sigma}^2$ l'écart-type de l'ASC empirique de test calculé sur les V itérations de la procédure de validation.

Par ailleurs, rappelons que contrairement aux diverses approches proposées dans l'état de l'art, la méthode TreeRank ne se contente pas d'optimiser le critère ASC de manière *globale* mais procède de façon itérative. Elle produit ainsi des règles de score *optimales* non seulement au sens de l'ASC mais plus largement au sens de la courbe COR. On peut donc s'attendre à ce que cette approche ordonne les *meilleures* observations de l'échantillon de façon plus pertinente que ses concurrentes. Malheureusement, comme nous l'avons déjà souligné dans la Partie 1.2.2.3 du Chapitre 1, l'évaluation de l'ASC empirique ne permet pas de mettre en évidence ce phénomène. Aussi, nous proposons de considérer le critère

Exemple	Taille de l'échantillon	Nombre de variables	Taux de positifs	Variables catégorielles	Descriptif
Australian Credit	690	14	0.44	oui	Evaluation du risque d'allocation de crédit
German Credit	1000	20	0.7	oui	Evaluation du risque d'allocation de crédit
Japanese Credit	690	15	0.45	oui	Evaluation du risque d'allocation de crédit
Breast Cancer Diagnosis	569	30	0.37	oui	Détection de tumeurs malignes
Breast Cancer Original	683	9	0.35	oui	Détection de tumeurs malignes
Heart Disease	270	13	0.44	oui	Détection de problèmes cardiaques
Hepatitis	112	18	0.17	oui	Détection de l'hépatite
Autos MPG	392	7	0.53	oui	Niveau de consommation de véhicules
Congressional Vote	232	16	0.53	oui	Résultats de vote du congrès Américain
Ionosphere	351	34	0.64	oui	Détection de structures dans la ionosphère

Table 5.2 – Description des données *benchmark*.

d'ASC *tronqué* introduit dans [Clémençon & Vayatis 2007].

Notons u la proportion des *meilleures* observations sur lesquelles on souhaite se focaliser. Ces observations constituent l'ensemble de niveau $C_{s,u} = \{x \in \mathcal{X} \mid s(x) \geq \mathcal{Q}_s(u)\}$, pour une règle de score $s \in \mathcal{S}$, où $\mathcal{Q}_s(u)$ représente le quantile d'ordre $(1-u)$ de la variable aléatoire $s(X)$. En posant les notations suivantes pour tout couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$,

$$\begin{aligned} \alpha_s(u) &= \mathbb{P}\{s(X) \geq \mathcal{Q}_s(u) \mid Y = -1\} \text{ et} \\ \beta_s(u) &= \mathbb{P}\{s(X) \geq \mathcal{Q}_s(u) \mid Y = +1\}, \end{aligned}$$

on peut re-paramétrer par la proportion u la courbe COR associée à la fonction s sous la forme du graphe suivant :

$$\begin{aligned} \mathrm{COR(s)} :]0,1[&\rightarrow [0,1]^2 \\ u &\rightarrow (\alpha_s(u), \beta_s(u)). \end{aligned}$$

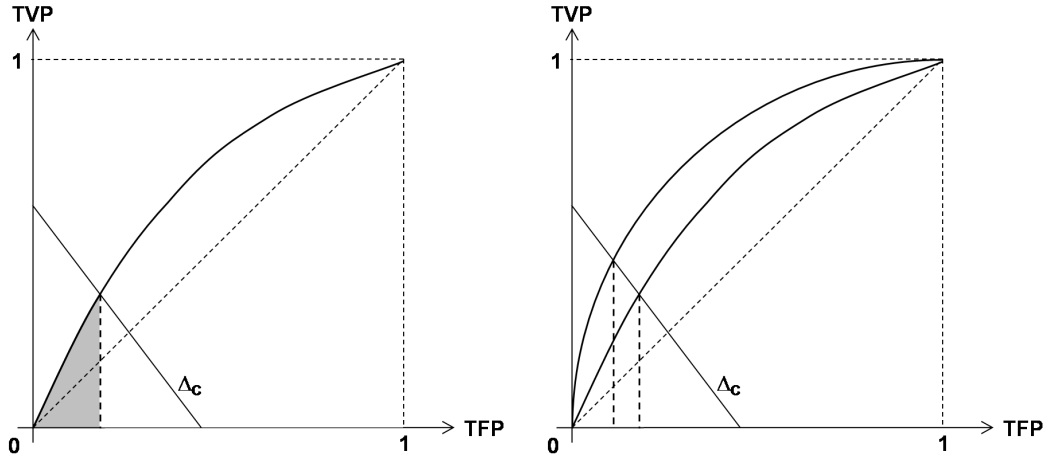
Un critère très intuitif est l'ASC *partielle*, initialement introduite dans [Dodd & Pepe 2003] afin de se focaliser sur les *meilleures* observations et définie comme suit :

$$\mathrm{PartASC(s,u)} = \int_0^{\alpha_s(u)} \beta_s(\alpha) \mathrm{d}\alpha.$$

Cependant, dans [Clémençon & Vayatis 2007] la pertinence de ce critère est remise en question. Afin de présenter l'argument géométrique avancé par les auteurs, il nous faut introduire la notion supplémentaire de *droite de contrôle*. Pour cela, il suffit de remarquer que pour un taux de *positifs* p et pour une proportion $u = u_0$ fixés, les taux de vrais positifs et de faux positifs associés à une fonction $s \in \mathcal{S}$ sont liés linéairement comme suit :

$$u_0 = p\beta_s(u_0) + (1-p)\alpha_s(u_0). \tag{5.1}$$

Nous appellerons *droite de contrôle* la droite Δ_c d'équation (5.1). Celle-ci permet de *délimiter* la portion de la courbe $\mathrm{COR}(\mathrm{s},.)$ associée aux *meilleures* observations comme étant le graphe reliant l'origine du plan $(\mathrm{TFP}, \mathrm{TVP})$ au point d'intersection entre la courbe COR et la droite Δ_c. Cette portion de courbe étant identifiée, on peut facilement visualiser l'ASC partielle $\mathrm{PartASC}(\mathrm{s}, \mathrm{u}_0)$ représentée en gris sur le graphe a de la Figure 5.1 ci-dessous.



a. Droite de contrôle et ASC partielle. b. Comparaison d'ASC partielles.

FIGURE 5.1 – Critère de l'ASC partielle.

L'argument géométrique avancé dans le Lemme 18 de [Clémençon & Vayatis 2007] montre simplement que le maximum de l'ASC partielle n'est pas nécessairement atteint par la fonction de régression et souligne ainsi en quoi ce critère ne semble pas pertinent pour la problématique d'ordonnancement. En effet, comme on peut le voir sur le graphe b de la Figure 5.1, pour une proportion $u = u_0$ fixée, la courbe COR^* associée à la probabilité η présente bien un taux de vrais positifs supérieur à celui d'une règle de score s quelconque, mais dans le même temps, son taux de faux positifs diminue et devient inférieur à celui atteint par la fonction s. Ainsi, il est fort probable de trouver une fonction de score $s \in \mathcal{S}$ *non-optimale* au sens de l'ASC mais telle que $\mathrm{PartASC}(\mathrm{s}, \mathrm{u}_0) > \mathrm{PartASC}(\eta, \mathrm{u}_0)$. Aussi, les auteurs de cette contribution proposent un nouveau critère permettant de contourner ce problème (cf Théorème 20 de [Clémençon & Vayatis 2007]). Ils introduisent ainsi la notion d'ASC *tronquée* ou *locale* définie comme suit, pour tout $s \in \mathcal{S}$:

$$\mathrm{LocASC}(\mathrm{s}, \mathrm{u}_0) \quad = \quad \mathbb{P}\{s(X) > s(X'), s(X) \geq \mathcal{Q}_s(u_0) \mid (Y, Y') = (+1, -1)\},$$

où (X', Y') est une copie *i.i.d.* du couple de variables aléatoires $(X, Y) \in \mathcal{X} \times \{-1, +1\}$.

Notons que ce critère est lié à l'ASC partielle comme suit :

$$\mathrm{LocASC}(\mathrm{s},\mathrm{u}_0) \quad = \quad \mathrm{PartASC}(\mathrm{s},\mathrm{u}_0) + \beta_{\mathrm{s}}(\mathrm{u}_0) - \alpha_{\mathrm{s}}(\mathrm{u}_0)\beta_{\mathrm{s}}(\mathrm{u}_0),$$

et peut être considéré comme une *normalisation* de celle-ci. De même que précédemment, nous avons représenté l'ASC tronquée en gris sur la Figure 5.2 ci-dessous. Divers résultats théoriques relatifs à l'optimisation empirique de ce critère peuvent être trouvés dans la Section 4 de [Clémençon & Vayatis 2007].

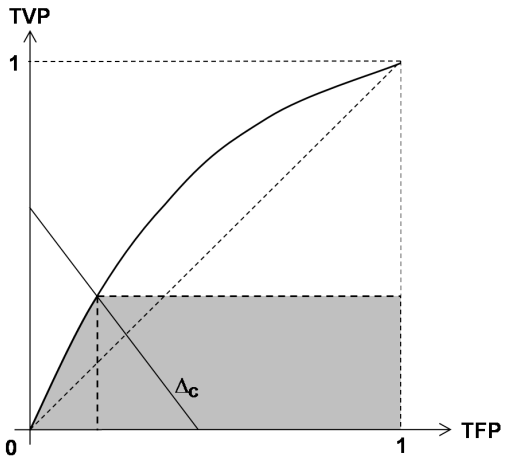


Figure 5.2 – Critère de l'ASC tronquée.

5.1.3 TreeRank dans tous ses états

Dans cette première étude empirique, nous nous proposons de comparer huit versions de l'heuristique TreeRank, dont cinq sont fondées sur l'implémentation récursive d'une version pondérée de l'algorithme de classification CART et trois reposent sur la mise en oeuvre de classifieurs SVM. En ce qui concerne les cinq premières, nous allons considérer l'heuristique $\mathrm{TRK}_{\mathrm{CART}}$ ainsi que la version *ré-échantillonnée* $\mathrm{Bagg}_{\mathrm{CART}}$ et les versions *ré-échantillonnées et randomisées* $\mathrm{RF1}_{\mathrm{CART}}$, $\mathrm{RF3}_{\mathrm{CART}}$ et $\mathrm{RF5}_{\mathrm{CART}}$, que nous avons introduites dans le chapitre précédent. Dans le cas de la mise en oeuvre de classifieurs SVM, nous testons pour chaque exemple deux versions de $\mathrm{TRK}_{\mathrm{SVM}}$ (cf Chapitre 2) fondées respectivement sur l'utilisation d'un noyau gaussien Rbf et d'un noyau linéaire et nous conservons l'heuristique présentant les meilleures performances, en termes d'ASC. Pour ce noyau, nous considérons les versions $\mathrm{Bagg}_{\mathrm{SVM}}$ et $\mathrm{RF1}_{\mathrm{SVM}}$ *ré-échantillonnées* respectivement selon les procédures de *bagging* et de *forêt aléatoire* introduites au chapitre précédent.

Pour chaque exemple, nous évaluons l'ASC empirique totale et les ASC empiriques tronquées pour trois proportions $u \in \{0.05, 0.1, 0.2\}$, *moyennées* sur les $V = 10$ échantillons de validation $(\mathcal{D}^{(v)})_{1\leq v\leq V}$. Pour les diverses versions *ré-échantillonnées* de l'heuristique TreeRank, on considère à chaque itération $v \in \{1, \cdots, V\}$ de la procédure de validation croisée par blocs, $B = 100$ réplications bootstrap de l'échantillon d'apprentissage $\mathcal{D}^{(-v)}$ pour construire les règles de score *médianes*, au sens du coefficient $\tilde{\rho}$ de Spearman.

Soit $q \geq 1$ le nombre de variables *explicatives* disponibles pour chaque exemple considéré,

on note $\#(\mathcal{RF}_1)$ le nombre de variables sélectionnées aléatoirement, à chaque itération de TreeRank, pour scinder les noeuds de l'arbre principal d'ordonnancement et $\#(\mathcal{RF}_2)$ le nombre de variables sélectionnées aléatoirement parmi ces dernières, à chaque itération de l'algorithme CART, pour scinder les noeuds d'un sous-arbre de classification. Enfin, pour tout $z \in \mathbb{R}$, on notera $\lfloor z \rfloor$ la partie entière de z. Munis de ces notations, nous rappellons ci-dessous les procédures de *randomisation* $(\mathcal{RF}_1, \mathcal{RF}_2)$ mises en oeuvre pour les trois versions de *forêts aléatoires* considérées :

- RF1_{CART} et RF1_{SVM} : pour chaque itération $b \in \{1, \cdots, B\}$, $\#(\mathcal{RF}_1) = \lfloor q/4 \rfloor$ variables sont sélectionnées aléatoirement à chaque noeud de l'arbre principal d'ordonnancement $\mathcal{T}^{(b)}$ pour procéder à sa scission en deux sous-ensembles non-vides,

- RF3_{CART} : pour chaque itération $b \in \{1, \cdots, B\}$, $\#(\mathcal{RF}_2) = \lfloor q/4 \rfloor$ variables sont sélectionnées aléatoirement à chaque noeud des sous-arbres de classification pour procéder à leur scission en deux sous-ensembles non-vides,

- RF5_{CART} : pour chaque itération $b \in \{1, \cdots, B\}$, $\#(\mathcal{RF}_1) = \lfloor q/2 \rfloor$ variables sont sélectionnées aléatoirement à chaque noeud de l'arbre principal d'ordonnancement $\mathcal{T}^{(b)}$ et $\#(\mathcal{RF}_2) = \lfloor q/4 \rfloor$ variables sont sélectionnées aléatoirement parmi celles-ci à chaque noeud des sous-arbres de classification.

Les résultats de cette étude sont résumés dans le Tableau A.1 de l'Annexe A qui contient, pour chaque exemple et chaque heuristique, la valeur de l'ASC moyenne $\overline{\text{ASC}}^{(t)}$ accompagnée, entre parenthèses, de son écart-type estimé sur les V itérations de validation et dans le Tableau A.2 de l'Annexe A contenant les ASC tronquées moyennes pour les trois proportions $u \in \{0.2, 0.1, 0.05\}$ (dans cet ordre), accompagnées de leurs écart-types entre parenthèses. Pour plus de lisibilité, nous proposons de visualiser ces résultats graphiquement sur la Figure 5.3 ci-dessous. Celle-ci présente une collection d'histogrammes, correspondant chacun à un des exemples traités, constitués des *couples* $(\overline{asc}^{(t)}, \hat{\sigma}^2)$ pour chaque heuristique, les ASC partielles étant représentées par les *scissions* horizontales portées par la *barre* représentant la valeur de l'ASC empirique.

On peut commencer par remarquer que ces résultats confirment un certain nombre de points vus dans les chapitres précédents. Notamment, tout comme dans les exemples traités au Chapitre 2, on constate que les meilleurs résultats, que ce soit en termes d'ASC ou d'*instabilité* (mesurée par l'écart-type $\hat{\sigma}^2$ de $\overline{\text{ASC}}^{(t)}$), sont obtenus par les heuristiques fondées sur les classifieurs SVM. De plus, tout comme dans le Chapitre 4, on note aussi que les versions *ré-échantillonnées* produisent des règles d'ordonnancement plus performantes que les heuristiques TRK_{CART} et TRK_{SVM}. Ce dernier point est surtout visible en ce qui concerne les heuristiques fondées sur l'algorithme de classification CART. En effet, dans la plupart des exemples traités, les performances de l'heuristique TRK_{SVM} ne sont que légèrement améliorées par les procédures de *ré-échantillonnage*. Soulignons de plus que dans le cas de la mise en oeuvre de Machines à Vecteurs Supports, la *randomisation* n'apporte pas nécessairement d'amélioration : sur les 13 exemples traités, les performances des heuristiques Bagg_{SVM} et RF1_{SVM} sont similaires.

Par ailleurs, comme nous l'avions déjà remarqué dans le chapitre précédent, on constate que sur les 3 jeux de données simulés, les performances des heuristiques *ré-échantillonnées* diminuent, au sens de l'ASC et de l'instabilité, lorsque la randomisation devient trop im-

portante : sur les exemples *GaussEasy20d* et *GaussMed20d*, la version la plus *randomisée* RF5_{CART} est celle qui présente l'ASC la plus faible et dans le même temps l'instabilité la plus importante. A l'inverse, il est intéressant de remarquer que sur les données *benchmark*, la *randomisation* ne dégrade pas (voire améliore) les performances des heuristiques *ré-échantillonnées*.

On constate aussi que sur l'ensemble des 10 jeux de données réelles étudiés, les 4 heuristiques de forêts aléatoires, qu'elles soient fondées sur l'algorithme CART ou sur des classifieurs SVM, et l'heuristique Bagg_{SVM} présentent des performances similaires. De plus, celles-ci sont généralement (nettement) supérieures à celles des 3 versions TRK_{CART}, TRK_{SVM} et $\text{Bagg}_{\text{CART}}$. Ainsi, cette première étude empirique permet de mettre en évidence, d'une part l'intérêt de mettre en oeuvre une procédure de *ré-échantillonnage* et d'autre part, l'impact de la *randomisation* sur les heuristiques fondées sur l'algorithme de classification CART, qui leur permet d'atteindre des performances similaires à celles des versions *ré-échantillonnées* Bagg_{SVM} et RF1_{SVM} basées sur des classifieurs SVM. Soulignons que les résultats du Tableau A.2 relatifs aux ASC tronquées confirment ces deux tendances.

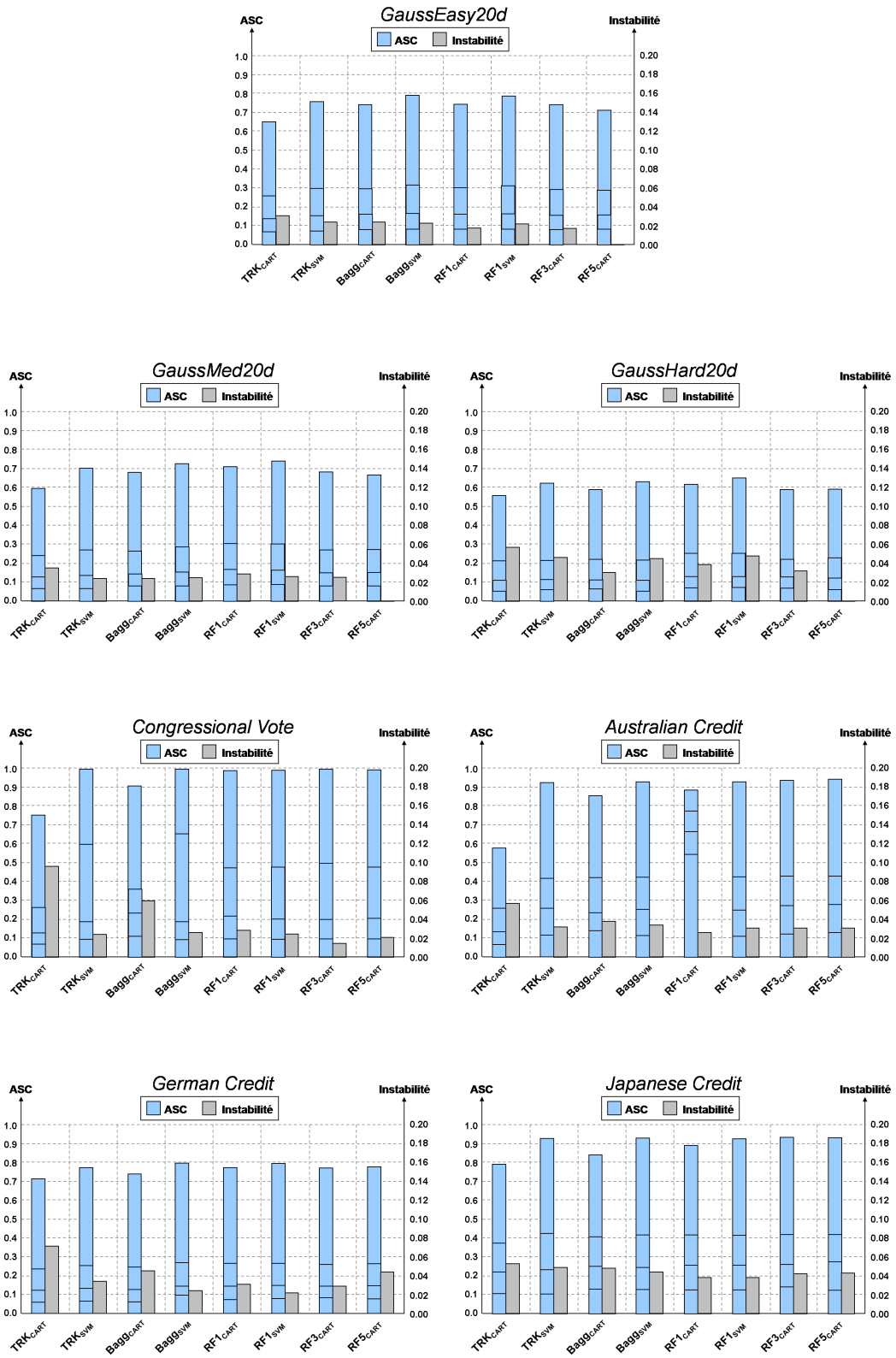
5.1.4 TreeRank et quelques concurrents

Dans cette deuxième étude, nous nous proposons de comparer la méthode de scoring TreeRank présentée dans ce manuscrit avec quelques unes de ses concurrentes. Comme nous l'avons souligné dans l'introduction de ce chapitre, nous en avons retenu 3, fondées respectivement sur la mise en oeuvre d'une procédure de type *boosting* (RankBoost) et d'heuristiques de type SVM (RankSVM et RankRLS). Comme nous allons le voir, ces 3 méthodes reposent en réalité sur un même principe qui consiste à définir une règle d'ordonnancement en résolvant un problème de classification binaire sur les paires d'observations de l'échantillon d'apprentissage.

Afin d'établir la comparaison avec la méthode TreeRank, nous avons choisi de sélectionner deux heuristiques parmi les huit testées dans la partie précédente : une première basée sur la mise en oeuvre d'une version pondérée de l'algorithme CART, RF3_{CART}, et une seconde fondée sur la mise en oeuvre de classifieurs SVM, RF1_{SVM}. En effet, il nous semble plus juste de comparer d'une part les heuristiques RF3_{CART} et RankBoost, procédant toutes deux à des *scissions* de l'ensemble des observations perpendiculairement aux dimensions de l'espace $\mathcal{X}$ et, d'autre part, les heuristiques RF1_{SVM}, RankSVM et RankRLS, fondées toutes trois sur la mise en oeuvre de Machines à Vecteurs Supports. Avant de présenter les résultats de cette étude empirique, nous décrivons brièvement les principes fondamentaux des approches concurrentes, dans le contexte spécifique de l'ordonnancement de données étiquetées de façon binaire.

L'objectif de la méthode RankBoost, proposée dans [Freund *et al.* 2003], est de produire une règle de score s minimisant le nombre de *paires discordantes*[1] en combinant linéairement une collection de règles de prédiction « *faibles* » apprises sur des *ré-échantillons* des données d'apprentissage. Cette approche itérative repose sur le ré-échantillonnage *adaptatif* des observations d'origine dans l'esprit de la procédure de *boosting* introduite

1. le nombre de paires d'observations $(X_i, X_j)_{i \neq j}$ de $\mathcal{X}^2$ telles que $s(X_i) \leq s(X_j)$ alors que $(Y_i, Y_j) = (+1, -1)$, ou inversement telles que $s(X_i) \geq s(X_j)$ alors que $(Y_i, Y_j) = (-1, +1)$



dans [Freund & Schapire 1999] et procède en 3 étapes fondamentales.

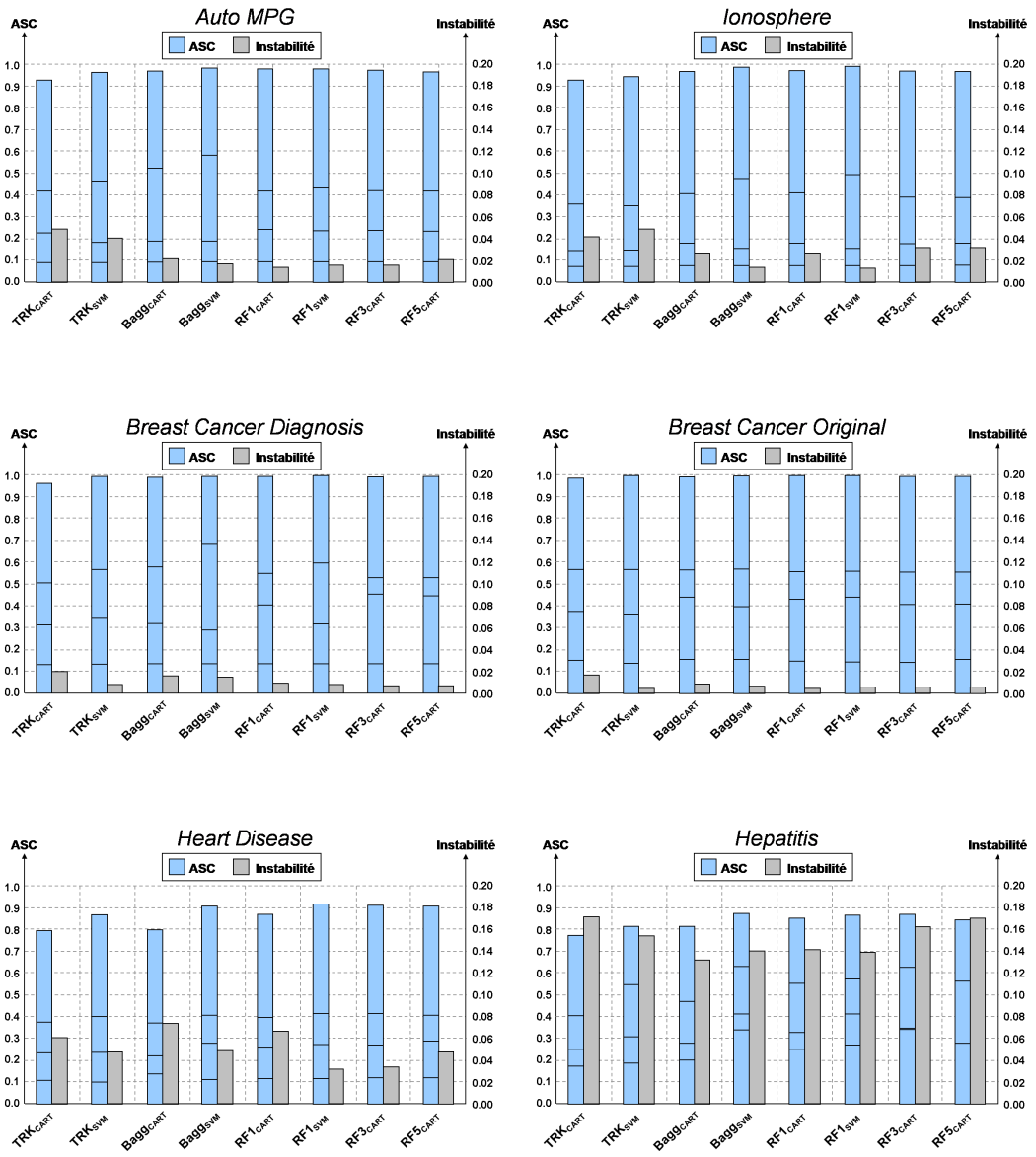


Figure 5.3 – Synthèse graphique de la comparaison de 8 heuristiques TreeRank : représentation de la performance $\overline{\mathrm{ASC}}^{(t)}$, de son écart-type $\hat{\sigma}^2$ et des ASC partielles pour les proportions $\{20\%, 10\%, 5\%\}$.

A chaque itération $t \in \{1, \cdots, T\}$,la première étape consiste à produire une règle de prédiction « faible » $\widetilde{s}_t$ à partir des données d'apprentissage. La seconde étape consiste à déterminer le poids a_t de cette fonction de score dans la règle d'ordonnancement finale $s \in \mathcal{S}$. Diverses approches sont envisagées dans [Freund *et al.* 2003] pour résoudre cette étape. Les auteurs proposent notamment de minimiser un majorant de l'*erreur d'ordonnancement* évaluée par le nombre de *paires discordantes*. Enfin, la troisième étape consiste à mettre à jour les pondérations des observations d'apprentissage. De la même façon que dans la procédure de *boosting*, cette pondération dépend naturellement du poids a_t et des

prédictions de la règle $\widetilde{s}_t$, le principe étant d'attribuer un poids plus important aux paires d'observations mal ordonnées par la règle de score courante afin de se concentrer sur ces observations à l'itération suivante. Finalement, la règle d'ordonnancement produite par la procédure RankBoost est définie comme la combinaison linéaire de la collection de règles de prédictions « faibles »apprises à chaque itération :

$$\forall\, x \in \mathcal{X},\ s(x) = \sum_{t=1}^{T} a_t \widetilde{s}_t(x).$$

Dans l'étude empirique présentée ci-après, nous avons mis en oeuvre la version de l'algorithme RankBoost disponible dans la boîte à outil *SVM and Kernel Methods Matlab Toolbox* implémentée par A. Rakotomamonjy [2]. Dans cette version, la règle d'ordonnancement finale $s \in \mathcal{S}$ est produite de sorte à maximiser l'ASC empirique et les règles de prédiction « faibles » sont des fonctions de score binaires de la forme :

$$\forall\, x \in \mathcal{X},\ \widetilde{s}_t(x) = \begin{cases} +1, & \text{si } x^{(j)} > \theta_i \\ -1, & \text{sinon} \end{cases},$$

où θ_i représente le $i^{\text{ème}}$ seuil de la $j^{\text{ème}}$ composante $x^{(j)}$ de l'observation x. Enfin, pour les 13 exemples considérés dans cette étude, chaque règle d'ordonnancement produite par l'algorithme RankBoost repose sur la combinaison de $T = 100$ règles « faibles » de cette forme.

Les deux autres méthodes que nous allons considérer dans cette étude, RankSvm et RankRLS, visent elles aussi à produire une règle de score minimisant le nombre de paires discordantes. Elles reposent toutes deux sur la mise en oeuvre d'heuristiques de type SVM pour minimiser un critère de la forme :

$$\frac{2}{n(n-1)} \sum_{i<j} d(Y_i - Y_j, f(X_i) - f(X_j)) + \lambda ||f||_{\mathbf{K}}, \tag{5.2}$$

où $(i,j) \in \{1, \cdots, n\}^2$, $n \geq 1$ étant la taille de l'échantillon $\mathcal{D}_n = \{(X_i, Y_i),\ 1 \leq i \leq n\}$ constitué de copies *i.i.d.* du couple $(X,Y) \in \mathcal{X} \times \{-1,+1\}$; pour tout $x \in \mathcal{X}$, $f(x) = \sum_{i=1}^{n} a_i \mathbf{K}(x, X_i)$, $\lambda \in \mathbb{R}$ est un paramètre de régularisation, $||f||_{\mathbf{K}}$ est la norme RKHS (pour *Reproducing Kernel Hilbert Space*) de f associée au noyau $\mathbf{K}$ et enfin $d(.,.)$ est une fonction de coût fixée.

Dans le cas de la méthode RankSvm, le critère (5.2) est optimisé au moyen d'une procédure SVM fondée sur la fonction de coût *classique* « *hinge loss* » donnée par

$$\forall\, (u,v) \in \{-2, 0, +2\} \times \mathbb{R},\ d(u,v) = (1 - uv)_+,$$

où la notation $(.)_+$ désigne la *partie positive*. Cette méthode s'inspire de l'approche proposée dans [Herbrich *et al.* 2000] dans le contexte de la régression ordinale. Elle a été introduite dans [Joachims 2002b], dans le contexte spécifique de la recherche de document, et dans [Rakotomamonjy 2004]. Ces deux contributions proposent respectivement

2. http ://asi.insa-rouen.fr/enseignants/ arakotom/toolbox/index.html

d'optimiser le τ de Kendall et l'ASC en résolvant le problème suivant :

$$\min \frac{1}{2}||\mathbf{w}||^2 + c\sum_{i=1}^{n_+}\sum_{j=1}^{n_-}\xi_{i,j}$$
$$\text{sous les contraintes } f(x_i) - f(x_j) \geq 1 - \xi_{i,j}, \forall\, 1 \leq i \leq n_+ \text{ et } 1 \leq j \leq n_-,$$
$$\text{et } \xi_{i,j} \geq 0, \forall\, 1 \leq i \leq n_+ \text{ et } 1 \leq j \leq n_-,$$

où n_+ représente le nombre d'observations *positives* dans l'échantillon et $n_- = n - n_+$, ce qui revient finalement à résoudre un problème de classification binaire sur les paires d'observations. Deux implémentations de cette méthode sont disponibles : l'algorithme ROC SVM de la boîte à outil *SVM and Kernel Methods Matlab Toolbox* implémentée par A. Rakotomamonjy[2] et l'algorithme SVM^{rank} implémenté par T. Joachims [3]. C'est cette dernière version que nous avons mise en oeuvre dans l'étude empirique présentée ci-après. Pour chaque exemple, nous avons paramétré l'algorithme de manière cohérente avec l'heuristique RF1_{SVM} : nous avons notamment choisi le même noyau (gaussien ou linéaire) et le même coefficient c.

Comme nous l'avons souligné, la méthode RankRLS, proposée dans [Pahikkala *et al.* 2007], repose elle-aussi sur la mise en oeuvre d'une heuristique de type SVM, mais dans cette approche, la fonction de coût « *hinge loss* » est remplacée par le critère des moindres carrés suivant :

$$\forall\, (u, v) \in \{-2, 0, +2\} \times \mathbb{R},\ d(u, v) = (u - v)^2.$$

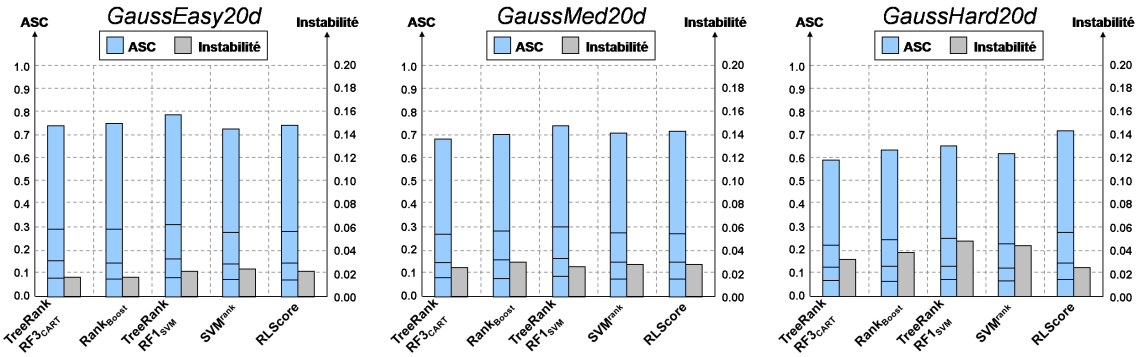
De même que précédemment, cette approche revient à résoudre un problème de classification binaire sur les paires d'observations, mais repose sur la minimisation d'un estimateur des moindres carrés régularisés de l'*erreur d'ordonnancement*. Une implémentation de cette méthode (RLScore) est proposée par T. Pahikkala [4]. De même que pour la méthode RankSVM, nous avons paramétré cet algorithme en choisissant pour chaque exemple, le noyau $\mathbf{K}$ utilisé pour l'heuristique RF1_{SVM}. Pour le choix du paramètre de régularisation λ, nous avons testé plusieurs valeurs sur l'intervalle $[-10, +10]$ et nous avons conservé celle produisant le meilleur résultat.

De même que dans l'étude précédente, nous comparons les performances de ces heuristiques sur les 13 jeux de données déjà introduits. Les résultats de cette étude sont résumés dans le Tableau A.3 de l'Annexe A qui contient, pour chaque exemple et chaque heuristique, la valeur de l'ASC moyenne $\overline{\text{ASC}}^{(t)}$ accompagnée, entre parenthèses, de son écart-type estimé sur les V itérations de validation et dans le Tableau A.4 de l'Annexe A contenant les ASC tronquées moyennes pour les trois proportions $u \in \{0.2, 0.1, 0.05\}$ (dans cet ordre), accompagnées de leurs écart-types entre parenthèses. Enfin, la Figure 5.4 fournit une représentation graphique de l'ensemble de ces résultats. De même que précédemment, chaque histogramme est constitué des *couples* $(\overline{asc}^{(t)}, \hat{\sigma}^2)$ pour chaque heuristique, les ASC partielles étant représentées par les *scissions* horizontales portées par la *barre* représentant la valeur de l'ASC empirique.

Commençons par étudier les résultats du premier tableau, relatifs à la *performance*, mesurée en termes d'ASC totale, et à l'*instabilité*, évaluée par l'écart-type $\hat{\sigma}^2$ de l'ASC.

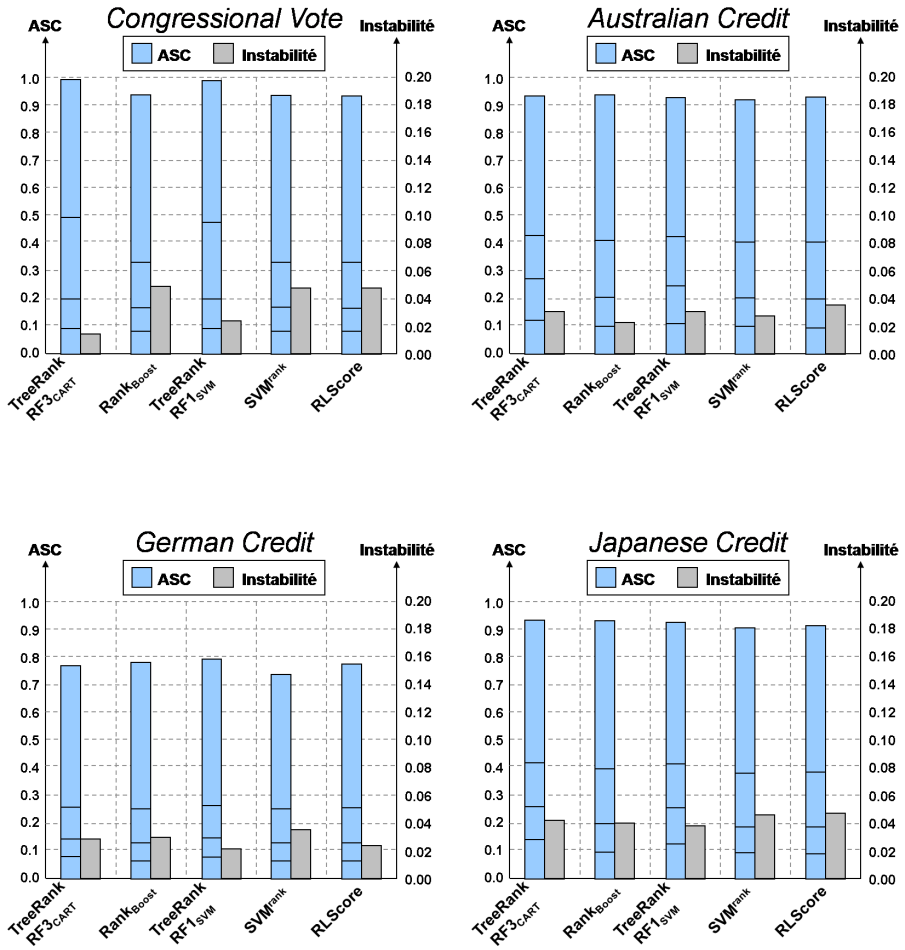
3. http ://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html
4. http ://staff.cs.utu.fi.∼aatapa/software/RLScore/

On peut déjà remarquer que, même si l'on peut observer quelques différences sur les 3 mélanges de gaussiennes, les 5 heuristiques présentent des résultats assez similaires, que ce soit en termes de performance ou d'instabilité. Si l'on compare les performances des 3 heuristiques « concurrentes » entre elles, on constate tout de même une légère supériorité de l'heuristique RankBoost, qui présente les meilleurs résultats sur les jeux de données benchmark, alors que sur les jeux de données simulés, c'est l'algorithme RLScore qui s'avère être le plus efficace.



Si l'on compare maintenant les deux heuristiques procédant à des *scissions* perpendiculaires aux axes de l'espace $\mathcal{X}$, on constate que si l'heuristique RankBoost est plus performante sur les trois mélanges de gaussiennes, la version $\mathrm{RF3_{CART}}$ de la méthode TreeRank présente des performances similaires, voire légèrement supérieures sur les données benchmark, exception faite de l'exemple *German Credit*. De même, l'étude des approches fondées sur Machines à Vecteurs Supports montre que l'heuristique $\mathrm{RF1_{SVM}}$ est légèrement plus performante que ses deux concurrentes sur l'ensemble des 13 jeux de données réelles. En résumé, il ressort de cette étude que RankBoost et $\mathrm{RF1_{SVM}}$ sont les deux heuristiques les plus performantes. De plus, en observant les graphiques de la Figure 5.4, on voit que les performances de la version *ré-échantillonnée* et *randomisée* de la méthode TreeRank sont légèrement supérieures à celles de l'heuristique RankBoost.

Considérons maintenant la deuxième partie de cette étude et la comparaison des ASC tronquées. Les résultats consignés dans le Tableau A.4 montrent clairement la supériorité de la méthode TreeRank en ce qui concerne l'ordonnancement des *meilleures* observations. En effet, quelle que soit la version considérée ($\mathrm{RF3_{CART}}$ ou $\mathrm{RF1_{SVM}}$), la méthode de scoring proposée dans ce manuscrit présente de bien meilleures performances, en termes d'ASC tronquées, que ses concurrentes. Ce résultat n'est toutefois pas réellement surprenant et résulte du fait que, contrairement aux méthodes RankBoost, RankSVM et RankRLS, qui reposent sur l'optimisation de l'ASC globale (ou d'une approximation de ce critère), la méthode TreeRank procède à l'optimisation récursive de l'ASC *locale* et permet ainsi de produire des règles de score optimales non seulement au sens de l'ASC mais plus généralement, au sens de la courbe COR. Soulignons tout de même les rares exceptions : l'heuristique RankBoost semble plus performante que la version $\mathrm{RF3_{CART}}$, pour ordonner les 20% et 10% *meilleures* observations sur les trois mélanges de gaussiennes, et notons que l'écart entre les performances des heuristiques a tendance à se réduire quand la proportion des meilleures observations à ordonner décroît.



5.1.5 Conclusion et perspectives

L'objectif de cette première partie était de comparer diverses versions de l'heuristique TreeRank entre elles et avec quelques méthodes de scoring concurrentes, proposées dans la littérature. La première étude empirique présentée nous a permis de mettre en évidence l'intérêt des procédures de *ré-échantillonnage* et de *randomisation*, permettant de *stabiliser* et d'améliorer significativement les performances de la méthode TreeRank, en particulier lorsque celle-ci repose sur l'implémentation récursive d'une version pondérée de l'algorithme de classification CART. Par ailleurs, en comparant les heuristiques $\mathrm{RF3}_{\mathrm{CART}}$ et $\mathrm{RF1}_{\mathrm{SVM}}$ avec les méthodes RankBoost, RankSVM et RankRLS, nous avons pu constater les bonnes performances de la méthode TreeRank, qui s'avère notamment être largement supérieure à ses concurrentes en ce qui concerne l'ordonnancement des *meilleures* observations d'un échantillon.

Nous envisageons de mener par la suite de nouvelles expériences, afin d'approfondir cette exploration empirique des performances de la méthode TreeRank, que ce soit en considérant de nouveaux critères de performances ou en se comparant à d'autres méthodes de scoring, sur des jeux de données peut-être mieux *adaptés* à la problématique d'ordonnancement binaire. Il serait par exemple intéressant de regarder l'*instabilité* des diverses méthodes, non pas au sens de l'écart-type du critère ASC mais au sens de la distance entre ordonnancements, comme nous l'avons proposé dans le chapitre précédent. Nous envisageons aussi d'évaluer les performances au sens de critères classiquement utilisés pour la comparaison d'heuristiques d'ordonnancement, comme par exemple les critères MAP (pour *Mean Average Precision*) et NDCG (pour *Normalized Discounted Cumulative Gain* (cf Partie 1.2.2.3 du Chapitre 1).

Cependant, notre objectif est surtout d'élargir le champ de cette étude expérimentale en considérant une plus vaste collection de méthodes concurrentes. En particulier, il serait judicieux de comparer l'algorithme TreeRank à la méthode d'ordonnancement proposée dans [Rudin 2006], qui permet notamment de se focaliser sur les *meilleures* observations de l'échantillon d'apprentissage. Il serait aussi intéressant de pouvoir se comparer avec d'autres méthodes fondées sur l'optimisation de critères empiriques évaluant l'*erreur d'ordonnancement*, comme par exemple certains algorithmes développés dans le cadre de partenariats avec l'entreprise Microsoft : RankNet ([Burges *et al.* 2005]), FRank ([Tsai *et al.* 2007]) ou encore LambdaRank ([Burges *et al.* 2006]). Enfin, nous pourrions également nous comparer à des méthodes plus orientées vers l'estimation statistique de la fonction de régression ou d'une transformée strictement croissante de cette dernière comme par exemple la *régression logistique à noyau* KLR (pour *Kernel Logistic Regression*) (voir par exemple [Hastie & Tibshirani 1990] et [Hastie *et al.* 2001]).

5.2 Objectivation de la prestation *brio*

Comme nous l'avons déjà souligné, la problématique d'ordonnancement binaire intervient de manière récurrente dans des domaines comme la finance ou la médecine par exemple, mais c'est dans un contexte tout à fait différent que ces travaux de thèse ont été lancés, celui de l'*objectivation des prestations* d'un véhicule automobile. Les Parties 5.2.1 et 5.2.2 suivantes sont consacrées à la description de cette problématique et de la méthodologie actuellement mise en oeuvre par le constructeur automobile Renault pour la résoudre. Puis, nous décrivons, dans la Partie 5.2.3, la prestation *brio* sur laquelle nous avons travaillé de manière plus spécifique, pour produire un *indice du brio*. Enfin, nous détaillons dans la dernière partie les résultats obtenus sur ces données d'objectivation via l'heuristique TreeRank, que nous comparons à une approche *plug-in* fondée sur la résolution d'un problème de type LASSO, proposée dans [Germain 2009] pour l'objectivation de l'*agrément de conduite*.

5.2.1 L'objectivation des prestations

Jusqu'à il y a encore quelques années, la conception d'un véhicule visait essentiellement à satisfaire diverses contraintes physiques et normes imposées et ce sans se préoccuper, dans un premier temps du moins, de la notion d'*agrément* de conduite ou de *confort*.

Celle-ci n'intervenait que plus tard dans le processus et, en règle générale, le nombre de leviers restant pour améliorer le produit était considérablement restreint. Cependant, les nombreux progrès techniques et le développement du marché automobile ont aiguisé les attentes des consommateurs, devenus de plus en plus exigeants vis-à-vis de la qualité du produit. Aujourd'hui, l'objectif des constructeurs, et de Renault en particulier, est d'intégrer l'agrément dès la phase de conception, afin de disposer de tous les leviers nécessaires pour concevoir des véhicules à la hauteur des exigences de la clientèle. Chez Renault, cette volonté s'est traduite par la mise en place d'un processus de déploiement des prestations et par le lancement de plusieurs thèses, qui ont contribué à la mise en place d'une *méthodologie d'objectivation*.

Un véhicule automobile offre à ses utilisateurs un grand nombre de *prestations*, qui couvrent des périmètres très divers allant de la sécurité au confort, en passant par la performance ou encore l'ergonomie, par exemple. Certaines de ces prestations sont aisément quantifiables, voire même normalisées, comme l'émission de polluants contrainte à diverses normes européennes. Cependant, dans la plupart des cas, la notion de prestation relève du domaine du subjectif : les prestations relatives au confort ou au plaisir de conduite, par exemple, sont liées au ressenti de l'utilisateur. Ce sont des *items* vis-à-vis desquels les clients vont pouvoir exprimer leurs sensations. La notion d'*objectivation*, quant à elle, renvoie au fait d'expliquer des données subjectives à partir de données objectives. *Objectiver une prestation* c'est donc identifier les critères physiques qui influencent la perception du client. C'est une étape cruciale qui s'inscrit dans le processus de déploiement des prestations mis en place chez Renault.

Ce processus doit permettre de caractériser et de quantifier les prestations, afin de tenir compte des attentes clients lors de la phase de conception des véhicules. Il est composé de deux phases, elles-mêmes scindées en plusieurs étapes. La phase de *conception* a pour but de définir des *cahiers des charges* (*CdC*), qui définissent les niveaux de prestation à atteindre et les caractéristiques techniques des différents organes et pièces d'un véhicule. Bien évidemment, ceux-ci doivent tenir compte des attentes clients relatives aux prestations considérées. La deuxième phase consiste ensuite à vérifier que les solutions techniques proposées permettent d'atteindre les engagements définis dans les cahiers des charges et que le client est satisfait du produit. Ces deux phases constituent respectivement les branches *descendante* et *ascendante* de ce que l'on appelle le *« V » de la prestation*, schématisé sur la Figure 5.5.

Pour définir un cahier des charges conforme aux attentes des utilisateurs, il faut tout d'abord les identifier : c'est la première étape de la branche descendante du « V », qui repose essentiellement sur l'analyse d'enquêtes ciblées, réalisées auprès de la clientèle (établie et potentielle), mais aussi sur l'analyse des informations disponibles par le biais de différents médias : presse automobile, associations de consommateurs, organismes spécialisés, etc. En se basant sur ces informations, on définit un *cahier des charges produit*, qui établit les caractéristiques globales du véhicule et fixe le niveau de prestation à atteindre, en tenant compte à la fois des attentes clients, mais aussi de la réalité du marché et de l'identité de la marque. Par exemple, en ce qui concerne l'*émission de polluants*, la marque Dacia du groupe Renault devra satisfaire les normes européennes, tout en proposant les véhicules les moins chers du marché.

Les étapes suivantes de la phase descendante du « V » consistent alors à proposer des

solutions techniques au niveau du véhicule, des organes et des pièces, pour atteindre le niveau de prestation fixé. Pour cela, il faut tout d'abord identifier quels sont les critères physiques qui influencent le ressenti de l'utilisateur : c'est l'étape d'*objectivation*. Celle-ci va permettre de lister les différents *leviers physiques*, dont on disposera pour atteindre la cible fixée par le cahier des charges produit. Il faut ensuite décliner ces leviers au niveau des organes du véhicule et des pièces qui les composent et définir des solutions techniques adaptées. Ces deux dernières étapes relèvent de la compétence des ingénieurs dits *métiers*, experts de la prestation considérée.

La phase ascendante du « V », consiste à vérifier si le niveau de prestation finalement atteint est bien conforme au cahier des charges initialement défini et aux attentes des utilisateurs. Cette validation s'effectue d'abord au niveau des pièces, puis au niveau des organes et enfin au niveau du véhicule, par des ingénieurs experts de la prestation. On retourne ensuite vers le client pour la dernière étape de validation de la conception, afin de vérifier que le niveau de prestation proposé convient aux utilisateurs.

5.2.2 Méthodologie d'objectivation des prestations

L'objectivation d'une prestation repose sur l'analyse de données *objectives* et *subjectives* recueillies pendant une campagne d'essais. Les données objectives sont des grandeurs physiques mesurées -ou calculées à partir de mesures- sur un véhicule pendant la réalisation d'une *manoeuvre* caractéristique de la prestation considérée (changement de rapport, démarrage du véhicule, manoeuvre de dépassement, etc.). Ces données sont de nature différente selon la prestation considérée. Dans le cadre d'une étude acoustique par exemple, on mesurera les niveaux de décibels ou les vibrations générées à l'aide de capteurs disposés dans l'habitacle. Pour une étude du confort de siège, on étudiera des cartographies de pression, relevées par des capteurs installés dans un siège test. Les grandeurs physiques peuvent donc aussi bien être des données réelles que des signaux temporels ou des images et impliquent ainsi la mise en oeuvre de processus de mesures plus ou moins complexes et coûteux. Quant aux données subjectives, ce sont des notes -cotations- traduisant l'évaluation subjective de la prestation du véhicule pendant la manoeuvre, par l'utilisateur, *i.e.* son ressenti.

5.2.2.1 Recueil des données d'objectivation

Idéalement, l'objectivation d'une prestation repose sur l'analyse de données recueillies pendant une campagne d'essais en clientèle, *i.e.* une campagne pendant laquelle un panel de véhicules est testé par une population de clients dits *naïfs*, *i.e.* non-experts de la prestation. L'organisation de telles campagnes nécessite cependant de définir une procédure précise et adaptée. Dans [Ansaldi 2002], une *méthodologie* en 3 étapes est proposée pour réaliser ce genre d'études d'objectivation. La première consiste à définir la prestation que l'on souhaite objectiver et à identifier des manoeuvres caractéristiques de celle-ci ainsi que les mesures à réaliser sur le véhicule. Ces choix relèvent de l'expertise des ingénieurs métiers. La deuxième étape est marquée par la réalisation d'une campagne d'essais : les manoeuvres identifiées sont réalisées sur des véhicules instrumentés par un panel de clients testeurs. Pendant cette campagne, il faut recueillir, pour chaque essai, les mesures physiques délivrées par les capteurs équipant le véhicule, ainsi que le ressenti subjectif des testeurs. Ces données sont ensuite pré-traitées, généralement par les ingénieurs métiers

experts de la prestation, qui en extraient les critères physiques pertinents et construisent la base de donnée pour la modélisation. Enfin, la dernière étape consiste à objectiver la prestation à proprement parler, *i.e.* à modéliser le ressenti subjectif à partir des mesures objectives.

Deux des principales difficultés de cette procédure résident dans la réalisation des essais et dans la cotation subjective de la prestation. Pour la réalisation de la campagne d'essais, il est suggéré dans [Ansaldi 2002] de définir des manoeuvres suffisamment simples et reproductibles, pour qu'elles puissent être aisément réalisées par tous les clients-testeurs et que les mesures soient robustes. La définition du mode de recueil de la cotation subjective est un autre point crucial : il est important de choisir une cotation adaptée au niveau d'expertise de l'essayeur. Lorsque les véhicules sont notés par des clients « naïfs », au sens où ils ne sont pas « experts » de la prestation, il est important de considérer une notation suffisamment simple et robuste. Dans [Ansaldi 2002], diverses formes de cotations sont considérées :

(i) **Cotation absolue** des véhicules du panel sur une échelle de valeurs continues -$[0, 10]$ par exemple- ou graphiquement par positionnement sur un segment.

(ii) **Cotation relative** des véhicules du panel sur une échelle de valeurs continues -$[0, 10]$ par exemple- ou graphiquement par positionnement sur un segment.

(iii) **Comparaisons par paires** des véhicules du panel.

(iv) **Listes de préférences** sur la totalité ou une partie des véhicules du panel.

(v) **Cotation discrète** sur un nombre fini et restreint de classes -bon *vs* mauvais ou excellent *vs* bon *vs* médiocre *vs* mauvais par exemple-.

Chacune de ces cotations présente un certain nombre d'avantages et d'inconvénients. Par exemple, il est préférable de travailler sur des cotations absolues -pour se laisser la possibilité, par exemple, d'ajouter des véhicules au panel ultérieurement-, mais ce type de cotation est difficile à obtenir d'un client naïf. Cependant, définir une cotation relative n'est pas plus aisé, surtout si le client doit noter plus de deux véhicules. En effet, pour pouvoir coter un ensemble de véhicules, le client devra tous les essayer avant de les noter. Si le nombre de véhicules à noter est important, le client aura probablement des difficultés à se remémorer les différents essais. Par ailleurs, coter sur une échelle de valeurs continues comme $[0, 10]$ par exemple, est un exercice difficile pour des testeurs non-experts de la prestation. On risque notamment de recueillir des données particulièrement bruitées et peu robustes d'un testeur à un autre. D'après [Ansaldi 2002], les modes d'évaluation les plus adaptés à une campagne d'essais en clientèle sont l'établissement de listes de préférences, les comparaisons par paires et la cotation discrète avec un petit nombre de classes. Pour les autres, un étalonnage s'avère souvent nécessaire pour corriger le biais induit par les essayeurs.

Malheureusement, l'organisation de telles campagnes d'essais est compliquée d'un point de vue logistique (disponibilité des clients et des véhicules), et particulièrement coûteuse (rémunération éventuelle, assurances, matériel de mesure, formation, etc.). Aussi, en règle générale, les essais sont réalisés en interne par des experts de la prestation, que l'on peut

considérer comme des clients particulièrement exigeants. Il paraît donc raisonnable de penser que si l'on parvient à les satisfaire, on peut espérer combler les attentes d'une large majorité de la clientèle.

Quand les campagnes d'essais sont réalisées par des ingénieurs experts de la prestation, deux formes de cotations sont privilégiées selon les prestations considérées : une cotation binaire de type « bon *vs* mauvais » et une cotation continue sur une échelle de 0 à 10. La plupart du temps, les experts préfèrent utiliser la notation continue sur $[0, 10]$, mais, du point de vue de l'objectivation, celle-ci présente des inconvénients majeurs. Premièrement, elle contient un *seuil*, qui correspond à la note minimale pour la validation d'une prestation en projet. Or, ce seuil varie dans le temps et avec l'augmentation du niveau d'exigence de la clientèle et des normes diverses, ce qui rend difficile la comparaison de véhicules sur une période de temps trop longue. De plus, les experts ont tendance à noter sur un intervalle beaucoup plus petit que l'échelle $[0, 10]$ initiale. Selon les études, les véhicules sont généralement notés entre 5.5 et 7.75, par pas de 0.25. Finalement, sur la plupart des échantillons, on dispose de cotations discrètes ordonnées avec un petit nombre de classes, variable selon les experts. Aussi, se ramène-t-on souvent à une cotation binaire, pour plus de robustesse et pour s'affranchir du seuil présent dans la cotation continue.

5.2.2.2 Modélisation du ressenti subjectif

La méthode actuellement mise en oeuvre chez Renault pour modéliser le ressenti subjectif à partir des critères physiques mesurés a été proposée dans [Germain 2007b] (voir aussi [Germain 2007a] et [Germain 2009]). Celle-ci consiste à établir un compromis entre l'*erreur de modélisation* et la *complexité* du modèle, en résolvant un problème de type LASSO (voir [Tibshirani 1996] et [Hastie *et al.* 2001] par exemple) de la forme

$$\forall\, \lambda \in [0, 1]\,,\ \beta(\lambda) = \underset{\beta \in \mathbb{R}^q}{\arg\min}\{-\log \mathcal{L}_n(\beta) + \lambda ||\beta||_1\}, \tag{5.3}$$

où β est le paramètre du modèle de régression $Y = \beta X + \epsilon$, où ϵ représente un bruit blanc gaussien et $X \in \mathcal{X} = \mathbb{R}^q$, Y pouvant être continu ou binaire. On note $||\beta||_1$ la norme $\mathcal{L}_1$ de ce paramètre, $\mathcal{L}_n(\beta)$ le rapport de vraisemblance du modèle et λ est le coefficient de régularisation du problème LASSO.

Dans [Germain 2007b], ce problème est considéré dans un contexte de classification binaire, où $Y \in \{-1, +1\}$, et revient à estimer le paramètre $\beta \in \mathbb{R}^q$ pour un modèle *logit*. Dans cette contribution, l'auteur propose une méthode en deux étapes qui consiste, dans un premier temps, à *hiérarchiser* les variables en construisant une suite de modèles *imbriqués*[5] solutions du problème (5.3) pour différentes valeurs de λ puis, dans un deuxième temps, à sélectionner le *meilleur* modèle de cette suite, au sens d'un critère que nous allons préciser.

Pour *hiérarchiser* l'ensemble des variables, l'auteur propose de résoudre le problème (5.3) pour tout $\lambda \in [0, 1]$, en construisant un *chemin de régularisation*. Ce chemin, que l'on note $\lambda \to \beta(\lambda)$, peut être approché de manière efficace en mettant en oeuvre l'heuristique proposée dans [Hastie & Park 2007], inspirée du célèbre algorithme LARS ([Efron *et al.* 2004]). Si celui-ci dépend naturellement du coefficient λ, il semble judicieux de le représenter

5. suite de modèles de tailles croissantes, obtenus en ajoutant à chaque itération une nouvelle variable dans le modèle précédent

comme une fonction de $||\beta||_1$. En effet, la norme $\mathcal{L}_1$ du paramètre β permettant d'évaluer la complexité du modèle, cette représentation fournit directement la suite croissante des modèles imbriqués solutions de (5.3) (voir la Figure 5.6 ci-dessous).

Le chemin de régularisation étant défini, la seconde étape de la procédure consiste à sélectionner le *meilleur* modèle. Dans [Germain 2007b], cette étape est résolue par la minimisation du critère BIC (pour *Bayesian Information Criterion*, [Schwarz 1978]). Notons enfin qu'une fois le modèle sélectionné, il convient de réajuster l'estimation du coefficient de régression β en résolvant le problème de régression logistique non pénalisé.

Comme cela est précisé dans [Germain 2009], cette procédure présente toutefois quelques limites. Tout d'abord, s'agissant d'une approche de type *plug-in*, cette méthode se trouve confrontée au problème de la dimension de l'espace d'entrée $\mathcal{X}$. De plus, le critère BIC peut avoir tendance à sélectionner des modèles de petite taille, induisant parfois des problèmes de sous-apprentissage. Aussi, il peut être préférable de sélectionner le meilleur modèle par une procédure de validation croisée. Enfin, la difficulté majeure réside certainement dans l'interprétation du chemin de régularisation, qui peut parfois s'avérer délicate. En effet, la méthode proposée dans [Germain 2009] repose sur deux hypothèses fondamentales. Tout d'abord, on suppose que plus une variable est *significative*, plus elle apparaît *tôt* dans le chemin. On suppose de plus que la hiérarchie entre les variables reste inchangée tout au long du chemin, au sens où la valeur du coefficient associé à une variable entrant dans le modèle est à la fois supérieure à celle des variables qui entreront après elle et inférieure à celle des variables déjà sélectionnées. Cependant, cette dernière hypothèse peut être mise en défaut du fait principalement de la présence de colinéarités entre les prédicteurs et de l'augmentation de la dimension du problème. De ce fait, il est nécessaire, dans certaines situations, de revenir vers les experts de la prestation afin de confirmer les modèles sélectionnés.

5.2.3 La prestation *brio*

Ces travaux de thèse ont été plus particulièrement appliqués à l'objectivation de la prestation *brio*. On peut définir le *brio* d'un véhicule comme la *sensation à l'accélération* qu'il procure au conducteur. En d'autres termes, il s'agit de la perception que peut avoir un conducteur du comportement de son véhicule suite à une sollicitation d'accélération.

Pour caractériser cette prestation, les experts effectuent des manoeuvres d'accélération dites de *reprise*, consistant à accélérer soudainement et de façon significative. L'accélération à la sortie d'un rond-point ou lors du dépassement d'un autre véhicule sont deux situations quotidiennes correspondant à ce type de manoeuvre. Chez Renault, le brio est caractérisé par les trois manoeuvres de reprise suivantes :

- *reprise à enfoncement pédale de* 100% *après phase de régime stabilisé* : pour un rapport de boîte de vitesse engagé, on stabilise le régime moteur à une valeur fixée puis on enfonce la pédale d'accélérateur à 100%,
- *reprise à enfoncement pédale de* 50% *après phase de régime stabilisé* : pour un rapport de boîte de vitesse engagé, on stabilise le régime moteur à une valeur fixée puis on enfonce la pédale d'accélérateur à 50%,
- *reprise à enfoncement pédale de* 100% *après phase de décélération* : pour un rapport

de boîte de vitesse engagé, on laisse chuter le régime moteur jusqu'à atteindre une valeur fixée puis on enfonce la pédale d'accélérateur à 100%.

Au cours de ces manoeuvres, les experts procèdent notamment à la mesure de l'accélération du véhicule et à l'évaluation de son brio. Une fois les manoeuvres terminées, les essais sont dépouillés et les experts de la prestation extraient un certain nombre de *critères physiques* (à valeurs réelles) à partir des courbes d'accélération. Il s'agit principalement de critères de *délais*, relatifs au temps de réponse du véhicule, ou de *pentes* de la courbe d'accélération ou encore de certaines valeurs d'accélération spécifiques, comme l'accélération maximale par exemple. Nous avons représenté sur la Figure 5.7 ci-dessous la courbe (temporelle) d'accélération d'un véhicule mesurée au cours d'une manoeuvre de reprise à enfoncement pédale maximal après une phase de régime stabilisé.

D'après la description de la problématique d'objectivation que nous avons donnée précédemment, le but principal de l'objectivation de la prestation brio est d'identifier les critères physiques ayant un impact significatif sur le ressenti subjectif des clients. Cependant, l'étude à laquelle nous avons participé dans le cadre de ces travaux de thèse avait un second objectif : la construction d'un *indice du brio*. En effet, pour certaines prestations, il est intéressant de définir un indice afin d'évaluer le niveau de prestation atteint et de situer un véhicule par rapport à ses principaux concurrents. Dans ce cas précis, l'objectivation d'une prestation peut alors être formulée comme un problème d'ordonnancement (éventuellement binaire selon la nature de la cotation subjective formulée). La Partie 5.2.4 suivante est consacrée à la description des données de l'étude d'*indice brio* et des résultats obtenus via la méthode TreeRank proposée dans ce manuscrit, dont nous comparons les performances avec l'approche introduite dans [Germain 2009], actuellement mise en oeuvre chez Renault pour résoudre les problématiques d'objectivation des prestations.

5.2.4 Construction d'un indice du *brio*

Les données disponibles dans le cadre de l'étude d'*indice brio* proviennent d'une vaste campagne d'essais, menés par des experts de la prestation sur un panel constitué de plus de 50 véhicules de *motorisations* (*diesel*, *essence turbo*, *essence atmosphérique*) et de *segments* (petit, moyen, gros véhicule) différents. Malheureusement, tous ces essais ne sont pas nécessairement comparables. En effet, la différence de motorisation implique des profils d'accélération différents et l'on s'attend à ce que les critères physiques *significativement influents* sur le ressenti subjectif du brio ne soient pas les mêmes d'une motorisation à l'autre. Aussi, nous nous sommes restreints à la construction d'un indice du brio pour les véhicules « *essence atmosphérique* » (*i.e.* non-équipés de turbos).

Pour ce type de motorisation, 19 véhicules ont été testés sur 6 manoeuvres de reprise à enfoncement pédale de 100%, effectuées pour un rapport de boîte de vitesse fixé, après stabilisation du régime moteur à diverses valeurs. Pour chaque essai, les experts de la prestation nous ont fourni une collection de 37 critères physiques réels extraits de la courbe d'accélération du véhicule et une cotation binaire, représentant leur ressenti subjectif du brio. Nous avons toutefois dû supprimer 16 essais, pour lesquels les courbes d'accélération n'avaient pas été correctement enregistrées. Finalement, nous disposons d'une base de donnée constituée de $n = 98$ observations, auxquelles est associée une étiquette binaire, et de $p = 37$ prédicteurs.

Comme nous l'avons déjà souligné, la question spécifique de la construction d'un indice à partir de données étiquetées de façon binaire revient à résoudre un problème d'ordonnancement binaire. Nous proposons donc de modéliser la prestation brio en mettant en oeuvre la méthode TreeRank introduite dans ce manuscrit. De plus, nous proposons de comparer ses performances avec la méthode d'objectivation proposée dans [Germain 2009]. En effet, on peut voir cette approche de type *plug-in* comme une méthode de *scoring*, dont l'objectif est finalement d'estimer la probabilité à posteriori η.

Afin d'évaluer de la façon la plus pertinente possible les performances de ces deux méthodes, nous avons scindé en deux l'ensemble des observations, en allouant 85% des données à l'échantillon d'apprentissage $\mathcal{D}^{(a)}$ et en conservant les 15% restantes pour visualiser les courbes COR *test* obtenues au moyen de ces deux approches. Nous avons aussi comparé diverses versions de l'heuristique TreeRank sur ces données. Cependant, l'interprétabilité des modèles obtenus étant cruciale dans cette étude, dont un des objectifs reste l'identification des critères physiques *significativement explicatifs* du ressenti subjectif, nous nous sommes restreints à des heuristiques fondées sur l'implémentation récursive d'une version pondérée de l'algorithme CART. De plus, étant donné l'*instabilité* de l'heuristique $\mathrm{TRK}_{\mathrm{CART}}$ mise en évidence dans le chapitre précédent, nous avons choisi de mettre en oeuvre des procédures de *ré-échantillonnage* et de *randomisation* et de construire des *forêts* constituées de $B = 30$ arbres d'ordonnancement. En conservant les mêmes notations que dans la partie précédente, nous comparons les heuristiques $\mathrm{Bagg}_{\mathrm{CART}}$, $\mathrm{RF1}_{\mathrm{CART}}$, $\mathrm{RF3}_{\mathrm{CART}}$ et $\mathrm{RF5}_{\mathrm{CART}}$ à l'approche de type LASSO, que nous noterons ChReg.

Une portion du chemin de régularisation obtenu via la méthode ChReg, représentant les 10 premières variables entrant dans le modèle, a été reproduite sur le premier graphique de la Figure 5.8 ci-dessous, les valeurs du critère BIC associées étant tracées sur le second graphique de la figure. Comme nous l'avons déjà souligné, ce critère tend à sélectionner des modèles de petite taille. Dans cet exemple précis, le modèle sélectionné ne contient qu'un unique prédicteur : la $6^{\text{ème}}$ composante $X^{(6)}$. Afin de confirmer ce résultat, nous avons mis en oeuvre une procédure de validation croisée par blocs (avec $V = 5$) et une procédure de *leave-one-out*, qui ont toutes deux conduit à la sélection de ce même modèle.

En réalité ce prédicteur est connu des experts qui le prennent effectivement en compte pour objectiver la prestation brio. Cependant, l'observation des courbes COR de *test*, tracées sur la Figure 5.9 suivante, tend à montrer que ce seul prédicteur ne suffit pas à expliquer le ressenti subjectif des testeurs. En effet, il apparaît clairement sur ce graphe que la procédure TreeRank est plus performante que l'approche LASSO. Quelle que soit l'heuristique mise en oeuvre, la méthode TreeRank induit un gain, en termes d'ASC, d'au minimum 24%.

Par ailleurs, on peut constater que sur cet exemple, la procédure de *randomisation* ne permet pas d'améliorer les performances de la méthode de scoring proposée dans ce manuscrit. En effet, la meilleure performance est atteinte par l'heuristique $\mathrm{Bagg}_{\mathrm{CART}}$, qui présente une ASC de test de 43% supérieure à celle de la procédure ChReg, alors que les heuristiques *randomisées* induisent un gain en ASC par rapport à cette méthode compris entre 24% et 33% « seulement ».

Finalement, nous avons proposé aux experts un *indice du brio*, constant par morceaux, défini par l'agrégation de 30 arbres d'ordonnancements. L'heuristique $\mathrm{Bagg}_{\mathrm{CART}}$ étant fon-

dée sur la mise en oeuvre de l'algorithme de classification CART, il est possible d'identifier les variables les plus influentes sur le ressenti subjectif, en moyennant leurs *impacts* respectifs, évalués au moyen de la formule (2.17) indiquée dans la Partie 2.3.1.3 du Chapitre 2). Les graphes *a* et *b* de la Figure 5.10 représentent l'*impact moyen relatif* de chaque prédicteur dans les deux meilleurs modèles obtenus par les heuristiques $\text{Bagg}_{\text{CART}}$ et RF3_{CART}. On peut tout d'abord remarquer la relative *parcimonie* de l'heuristique *randomisée*, qui met en évidence un nombre plus restreint de prédicteurs *plus significativement influents*. On constate notamment que sur le graphique *b*, certaines variables qui apparaissent comme particulièrement influentes dans le modèle produit par l'algorithme $\text{Bagg}_{\text{CART}}$, présentent un impact relativement faible, comme par exemple les prédicteurs $X^{(4)}$, $X^{(5)}$, $X^{(8)}$ et $X^{(10)}$, faisant partie de la liste de critères proposés aux experts et retenus pour la caractérisation de la prestation brio.

5.2.5 Conclusion et perspectives

Nous avons consacré cette dernière partie à la description de la problématique industrielle d'*objectivation des prestations* ayant motivé le lancement de ces travaux de thèse en partenariat avec le constructeur automobile Renault. La méthode de scoring développée pendant ces trois ans a été mise en oeuvre pour *objectiver* la prestation *brio*, relative à la sensation d'accélération procurée par un véhicule, et construire un indice permettant de quantifier le niveau de prestation atteint et de situer un véhicule par rapport à ses principaux concurrents.

Les résultats expérimentaux que nous venons de décrire montrent clairement que la méthode développée dans ces travaux de thèse permet de produire des modèles d'ordonnancement plus performants que l'approche *plug-in* actuellement utilisée pour l'objectivation des prestations. De plus, cette méthode nous a permis de proposer un modèle interprétable et d'identifier les critères physiques ayant un impact majeur sur le ressenti des clients via-à-vis de la prestation brio. Pour approfondir cette étude, il pourrait être intéressant de construire des indices du brio pour d'autres types de motorisation. On peut notamment se demander si les critères impactant le brio des véhicules *essence atmosphérique* seront les mêmes que ceux qui expliquent le brio des véhicules *essence sur-alimentés*.

Rappelons pour terminer que la méthode de scoring TreeRank peut aussi être utilisée dans de nombreux autres domaines d'application, comme par exemple l'objectivation du calcul, qui permettrait notamment de réduire le temps de calcul dans des études d'optimisation. Le chapitre suivant donne un aperçu d'une autre application possible de cet algorithme, qui est mis en oeuvre dans une procédure permettant de tester l'homogénéité d'un échantillon d'observations définies dans un espace multi-dimensionnel.

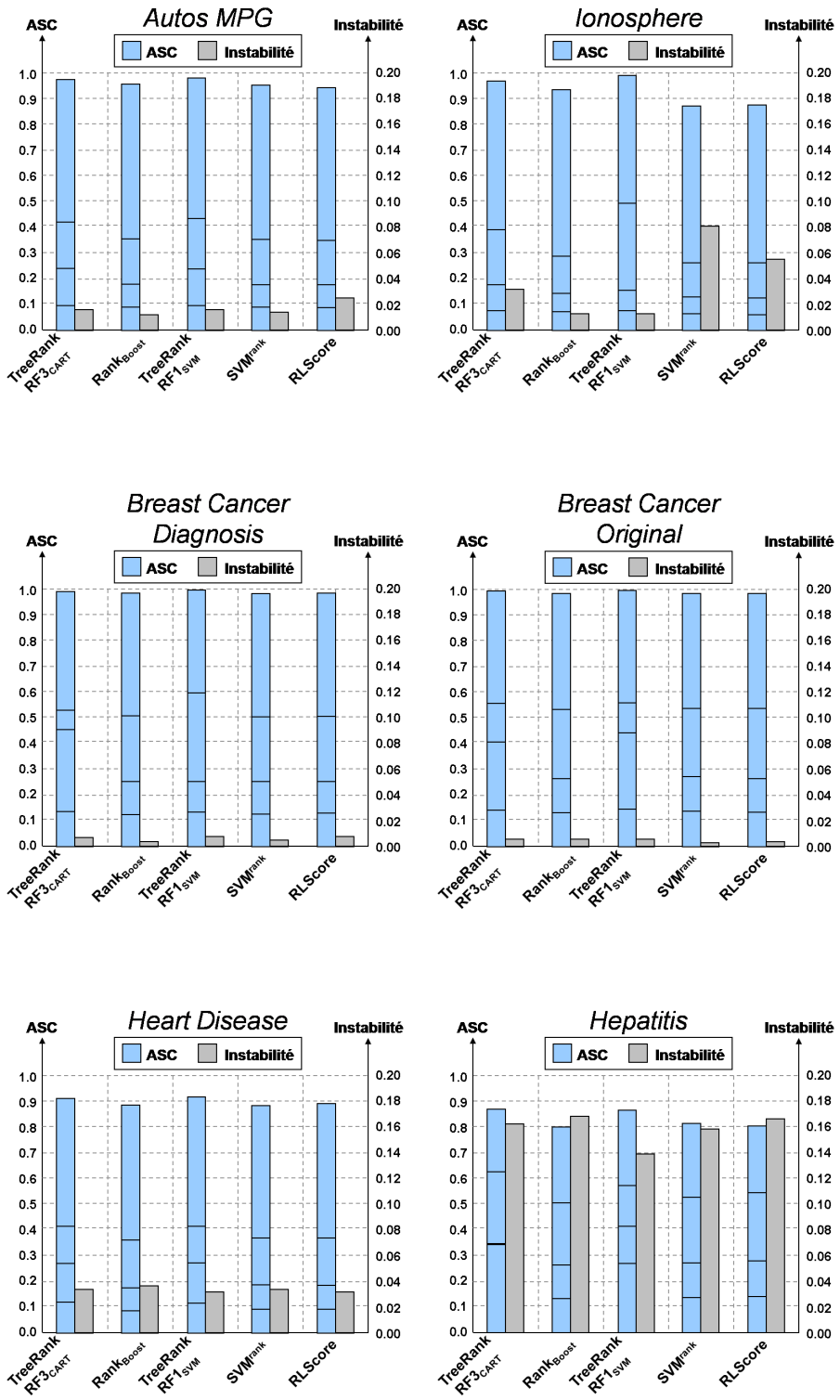


Figure 5.4 – Synthèse graphique de la comparaison de 5 heuristiques de scoring : représentation de la performance $\overline{\mathrm{ASC}}^{(t)}$, de son écart-type $\hat{\sigma}^2$ et des ASC partielles pour les proportions $\{20\%, 10\%, 5\%\}$.

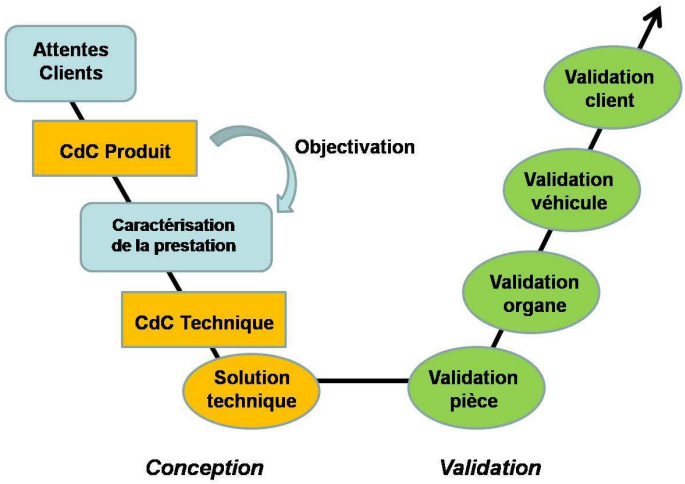


FIGURE 5.5 – Le « V » de la prestation

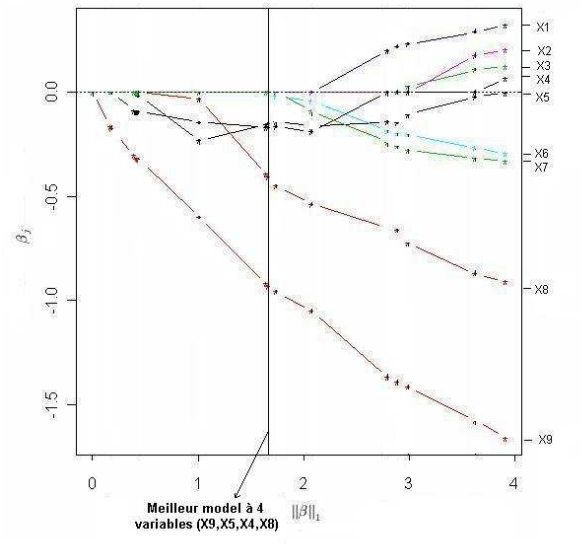


FIGURE 5.6 – Exemple de chemin de régularisation

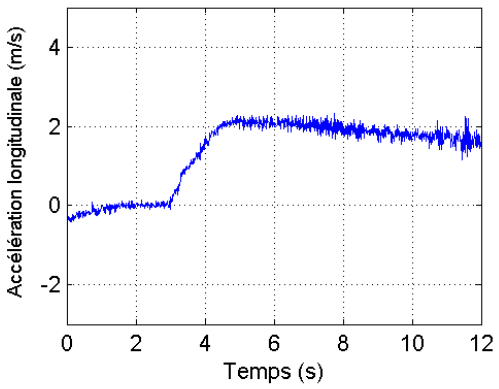


FIGURE 5.7 – Exemple de courbe d'accélération

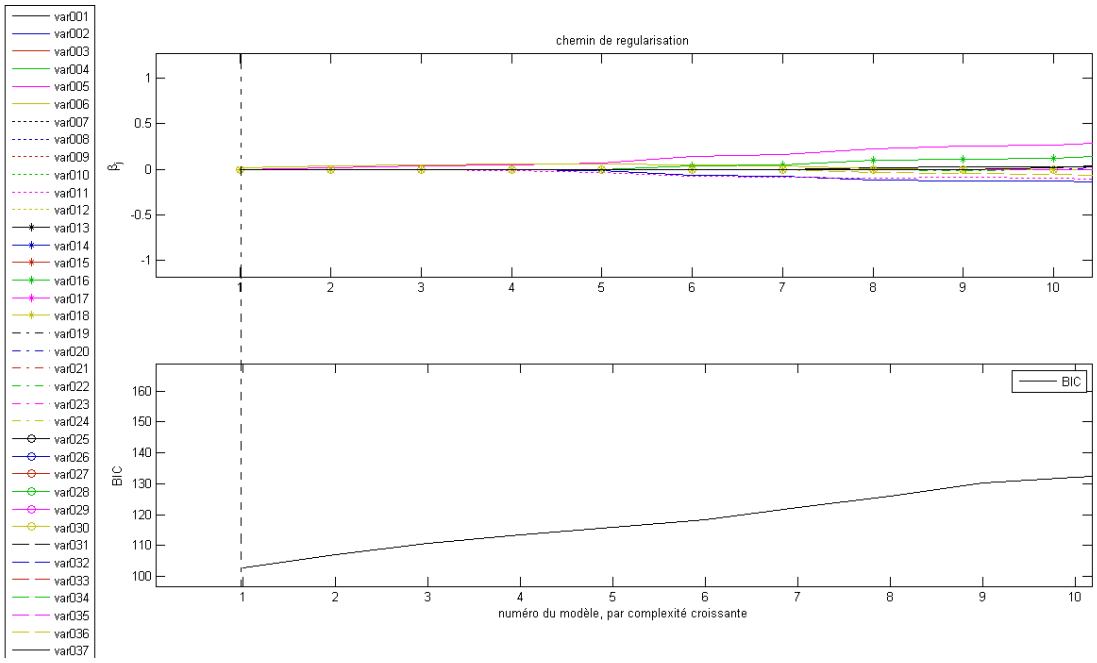


Figure 5.8 – Méthodes ChReg : chemin de régularisation et critère BIC associé.

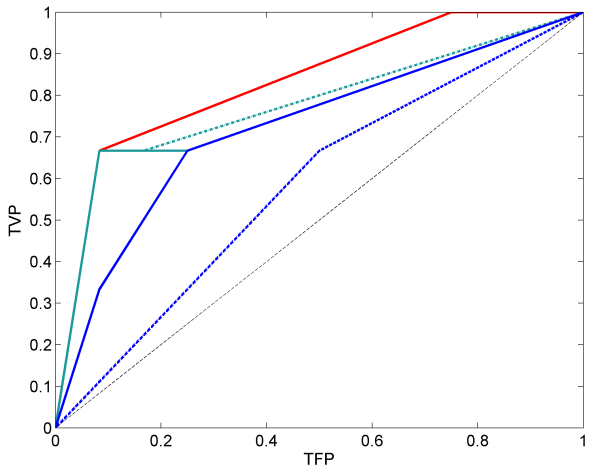
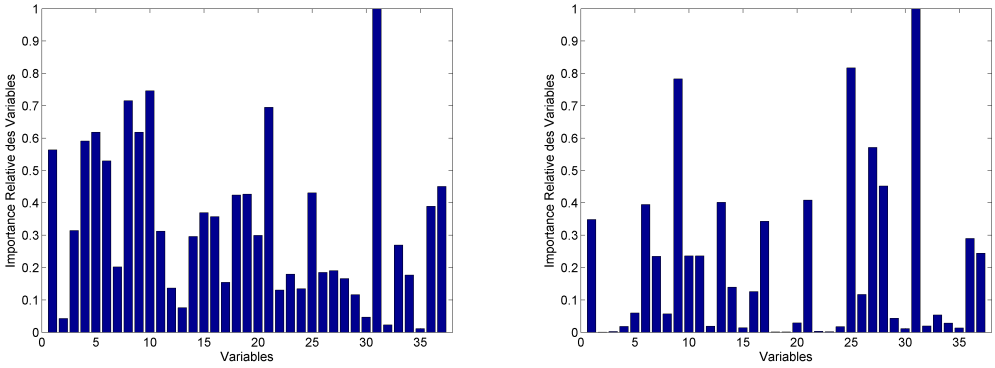


Figure 5.9 – Courbes COR test obtenues sur les données d'objectivation par 5 heuristiques de scoring : ChReg (pointillés bleus), $\text{Bagg}_{\text{CART}}$ (rouge), RF1_{CART} (vert), RF3_{CART} (pointillés verts) et RF5_{CART} (bleu)).



a. Heuristique $\text{Bagg}_{\text{CART}}$.

b. Heuristique RF3_{CART}.

FIGURE 5.10 – Représentation graphique de l'importance relative moyenne des prédicteurs pour les heuristiques $\text{Bagg}_{\text{CART}}$ et RF3_{CART}.

Chapitre 6

Tests d'homogénéité

Tester l'homogénéité de deux *populations* est un problème crucial dans de nombreux domaines comme la médecine, la biologie, la psychométrie ou même l'exploitation de bases de données (*Database Attribute Matching*), qui consiste à déterminer si deux échantillons d'observations sont issus de la même loi de probabilité. Les *essais cliniques* sont un exemple classique d'application, ils permettent notamment de statuer sur l'efficacité d'un nouveau traitement en se basant sur l'observation de deux groupes de patients, l'un traité par le médicament à l'étude et l'autre par un placebo, et en testant l'homogénéité des deux populations.

Les praticiens disposent de nombreux outils pour tester l'homogénéité dans le cas unidimensionnel, *i.e.* lorsque les observations sont définies sur l'espace $\mathcal{X} = \mathbb{R}$. Cependant ce problème devient plus délicat dès lors que l'on observe une population dans un espace $\mathcal{X} = \mathbb{R}^q$, $q \geq 3$, pouvant être de grande dimension. Dans ce contexte, l'approche la plus classique consiste à évaluer si la distance entre les distributions empiriques des deux populations observées est *significative*. Plusieurs méthodes ont été proposées dans la littérature, fondées sur diverses *(pseudo-)métriques*, mais celles-ci se heurtent à un certain nombre de difficultés liées notamment à l'estimation des lois de probabilité en grande dimension et à la définition d'une renormalisation rendant la statistique de test *pivotale*.

L'approche que nous présentons dans ce chapitre est totalement différente et s'inspire du cas uni-dimensionnel : nous proposons d'exploiter le principe du *scoring* afin de *projeter* les observations multi-dimensionnelles sur l'espace $\mathbb{R}$ et de tester l'homogénéité au moyen de tests classiques en dimension $q = 1$. Plus précisément, en nous appuyant sur les liens étroits entre le critère ASC et la statistique de Mann-Whitney, nous proposons une procédure en deux étapes, fondée sur l'optimisation de l'ASC, permettant d'étendre le test de rang de Wilcoxon-Mann-Whitney au cas multi-dimensionnel.

Dans la première partie de ce chapitre, nous commençons par rappeler la problématique et établissons un rapide panorama des différents tests proposés dans la littérature, dans les deux contextes uni- et multi-dimensionnels. Nous mettons ensuite en évidence le lien entre le critère d'ordonnancement de l'ASC et le problème du test de l'homogénéité de deux populations en grande dimension. En nous appuyant sur ce résultat, nous proposons une procédure de test en deux étapes, reposant sur l'optimisation de l'ASC empirique et la mise en oeuvre du célèbre test de rang de Wilcoxon-Mann-Whitney. Nous établissons la consistance de cette procédure, sous certaines conditions relatives à l'étape d'optimisation du critère ASC, et étudions empiriquement sa capacité à détecter la déviation par rapport

à l'hypothèse d'homogénéité sur des exemples de divers niveaux de complexité.

6.1 Tester l'homogénéité d'une population

Le problème statistique du test de l'homogénéité de deux populations consiste à déterminer si celles-ci sont issues de la même loi de probabilité ou s'il existe des différences *significatives* entre les distributions des deux échantillons observés. Ce type de test est souvent utilisé en pratique, par exemple dans le domaine médical où pour évaluer l'efficacité d'un médicament, on compare ses effets à ceux d'un placebo en testant l'homogénéité de deux populations de patients ayant suivi l'un ou l'autre de ces traitements.

Par analogie avec les notations introduites dans les chapitres précédents, considérons un échantillon $\mathcal{D}_N = \{(X_i, Y_i),\, 1 \leq i \leq N\}$ constitué de N copies *i.i.d.* du couple de variables aléatoires $(X, Y) \in \mathcal{X} \times \{-1, +1\}$, avec $\mathcal{X} = \mathbb{R}^q$, $q \geq 1$. Les observations $\mathbf{X^N} = \{X_1, \cdots, X_N\}$ de $\mathcal{D}_n$ se scindent naturellement en une population de *positifs*, que l'on notera $\mathbf{X}_+ = \{X_1^+, \cdots, X_n^+\}$, contenant les observations $(X_i)_{1 \leq i \leq n}$ de $\mathcal{D}_N$ telles que pour tout $i \in \{1, \cdots, n\}$, $Y_i = 1$, et une population de *négatifs*, notée $\mathbf{X}_- = \{X_1^-, \cdots, X_m^-\}$, constituée des observations $(X_j)_{1 \leq j \leq m}$ de $\mathcal{D}_N$ telles que pour tout $j \in \{1, \cdots, m\}$, $Y_j = -1$. Avec ces nouvelles notations, on a $\mathbf{X^N} = \mathbf{X}_+ \cup \mathbf{X}_-$ et $N = n + m$.

De même que précédemment, on note $G(dx)$ la distribution des *positifs* et $H(dx)$ celle des *négatifs*. On les suppose toutes deux définies sur un même support $\mathcal{X}$, continues et surtout inconnues et leur comportement asymptotique est défini comme suit :

$$\frac{n}{N} \longrightarrow_{n,\, m \to \infty} p \in]0, 1[\,, \tag{6.1}$$

où p est le taux théorique de *positifs* dans l'espace $\mathcal{X} \times \{-1, +1\}$. Enfin, on notera $\mathbb{P}_{G,H}$ la distribution définie sur l'espace $\mathcal{X} \times \{-1, +1\}$.

Tester l'homogénéité des deux populations que nous venons de définir consiste à tester les hypothèses suivantes :

$$\mathcal{H}_0 \,:\, G = H \;\; \textit{versus} \;\; \mathcal{H}_1 \,:\, G \neq H, \tag{6.2}$$

où $\mathcal{H}_0$ est l'hypothèse *nulle* ou d'*homogénéité* et $\mathcal{H}_1$ est l'*alternative*, en se basant sur les échantillons $\mathbf{X}_+$ et $\mathbf{X}_-$. Formellement, on peut définir un test d'homogénéité comme une fonction ϕ définie par

$$\phi \,:\, \mathcal{X} \times \{-1, +1\} \;\; \to \;\; \{-1, 1\} \tag{6.3}$$

$$\mathcal{D}_N \;\; \to \;\; \begin{cases} -1, & \text{si } \mathcal{H}_0 \textit{ est vérifiée} \\ +1, & \text{sinon} \end{cases}. \tag{6.4}$$

Dans ce chapitre, on notera respectivement ν et γ les erreurs de première et deuxième espèce, associées respectivement à la détection d'un *faux positif* et d'un *faux négatif*[1]. En pratique, pour évaluer la performance du test, on contrôlera l'erreur de type I en fixant le niveau $(1 - \nu)$ (à 5% par exemple) et on calculera sa puissance $(1 - \gamma)$ pour ce niveau fixé. Nous allons voir que, s'il existe de nombreux outils pour tester l'homogénéité de deux

1. en d'autres termes les probabilités de rejeter à tort respectivement l'hypothèse nulle et l'alternative

populations, ce problème devient plus délicat à résoudre lorsque la dimension q de l'espace augmente.

6.1.1 Cas uni-dimensionnel

Le problème du test de l'homogénéité de deux populations a été largement étudié dans le cas uni-dimensionnel ($\mathcal{X} = \mathbb{R}$) et de nombreux tests ont été proposés, que ce soit dans un cadre paramétrique, supposant une connaissance a priori sur les lois de distributions, ou non-paramétrique (voir par exemple [Lehman & Romano 2005]). Dans la première catégorie on peut citer notamment le test de Student ou *t-test*, permettant de tester l'homogénéité entre les paramètres des distributions, celui de Fisher, permettant de détecter des différences dans la dispersion des distributions ou encore l'ANOVA (pour Analyse de la Variance) mise en oeuvre lorsque l'on considère des données catégorielles.

Dans la seconde catégorie, on peut citer les tests du χ^2, de Kolmogorov-Smirnov ou de Cramer-von Mises par exemple, qui évaluent l'*écart* entre les deux distributions au sens de différentes métriques, ou encore les tests de Wilcoxon-Mann-Whitney, de la médiane ou du *logrank*, appartenant à la famille des *tests de rangs* (voir notamment [Hajek 1962], [Hajek & Sidak 1967], [Lehman & Romano 2005], [Lehmann 2006], [van der Vaart 1998] et [Serfling 1980]). Plus précisément, ces tests sont fondés sur des R-statistiques *linéaires*, de la forme

$$\sum_{i=1}^{N} c_{N_i} a_{N_i} \mathcal{R}_{N_i},$$

où les vecteurs $(c_{N_1}, \cdots, c_{N_N})$ et $(a_{N_1}, \cdots, a_{N_N})$ représentent respectivement des *coefficients* et des *scores*, et où pour tout $i \in \{1, \cdots, N\}$, le rang $\mathcal{R}_{N_i}$ de l'observation X_i dans $\mathbf{X^N}$ est défini comme suit :

$$\mathcal{R}_{N_i} = N \cdot F_{n,m}(X_i), \tag{6.5}$$

où

$$F_{n,m}(t) = \frac{n}{N} \cdot \widehat{G}_n(t) + \frac{m}{N} \cdot \widehat{H}_m(t),$$

en notant

$$\widehat{G}_n(t) = \frac{1}{n} \sum_{i \leq n} \mathbb{I}\{X_i^+ \leq t\} \text{ et } \widehat{H}_m(t) = \frac{1}{m} \sum_{j \leq m} \mathbb{I}\{X_j^- \leq t\}$$

les contreparties empiriques des fonctions de répartition G et H.

Le principe de ces tests est très intuitif et repose sur le simple fait que, sous l'hypothèse nulle, les rangs des *positifs* et des *négatifs* sont distribués uniformément sur $\{1, \ldots, N\}$. Ainsi, on peut détecter des différences entre les distributions en étudiant le positionnement des rangs des *positifs* par rapport à l'ensemble des observations de l'échantillon $\mathcal{D}_N$. Pour cela on peut utiliser par exemple la statistique de Wilcoxon, définie par la somme des rangs des *positifs* $W_{n,m} = \sum_{i=m+1}^{N} \mathcal{R}_{N_i}$, ou de manière équivalente, la statistique de Mann-Whitney $U_{n,m} = 1/nm \cdot \sum_{i,j} \mathbb{I}\{X_j^- < X_i^+\}$, puisque ces deux quantités sont liées comme suit : $W_{n,m} = nm \cdot U_{n,m} + 1/2 \cdot n(n+1)$. Une autre possibilité est de considérer le test de la médiane, fondé sur la statistique $\sum_{i=m+1}^{N} \mathbb{I}\{\mathcal{R}_{N_i} \leq (N+1)/2\}$ dénombrant les *positifs* de rang inférieur à celui de la médiane de l'échantillon.

Les tests de rang sont très populaires auprès des praticiens et ce pour plusieurs raisons. D'une part, comme nous venons de le voir, ces tests sont généralement *intuitifs* et faciles à mettre en oeuvre. D'autre part, les statistiques de rangs ont la particularité d'être indépendantes de la distribution des données de $\mathcal{D}_N$ sous l'hypothèse nulle. On peut donc définir des valeurs *seuil* de façon non-asymptotique, quelque soit le niveau $(1-\nu)$ du test, en tabulant simplement la loi de la R-statistique sous $\mathcal{H}_0$. Enfin et surtout, ces statistiques permettent de définir, selon la classe des hypothèses *alternatives* considérées, des tests *uniformément (localement) les plus puissants*, dont on peut montrer qu'ils sont *asymptotiquement efficaces* (voir [van der Vaart 1998] et [Lehman & Romano 2005]).

6.1.2 Cas multi-dimensionnel

Tester l'homogénéité en grande dimension est un problème plus délicat. Un certain nombre de tests ont toutefois été proposés dans la littérature, dont la plupart reposent sur une même approche consistant à évaluer la *dissimilarité* entre les distributions des *positifs* et des *négatifs* au moyen d'une (pseudo-)métrique probabiliste d sur l'espace des lois définies sur $\mathcal{X} = \mathbb{R}^q$. (Nous renvoyons à la référence [Rachev 1991] pour un descriptif de métriques définies sur des ensembles de lois de probabilité et de leurs applications.)

En se basant sur le fait que sous l'hypothèse nulle, la distance $d(G, H)$ est nulle, une procédure de test consiste à calculer les estimateurs $\widehat{G}_n$ et $\widehat{H}_m$ des distributions des deux populations et à rejeter l'hypothèse $\mathcal{H}_0$ pour de *grandes* valeurs de la statistique de test $d(\widehat{G}_n, \widehat{H}_m)$. Diverses (pseudo-)métriques ont été considérées dans la littérature comme par exemple la distance du χ^2, la divergence de Kullback-Leibler, la distance de Hellinger ou encore la distance de Kolmogorov-Smirnov.

Parmi les tests proposés dans la littérature, on peut citer par exemple les généralisations du test de Kolmogorov-Smirnov proposées dans [Bickel 1969] et [Friedman & Rafsky 1979], les tests introduits dans [Biau & L.Gyorfi 2005] et [Anderson *et al.* 1994], reposant respectivement sur le calcul de la distance $\mathcal{L}_1$ et $\mathcal{L}_2$ entre les densités estimées ou encore les tests proposés dans [Gretton *et al.* 2008b], fondés sur les généralisations de la distance de Kolmogorov-Smirnov du type

$$\mathrm{MMD(G,H)} = \sup_{\mathrm{f}\in\mathcal{F}} \left| \int_{\mathrm{x}\in\mathcal{X}} \mathrm{f(x)G(dx)} - \int_{\mathrm{x}\in\mathcal{X}} \mathrm{f(x)H(dx)} \right|, \tag{6.6}$$

où $\mathcal{F}$ est une classe de fonctions $f : \mathcal{X} \to \mathbb{R}$ *suffisamment riche*, telle que $\mathrm{MMD(G,H)} = 0$ si et seulement si $G = H$. Dans [Gretton *et al.* 2008b], cette distance est appelée la *divergence moyenne maximale* (*Maximum Mean Discrepancy*) et la classe $\mathcal{F}$ choisie est une boule unitaire d'un espace de Hilbert à noyau reproduisant $\mathcal{H}$ (RKHS). Les auteurs montrent que cette classe de fonctions est pertinente, en ce sens qu'elle satisfait la condition pré-citée et qu'elle permet de calculer efficacement la quantité (6.6). Notons qu'une variante de cette approche est proposée dans [Moulines *et al.* 2008], basée sur l'analyse discriminante de Fisher à noyau ([Mika *et al.* 1999]).

Lors de leur mise en pratique, ces méthodes se heurtent toutefois à un certain nombre de difficultés. Les deux principales résident dans la nécessité d'une part d'estimer de façon *consistante* des distributions définies sur l'espace $\mathcal{X}$ de grande dimension et d'autre part d'identifier une *standardisation* (ou *renormalisation*) appropriée pour rendre la statistique

de test *asymptotiquement pivotale*, *i.e.* pour faire en sorte que sa distribution limite soit *non-paramétrée*.

Enfin, on peut citer d'autres approches, comme l'extension du *t-test* au cas multivarié ([Kotz & Nadarajah 2004]), la généralisation du test de Wald-Wolfowitz (*Wald-Wolfowitz runs test*) proposée dans [Friedman & Rafsky 1979] et [Henze & Penrose 1999], qui consiste à dénombrer les connections reliant deux individus issus de populations différentes dans un réseau de type *spanning tree* ou encore le test fondé sur le décompte des *positifs* présents parmi les plus proches voisins de chaque *négatif* et inversement, proposé dans [Hall & Tajvidi 2002]. Le principal défaut de ces deux dernières approches réside dans le temps de calcul limitant fortement leur mise en pratique (voir [Gretton *et al.* 2008b]).

Dans ce chapitre, nous proposons une approche radicalement différente pour tester l'homogénéité de deux populations en grande dimension. Notre objectif est d'étendre l'utilisation des tests de rang au cas multi-dimensionnel, en exploitant les méthodes d'ordonnancement de données étiquetées de façon binaire. Dans la continuité des travaux présentés jusqu'à présent, nous allons considérer le problème d'ordonnancement binaire du point de vue de l'optimisation de l'ASC empirique. Etant donné la relation étroite, mise en évidence dans la Partie 1.2.2.1 du Chapitre 1, entre ce critère et la statistique de Mann-Whitney ([Mann & Whitney 1947]), *étendue* de sorte à tenir compte de la présence d'ex-aequo, cette approche nous amène à proposer une extension du test de Wilcoxon-Mann-Whitney ([Wilcoxon 1945]) au cadre multivarié. Soulignons toutefois qu'il est tout à fait possible de considérer d'autres critères comme par exemple le *P-Norm Push* ou les critères MAP et NCDG (cf Partie 1.2.2.3 du Chapitre 1), qui peuvent aussi s'exprimer sous la forme d'une R-statistique linéaire (voir [Clémençon & Vayatis 2008a]).

6.2 Optimiser l'ASC pour tester l'homogénéité en grande dimension

Dans cette partie, nous allons voir que l'application d'une méthode d'ordonnancement sur les données de l'échantillon $\mathcal{D}_N$ permet d'étendre le champ d'action des tests de rang au cadre multi-dimensionnel. En effet, supposons que l'on soit capable de *projeter* les observations de l'espace $\mathcal{X} = \mathbb{R}^q$, $q > 1$, sur la droite des réels au moyen d'une fonction de score $s : \mathcal{X} \to \mathbb{R}$ en préservant l'éventuelle *dissimilarité* entre les deux populations, de sorte que la fonction s attribue, avec une forte probabilité, des scores plus *importants* aux *positifs* et plus *faibles* aux *négatifs*. La dimension du problème étant alors réduite à 1, on peut ordonner les observations et réaliser un simple test de rang sur la réunion des échantillons $(s(X_1^+), \cdots, s(X_n^+))$ et $(s(X_1^-), \cdots, s(X_m^-))$.

Comme nous l'avons déjà indiqué, nous nous focalisons ici sur l'estimation de fonctions de score *optimales* au sens de l'ASC. Aussi, après avoir établi le lien entre ce critère et la problématique du test de l'homogénéité de deux populations, nous présentons une procédure de test en deux étapes permettant d'étendre le test de Wilcoxon-Mann-Whitney au cas multi-dimensionnel. Nous établissons ensuite la consistance de cette procédure, sous certaines conditions, et étudions enfin son *efficacité* d'un point de vue empirique, en mettant en oeuvre, sur des jeux de données simulées, une heuristique de test basée sur TreeRank.

6.2.1 *Scoring* et test d'homogénéité

Dans la Partie 1.2.1.4 du Chapitre 1, nous avons déjà pu établir un lien entre la courbe COR et la théorie des tests non-paramétriques, en introduisant ce critère comme la courbe de puissance du test d'hypothèse

$$\mathcal{H}_0 \ : \ X \sim H(dx) \ \text{ versus } \ \mathcal{H}_1 \ : \ X \sim G(dx).$$

Dans la problématique du test de l'homogénéité, notre approche repose sur le fait que cette courbe permet de visualiser la *dissimilarité* entre les distributions des *positifs* et des *négatifs*. En effet, plaçons-nous dans le cadre uni-dimensionnel où $\mathcal{X} = \mathbb{R}$, la justification de la Proposition 3 donnée dans le Chapitre 1 implique que la courbe COR est au-dessus de la diagonale du plan $(\mathrm{TFP}, \mathrm{TVP})$ si et seulement si la distribution dG des *positifs* est *stochastiquement* plus grande que celle des *négatifs*, notée dH et qu'elle coïncide avec cette diagonale quand $dG = dH$, *i.e.* sous l'hypothèse nulle.

Il en découle que la *distance* de la courbe COR à la diagonale est une mesure naturelle de la divergence par rapport à l'hypothèse d'homogénéité. Ainsi, le critère scalaire ASC, correspondant à la norme $\mathcal{L}_1$ de la courbe COR, peut être utilisé pour évaluer la *dissimilarité* entre les distributions. Dans ce cas, l'hypothèse nulle correspond au cas particulier où l'ASC est égale à 1/2 (cf Proposition 6 du Chapitre 1).

Replaçons-nous maintenant dans le contexte multivarié où $\mathcal{X} = \mathbb{R}^q$, $q \geq 2$, et reprenons les notations introduites dans les chapitres précédents. Dans ce cadre, les mesures de probabilité G et H sont entièrement caractérisées par les familles de distributions univariées $\{G_s\}_{s\in\mathcal{S}}$ et $\{H_s\}_{s\in\mathcal{S}}$. On peut donc envisager d'évaluer la *dissimilarité* entre $H(dx)$ et $G(dx)$ au moyen des courbes $\{\mathrm{COR(s,.)}\}_{\mathrm{s}\in\mathcal{S}}$, associées aux fonctions de score définies sur $\mathcal{X}$, ou au moyen de la collection de valeurs scalaires $\{\mathrm{ASC(s)}\}_{\mathrm{s}\in\mathcal{S}}$ et ainsi reformuler le test d'homogénéité comme suit :

$$\mathcal{H}_0 : \ \forall s \in \mathcal{S}, \mathrm{ASC(s)} = 1/2 \ \ \textit{versus} \ \ \mathcal{H}_1 : \ \exists \mathrm{s} \in \mathcal{S} \text{ telle que } \mathrm{ASC(s)} > 1/2.$$

Or, d'après la Proposition 3 du Chapitre 1, le supremum $\sup_{s\in\mathcal{S}} \mathrm{ASC(s)}$ est atteint par la classe $\mathcal{S}^*$ des transformées strictement croissantes du rapport de vraisemblance $\phi(x) = dG/dH(x)$, $x \in \mathcal{X}$, permettant de définir le test *uniformément le plus puissant*. Il apparaît donc clairement que les fonctions de score optimales $s^* \in \mathcal{S}^*$ sont des candidates naturelles pour la détection de l'alternative $\mathcal{H}_1$,au sens où la connaissance de l'ordonnancement induit sur les observations de $\mathcal{D}_N$ par une fonction de score optimale $s^* \in \mathcal{S}^*$ suffit pour détecter de manière optimale la déviation par rapport à l'hypothèse d'homogénéité. En effet, en s'appuyant sur le fait que

$$\forall\, s^* \in \mathcal{S}^*, \ \sup_{s\in\mathcal{S}} |\mathrm{ASC(s)} - 1/2| = \mathrm{ASC(s^*)} - 1/2,$$

on peut mettre en évidence le lien entre l'optimisation de l'ASC et le test de l'homogénéité de deux populations, que l'on peut finalement reformuler comme suit :

$$\mathcal{H}_0 : \ \mathrm{ASC}^* = 1/2 \ \ \textit{versus} \ \ \mathcal{H}_1 : \ \mathrm{ASC}^* > 1/2.$$

En effet, comme nous l'avons déjà indiqué dans la Partie 1.2.2.1 du Chapitre 1, une contrepartie empirique de l'ASC^* est donnée par la statistique de Mann-Whitney

$$U_{n,m} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[\mathbb{I}\{s^*(X_i^+) > s^*(X_j^-)\} + \frac{1}{2} \mathbb{I}\{s^*(X_i^+) = s^*(X_j^-)\} \right].$$

Or, nous avons vu qu'il suffisait de connaître l'ordre induit sur les observations de $\mathcal{X}$ par une fonction de score quelconque de $\mathcal{S}^*$ pour pouvoir évaluer la *distance* entre les distributions dG et dH (cf Remarque 6 du Chapitre 1). La statistique de Wilcoxon-Mann-Whitney calculée à partir des rangs des *images* $(s^*(X_i^+), s^*(X_j^-))_{i,j}$ fournit donc un estimateur asymptotiquement efficace de ASC* ([Freund *et al.* 2003], [Ferri *et al.* 2002], [Rakotomamonjy 2004]) et conduit au test uniformément (localement) le plus puissant. Ainsi, la construction d'une règle de score d'ASC optimale permet de calculer cette statistique et de tester l'homogénéité de deux populations, définies dans un espace multi-dimensionnel, à partir de ce test de rang. Notons que l'on étend classiquement le cadre de ce test au cas où certaines observations sont *ex-aequo* en attribuant à celles-ci le rang moyen ([Cheung & Klotz 1997]). Dans la suite, on notera $\mathcal{W}_{n,m}$ la distribution de cette statistique (adaptée à la présence d'*ex-aequo*) sous l'hypothèse d'homogénéité. Cette loi de probabilité étant tabulée (voir [Cheung & Klotz 1997]), il ne sera pas nécessaire d'avoir recours à des résultats d'approximation asymptotique pour construire un test de niveau $(1-\nu)$.

6.2.2 Une procédure de test en deux étapes

En conservant les notations introduites précédemment, on suppose que l'on dispose d'un échantillon $\mathcal{D}_N = \{(X_i, Y_i),\ 1 \leq i \leq N\}$, constitué de $N \geq 0$ copies *i.i.d.* du couple $(X, Y) \in \mathcal{X} \times \{-1, +1\}$. La procédure de test que nous proposons dans cette partie procède en deux étapes. Elle consiste à construire dans un premier temps, sur une partie des données d'apprentissage, une fonction de score $\hat{s}$ d'ASC *quasi-optimale*, puis à calculer la statistique de test

$$\widehat{W}_{n,m} = nm \cdot \widehat{U}_{n,m} + \frac{n(n+1)}{2}$$

à partir des rangs induits par la fonction $\hat{s}$ sur les observations de $\mathcal{D}_N$ non utilisées pour la phase d'apprentissage. Nous détaillons cette heuristique de test dans l'encadré ci-dessous, pour un niveau $\nu \in]0, 1[$ fixé.

Remarque 16 *La fonction de score $\hat{S}$ intervenant dans la première étape de l'heuristique de test, est une simple renormalisation qui permet d'ordonner les observations de l'échantillon $\mathbf{X^{N_1}}$ sur l'intervalle $[0, 1]$. Notons que si l'on considère la distribution $F(dx) = pG(dx) + (1-p)H(dx)$ et que l'on note $F_{s^*}(dt)$ son image par une fonction $s^* \in \mathcal{S}^*$, la fonction de score $S^* = F_{s^*} \circ s^*$ reste optimale et sous le mélange de distributions F, la variable de score $S^*(X)$ est distribuée uniformément sur $[0, 1]$. Observons de plus que la quantité* ASC$^*-1/2$ *peut être vue comme la distance de Earth Mover entre les distributions H_{S^*} et G_{S^*} pour cette normalisation :*

$$\mathrm{ASC}^* - 1/2 = \int_{\mathrm{t}=0}^{1} \{\mathrm{H_{S^*}(t)} - \mathrm{G_{S^*}(t)}\}\, \mathrm{dt}.$$

HEURISTIQUE DE TEST

Initialisation. Soit $n = n_0 + n_1$ et $m = m_0 + m_1$, on pose $\mathbf{X_{N_0}} = \{X_1^+, \ldots, X_{n_0}^+\} \cup \{X_1^-, \ldots, X_{m_0}^-\}$ et $\mathbf{X_{N_1}} = \{X_{n_0+1}^+, \ldots, X_n^+\} \cup \{X_{m_0+1}^-, \ldots, X_m^-\}$, avec $N_0 = n_0 + m_0$ et $N_1 = n_1 + m_1$ tels que

$$\forall\, i \in \{0,\ 1\},\ \frac{n_i}{N_i} \longrightarrow_{n_i, m_i \to \infty} p. \tag{6.7}$$

(6.8)

1. **Ordonnancement.** A partir des observations de l'échantillon $\mathbf{X_{N_0}}$, construire une fonction de score $\hat{s}(x) \in \mathcal{S}_0 \subset \mathcal{S}$ d'ASC empirique maximale. Pour tout $i \in \{n_0 + 1, \cdots, n\}$, calculer les rangs des observations *positives* X_i^+ de l'échantillon $\mathbf{X_{N_1}}$:
$$\forall\, i \in \{1, \cdots, n_1\},\ \widehat{\mathcal{R}}_i = N_1 \hat{S}(X_{n_0+i}^+),$$
où $\widehat{F}_{\hat{s}}(t) = 1/N_1(\sum_{i=1}^{n_1} \mathbb{I}\{\hat{s}(X_{n_0+i}^+) \leq t\} + \sum_{j=1}^{m_1} \mathbb{I}\{\hat{s}(X_{m_0+j}^-) \leq t\})$ et $\hat{S} = \widehat{F}_{\hat{s}} \circ \hat{s}$.
2. **Test de la somme des rangs de Wilcoxon.** Calculer la statistique de la somme des rangs $\widehat{W}_{n_1,m_1} = \sum_{i=1}^{n_1} \widehat{\mathcal{R}}_i$ et rejeter l'hypothèse d'homogénéité $\mathcal{H}_0$ quand :
$$\widehat{W}_{n_1,m_1} \geq Q_{n_1,m_1}(\nu),$$
où $Q_{n_1,m_1}(\nu)$ représente le quantile d'ordre $(1-\nu)$ de la distribution $\mathcal{W}_{n_1,m_1}$.

Le résultat suivant montre que l'étape d'apprentissage n'altère pas les propriétés de consistance du test, à condition toutefois que la règle de score produite soit *universellement consistante*.

Théorème 18 *Posons $\nu \in]0, 1/2[$ et supposons que la méthode d'ordonnancement mise en oeuvre à l'étape 1 de l'heuristique de test produise une règle de score $\hat{s}$, universellement consistante au sens de l'*ASC*. Alors le test de Wilcoxon de la somme des rangs, basé sur les rangs induits par la fonction $\hat{s}$, que l'on notera*

$$\phi = \mathbb{I}\left\{\widehat{W}_{n_1,m_1} \geq Q_{n_1,m_1}(\nu)\right\}$$

est universellement consistant au niveau ν quand n_i et m_i tendent vers l'infini pour tout $i \in \{0,1\}$, au sens où :

1. *ϕ est de niveau ν pour tout n_i et m_i, $i \in \{0,1\}$: $\mathbb{P}_{H,H}\{\phi = +1\} \leq \nu$ pour tout $H(dx)$.*
2. *la puissance du test ϕ tend vers 1 quand n_i et m_i, $i \in \{0,1\}$, tendent vers l'infini, quelle que soit l'alternative considérée : $\lim_{n_i,\, m_i \to \infty} \mathbb{P}_{G,H}\{\phi = +1\} = 1$ pour tout couple de distributions distinctes (G, H).*

Preuve 11 *(Preuve du Théorème 18)*
Notons tout d'abord que sous l'hypothèse d'homogénéité et conditionnellement à l'échantillon $\mathbf{X^{N_0}}$, la statistique $\widehat{W}_{n_1,m_1}$ suit la loi $\mathcal{W}_{n_1,m_1}$. Il vient que pour toute loi de distribution H et pour tout $\nu \in]0, 1/2[$ on a :

$$\mathbb{P}_{H,H}\left\{\widehat{W}_{n_1,m_1} > Q_{n_1,m_1}(\nu) \mid \mathbf{X^{N_0}}\right\} \leq \nu.$$

En prenant l'espérance, on obtient que le test est de niveau ν pour tout n et tout m.

*Pour tout $s \in \mathcal{S}$, considérons maintenant l'*ASC *empirique associée à la fonction s calculée*

à partir des observations de l'échantillon $\mathbf{X^{N_1}}$*, que l'on notera* $U_{n_1,m_1}(s)$*. On rappelle tout d'abord qu'il existe un théorème pour les* U*-statistiques, énoncé dans [Serfling 1980] dans le contexte spécifique du test de l'homogénéité de deux populations (two-sample problem) qui nous donne que :*

$$\sqrt{N}\{U_{n_1,m_1}(s) - \mathrm{ASC(s)}\} = \frac{\sqrt{\mathrm{N_1}}}{\mathrm{n_1}} \sum_{\mathrm{i=1}}^{\mathrm{n_1}} \Big\{ \mathrm{H_s(s(X^+_{i+n_0}))} - \mathbb{E}[\mathrm{H_s(s(X^+_1))}] \Big\}$$
$$- \frac{\sqrt{N_1}}{m_1} \sum_{j=1}^{m_1} \Big\{ G_s(s(X^-_{j+m_0})) - \mathbb{E}[G_s(s(X^-_1))] \Big\} + o_{\mathbb{P}_{G,H}}(1),$$

quand n *et* m *tendent vers l'infini. Il en découle en particulier que pour tout couple de distributions* (G, H)*, la variable aléatoire centrée* $\sqrt{N}\{U_{n_1,m_1}(s) - \mathrm{ASC(s)}\}$ *est asymptotiquement gaussienne, de variance limite*

$$\sigma_s^2(G,H) = Var(H_s(s(X_1^+)))/p + Var(G_s(s(X_1^-)))/(1-p)$$

sous $\mathbb{P}_{G,H}$*. Notons de plus que* $\sigma_s^2(H,H) = 1/(12p(1-p))$ *pour toute fonction* $s \in \mathcal{S}$ *de distribution* $H_s(dt)$ *continue (cf Théroème* 12.4 *de [van der Vaart 1998] pour plus de détails).*

Plaçons-nous maintenant sous l'alternative $\mathcal{H}_1$ *caractérisée par un couple de distributions distinctes* (G, H)*, de sorte que* $\mathrm{ASC}^* > 1/2$*. En posant* $\widehat{U}_{n_1,m_1} = U_{n_1,m_1}(\hat{s})$*, l'erreur de type II du test* ϕ *donnée par*

$$\mathbb{P}_{G,H}\Big\{ \widehat{W}_{n_1,m_1} \leq Q_{n_1,m_1}(\nu) \Big\}$$

peut être bornée par la quantité suivante :

$$\mathbb{P}_{G,H}\{\sqrt{N_1}(\widehat{U}_{n_1,m_1} - \mathrm{ASC}^*) \leq \epsilon_{\mathrm{n_1,m_1}}(\nu)\} + \mathbb{P}_{\mathrm{G,H}}\{\sqrt{\mathrm{N_1}}(\mathrm{ASC(\hat{s})} - \mathrm{ASC}^*) \leq \epsilon_{\mathrm{n_1,m_1}}(\nu)\},$$

où

$$\epsilon_{n_1,m_1}(\nu) = \sqrt{N_1}(\frac{Q_{n_1,m_1}(\nu)}{n_1 m_1} - \frac{n_1+1}{2m_1} - \frac{1}{2}) - \sqrt{N_1}(\mathrm{ASC}^* - \frac{1}{2}).$$

Notons qu'en vertu du Théorème Centrale Limite (TCL) rappelé précédemment, on sait que la quantité $\sqrt{N_1}(Q_{n_1,m_1}(\nu)/(n_1 m_1) - (n_1+1)/(2m_1))$ *tend vers* $z_\nu/\sqrt{12p(1-p)}$*, où* z_ν *est le quantile d'ordre* $(1-\nu)$ *de la loi normale centrée réduite.*

Enfin, on peut décomposer la quantité $\mathrm{ASC}^* - \widehat{\mathrm{U}}_{\mathrm{n_1,m_1}}$ *sous la forme d'une somme comme suit :*

$$\mathrm{ASC}^* - \widehat{\mathrm{U}}_{\mathrm{n_1,m_1}} = \mathrm{ASC}^* - \mathrm{ASC(\hat{s})} + \mathrm{ASC(\hat{s})} - \widehat{\mathrm{U}}_{\mathrm{n_1,m_1}},$$

où le premier terme représente le déficit en ASC *induit par la fonction* $\hat{s}$ *et le second, la déviation entre l'*ASC *théorique et empirique calculée sur l'échantillon* $\mathbf{X^{N_1}}$*. En s'appuyant sur cette re-formulation, la convergence de l'erreur de type II vers* 0 *quand* n_i *et* m_i *tendent vers l'infini, pour tout* $i \in \{0,1\}$*, découle finalement de la combinaison entre l'hypothèse de consistance universelle de* $\hat{s}$*, le TCL pour les* U*-statistiques rappelé plus haut et le théorème de la convergence dominée.*

Remarque 17 *(Taux de convergence)*
En émettant des hypothèses adéquates sur la complexité de la classe des fonctions de score

*$\mathcal{S}_0$ sur laquelle on cherche à optimiser l'*ASC *empirique, on peut borner la capacité de généralisation (en termes d'*ASC*) des règles de score obtenues de manière indépendante de la distribution des données (voir notamment le Corollaire* 6 *de [Agarwal et al. 2005] et le Corollaire* 3 *de [Clémençon et al. 2008]). Le Théorème 18 indique que la combinaison de ces bornes avec le Théorème de Berry-Esseen pour les U-statistiques ([Serfling 1980]) permettrait de calculer un taux de convergence pour la détérioration de l'erreur de type II du test de Wilcoxon fondé sur des scores, sous n'importe quelle alternative* (G, H)*. Si l'on peut établir par exemple une borne classique* $1/\sqrt{N_0}$ *pour* $\hat{s}(x)$*, on peut alors montrer que choisir* $N_1 \sim N_0$ *conduit à un taux de l'ordre de* $O_{\mathbb{P}_{G,H}}(1/\sqrt{N_0})$ *pour le test* ϕ*.*

Naturellement, n'importe quel algorithme d'ordonnancement peut être mis en oeuvre à l'étape 1 de l'heuristique de test proposée. Il faut cependant s'assurer d'une part que la classe $\mathcal{S}_0$ des fonctions de score candidates soit suffisamment *riche* pour garantir un biais $\mathrm{ASC}^* - \sup_{s \in \mathcal{S}_0} \mathrm{ASC}(s)$ faible et, d'autre part, que l'on peut contrôler sa complexité au moyen par exemple de la dimension VC de la collection des ensembles

$$\{\{x \in \mathcal{X} : \ s(x) \geq t\}, \ (s,t) \in \mathcal{S}_0 \times \mathbb{R}\}$$

comme dans [Clémençon & Vayatis 2007] ou des moyennes de Rademacher conditionnelles comme dans [Clémençon *et al.* 2008]. Soulignons que sous ces hypothèses, des résultats de consistance universelle ont été établis pour les fonctions de score d'ASC empirique maximale, ainsi que des bornes généralisées indépendantes de la distribution (voir notamment [Agarwal *et al.* 2005] et [Clémençon *et al.* 2008]).

Rappelons une fois de plus que l'approche proposée peut aussi être étendue à d'autres critères d'ordonnancement. On pourrait notamment considérer d'autres statistiques de rangs. Notons d'ailleurs, qu'une ébauche des limites de la théorie garantissant la performance statistique de l'approche ERM pour des risques empiriques fonctionnels définis par des R-estimateurs peut être trouvée dans [Clémençon & Vayatis 2008a]. Une autre possibilité serait encore de mettre en oeuvre une analyse linéaire discriminante (LDA pour *Linear Discriminant Analysis*), à condition de se restreindre au cas particulier où G et H sont deux distributions gaussiennes de même structure de variance-covariance. On peut alors appliquer un test univarié de Student (ou *t-test*) sur les données *projetées* $\{\widehat{\delta}(X_i^+) : \ 1 \leq i \leq n\}$ et $\{\widehat{\delta}(X_j^-) : \ 1 \leq j \leq m\}$, où $\widehat{\delta}$ représente la fonction discriminante empirique, afin d'évaluer la déviation par rapport à l'hypothèse d'homogénéité. Cette procédure pourrait constituer une alternative intéressante aux extensions du *t-test standard* au cadre multi-dimensionnel (voir notamment [Kotz & Nadarajah 2004]).

6.2.3 Résultats expérimentaux

Dans cette partie, nous proposons d'évaluer empiriquement l'*efficacité* de l'heuristique de test précédemment décrite, reposant sur la mise en oeuvre de l'algorithme TREERANK. Pour ce faire, nous allons étudier l'évolution de la *puissance* $(1-\gamma)$ du test d'homogénéité ϕ, fondé sur les scores des observations de $\mathcal{X}$, pour des exemples de divers niveaux de *complexité*, à la fois au sens de la distance entre les distributions G et H et au sens de la dimension $q \geq 1$ du problème.

De même que dans le Chapitre 4, nous nous sommes inspirés des simulations proposées dans [Gretton *et al.* 2008a] pour générer quatre mélanges de gaussiennes. Dans chaque

exemple, le taux théorique de positifs est fixé à $p = 1/2$, les distributions des *positifs* et des *négatifs* ont les mêmes matrices de variance-covariance $\Sigma_+ = \Sigma_- = I_q$, où I_q désigne la matrice identité dans $\mathbb{R}^q$, et l'on fait varier la complexité du problème en contrôlant la distance euclidienne $\Delta_{+/-} \in \{0.2, 0.1, 0.08, 0.05\}$ entre leurs centres, et la dimension $q \in \{10, 30\}$. Afin de visualiser le niveau de difficulté des exemples générés, nous avons représenté les courbes COR* associées aux quatre mélanges en dimension 10 sur la Figure 6.1 ci-dessous, respectivement en noir pour le cas où $\Delta_{+/-} = 0.2$, en bleu pour $\Delta_{+/-} = 0.1$, en vert pour $\Delta_{+/-} = 0.08$ et en rouge pour le cas le plus difficile où $\Delta_{+/-} = 0.05$.

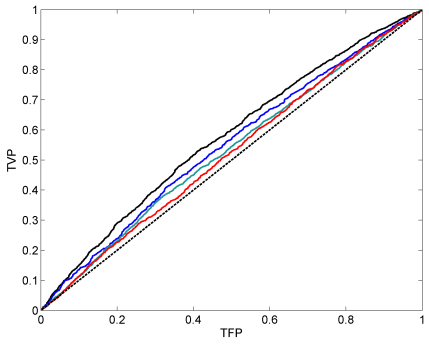


FIGURE 6.1 – Courbes COR* associées aux quatre exemples générés pour $q = 10$.

Pour chaque mélange de lois, nous avons généré $B = 100$ échantillons constitués de $N = 2000$ copies *i.i.d.* du couple $(X, Y) \in \mathbb{R}^q \times \{-1, +1\}$, afin de calculer la puissance *moyenne* de trois procédures de test pour un niveau $(1 - \nu) = 0.05$. Dans un premier temps, nous avons appliqué la procédure en deux étapes, proposée dans ce chapitre, que l'on notera $\mathrm{S} - \mathrm{WMW}_{\mathrm{Trk}}$. A la première étape de la procédure, l'optimisation de l'ASC empirique repose sur la mise en oeuvre de l'heuristique $\mathrm{TRK}_{\mathrm{CART}}$ (cf Chapitre 2), sur un sous-échantillon constitué de $N/2 = n_0 + m_0$ observations, sélectionnées aléatoirement parmi l'échantillon d'apprentissage. A la deuxième étape, le test de la somme des rangs de Wilcoxon est réalisé sur les observations restantes, avec $N/2 = n_1 + m_1$. Nous avons ensuite mis en oeuvre deux tests d'homogénéité, proposés dans [Gretton *et al.* 2008a], fondés sur l'estimation de la distance MMD entre les distributions des *positifs* et des *négatifs* sur une classe de fonctions $\mathcal{F}$ définie comme une boule unitaire d'un RKHS muni d'un noyau gaussien (cf Partie 6.1.2 précédente et [Gretton *et al.* 2008a]) : une version bootstrap $\mathrm{MMD}_{\mathrm{boot}}$ et une version basée sur l'approximation des *moments* par des courbes de Pearson $\mathrm{MMD}_{\mathrm{mom}}$. Les résultats obtenus sont résumés dans le Tableau 6.1 ci-dessous.

Sur les exemples en dimension $q = 10$, on constate que la puissance de la procédure de test proposée dans ce chapitre est, dans la plupart des cas, nettement meilleure que celle des tests MMD introduits dans [Gretton *et al.* 2008a] et que cet écart se creuse d'autant plus que la dimension du problème augmente. En effet, la puissance des deux tests MMD décroît brutalement dès le second mélange ($\Delta_{+/-}\mu = 0.1$), on constate une diminution de 76% en dimension 10 et de plus de 80% en dimension 30 alors que, dans le même temps, la puissance du test $\mathrm{S} - \mathrm{WMW}_{\mathrm{Trk}}$ décroît de « seulement » 36% et 45% respectivement pour ces deux dimensions. On remarque tout de même que la procédure proposée dans ce chapitre enregistre une baisse importante de sa puissance sur le quatrième et dernier mélange étudié, de l'ordre de 57% en dimension 10 et de 50% en dimension 30. Notons

Dim. q	MMD_{boot}	MMD_{mom}	$\text{S}-\text{WMW}_{\text{Trk}}$
Mélange 1 : $\Delta_{+/-}\mu = 0.2$			
$q = 10$	86%	86%	90%
$q = 30$	54%	58%	85%
Mélange 2 : $\Delta_{+/-}\mu = 0.1$			
$q = 10$	20%	20%	58%
$q = 30$	9%	7%	47%
Mélange 3 : $\Delta_{+/-}\mu = 0.08$			
$q = 10$	19%	19%	42%
$q = 30$	5%	7%	32%
Mélange 4 : $\Delta_{+/-}\mu = 0.05$			
$q = 10$	11%	13%	18%
$q = 30$	6%	6%	16%

Table 6.1 – Puissance moyenne des tests d'homogénéité réalisés pour un niveau de 5%.

qu'elle présente malgré tout une puissance supérieure à celle des tests MMD auxquels elle est comparée.

6.3 Conclusion et perspectives

Nous nous sommes intéressés au problème du test de l'homogénéité de deux populations définies dans un espace $\mathcal{X}$ pouvant être de grande dimension. Nous avons vu que si les praticiens disposent de nombreux outils pour résoudre ce problème quand $\mathcal{X} = \mathbb{R}$, la tâche est moins évidente dans le cas multi-dimensionnel où $\mathcal{X} = \mathbb{R}^q$, $q \geq 2$. La grande majorité des tests proposés dans ce contexte consistent à évaluer la *dissimilarité* entre les distributions des deux populations au moyen de diverses *pseudo-métriques* définies sur l'espace probabiliste $\mathcal{X}$. En pratique ces approches sont confrontées à un certain nombre de difficultés, comme par exemple la nécessité de calculer une renormalisation *adéquate*, afin de rendre la statistique de test *pivotale*.

Dans ce chapitre, nous avons proposé une approche totalement différente, inspirée du cas uni-dimensionnel et s'appuyant sur le principe du *scoring*. Notre idée consiste à *projeter* les observations de l'espace multi-dimensionnel $\mathcal{X} \times \{-1, +1\}$ sur la droite des réels au moyen d'une règle de score [2] et de tester l'homogénéité des *images* des deux populations définies sur $\mathbb{R}$ à partir d'un test classique. Plus précisément, en nous appuyant sur les relations existant entre le critère d'ordonnancement ASC et la statistique de test de Mann-Whitney, nous avons proposé une procédure permettant d'étendre le célèbre test de Wilcoxon-Mann-Whitney au cas multi-dimensionnel en se basant sur une étape *préalable* d'optimisation de l'ASC empirique sur une partie des données d'apprentissage.

Nous avons notamment montré la *consistance* de ce test, sous condition que la règle de score d'ASC maximale obtenue à l'issue de la première étape de la procédure soit *universellement consistante*. Nous avons aussi étudié, d'un point de vue empirique, l'évolution

2. de sorte que les *positifs* (resp. les *négatifs*) soient placés en tête (resp. en queue) de classement avec la plus forte probabilité

de la puissance d'une heuristique de test fondée sur la mise en oeuvre de l'algorithme TreeRank, en fonction de la complexité du problème, au sens de la distance entre les distributions des deux populations mais aussi de la dimension q du problème. Cette étude succincte nous a permis de mettre en évidence les bonnes performances de cette heuristique, notamment en comparaison des deux versions asymptotiques du test MMD proposé dans [Gretton *et al.* 2008a] auxquelles nous l'avons comparée.

Pour aller plus loin, il serait intéressant d'étudier d'un point de vue théorique le comportement de la puissance du test quand l'alternative $\mathcal{H}_1$ converge vers l'hypothèse d'homogénéité $\mathcal{H}_0$. Soulignons aussi que cette approche pourrait être mise en oeuvre pour des critères d'ordonnancement autres que l'ASC et permettre ainsi d'étendre le champ d'action d'un certain nombre de tests d'homogénéité *standards* dans le cas uni-dimensionnel.

Annexes

Annexe A

Résultats d'expériences

Cette annexe regroupe quatre tableaux résumant les résultats des deux études expérimentales, décrites dans les Parties 5.1.3 et 5.1.4 du Chapitre 5, réalisées sur une collection de 13 jeux de données. Nous en rappelons le contenu ci-dessous :

- le *Tableau A.1* indique les performances ($\overline{\text{ASC}}^{(t)}$), évaluées au moyen de l'ASC de test moyennée sur $V = 10$ échantillons de validations et de son écart-type $\hat{\sigma}^2$, indiqué entre parenthèses, de 8 versions de l'algorithme TreeRank.

- *Tableau A.2* : indique les performances de ces mêmes heuristiques, en termes d'ASC *tronquées*, les écarts-types $\hat{\sigma}^2$ étant indiqués entre parenthèses.

- le *Tableau A.3* indique les performances ($\overline{\text{ASC}}^{(t)}$), évaluées au moyen de l'ASC de test moyennée sur $V = 10$ échantillons de validations et de son écart-type $\hat{\sigma}^2$, indiqué entre parenthèses, de 5 heuristiques de scoring.

- *Tableau A.4* : indique les performances de ces mêmes heuristiques, en termes d'ASC *tronquées*, les écarts-types $\hat{\sigma}^2$ étant indiqués entre parenthèses.

	TRK_{CART}	TRK_{SVM}	$\text{Bagg}_{\text{CART}}$	Bagg_{SVM}	RF1_{CART}	RF1_{SVM}	RF3_{CART}	RF5_{CART}
Simulations								
GaussEasy20d	0.650 ($\pm$0.031)	0.754 ($\pm$0.024)	0.738 ($\pm$0.024)	0.790 ($\pm$0.023)	0.743 ($\pm$0.018)	0.786 ($\pm$0.022)	0.739 ($\pm$0.017)	0.711 ($\pm$0.028)
GaussMed20d	0.594 ($\pm$0.035)	0.702 ($\pm$0.024)	0.678 ($\pm$0.024)	0.726 ($\pm$0.025)	0.709 ($\pm$0.029)	0.738 ($\pm$0.026)	0.681 ($\pm$0.025)	0.665 ($\pm$0.026)
GaussHard20d	0.558 ($\pm$0.057)	0.621 ($\pm$0.046)	0.589 ($\pm$0.030)	0.629 ($\pm$0.045)	0.617 ($\pm$0.039)	0.650 ($\pm$0.048)	0.590 ($\pm$0.032)	0.591 ($\pm$0.051)
Données *benchmark* (UCI)								
Congressional Vote	0.749 ($\pm$0.096)	0.991 ($\pm$0.024)	0.905 ($\pm$0.06)	0.991 ($\pm$0.026)	0.985 ($\pm$0.028)	0.987 ($\pm$0.024)	0.992 ($\pm$0.015)	0.988 ($\pm$0.021)
Australian Credit	0.577 ($\pm$0.057)	0.921 ($\pm$0.032)	0.852 ($\pm$0.038)	0.925 ($\pm$0.034)	0.882 ($\pm$0.026)	0.926 ($\pm$0.031)	0.934 ($\pm$0.031)	0.939 ($\pm$0.031)
German Credit	0.711 ($\pm$0.071)	0.771 ($\pm$0.034)	0.737 ($\pm$0.045)	0.794 ($\pm$0.024)	0.771 ($\pm$0.031)	0.793 ($\pm$0.022)	0.769 ($\pm$0.029)	0.775 ($\pm$0.044)
Japanese Credit	0.789 ($\pm$0.053)	0.924 ($\pm$0.049)	0.839 ($\pm$0.048)	0.926 ($\pm$0.044)	0.887 ($\pm$0.038)	0.923 ($\pm$0.038)	0.931 ($\pm$0.042)	0.930 ($\pm$0.043)
Autos MPG	0.927 ($\pm$0.049)	0.961 ($\pm$0.041)	0.967 ($\pm$0.022)	0.982 ($\pm$0.017)	0.978 ($\pm$0.014)	0.978 ($\pm$0.016)	0.972 ($\pm$0.016)	0.964 ($\pm$0.021)
Ionosphere	0.926 ($\pm$0.042)	0.943 ($\pm$0.049)	0.966 ($\pm$0.026)	0.985 ($\pm$0.014)	0.971 ($\pm$0.026)	0.990 ($\pm$0.013)	0.968 ($\pm$0.032)	0.966 ($\pm$0.032)
Breast Cancer Diagnosis	0.958 ($\pm$0.02)	0.989 ($\pm$0.008)	0.986 ($\pm$0.016)	0.990 ($\pm$0.015)	0.990 ($\pm$0.009)	0.994 ($\pm$0.008)	0.988 ($\pm$0.007)	0.990 ($\pm$0.007)
Breast Cancer Original	0.984 ($\pm$0.017)	0.995 ($\pm$0.005)	0.990 ($\pm$0.009)	0.994 ($\pm$0.007)	0.995 ($\pm$0.005)	0.995 ($\pm$0.006)	0.993 ($\pm$0.006)	0.993 ($\pm$0.006)
Heart Disease	0.794 ($\pm$0.061)	0.868 ($\pm$0.048)	0.798 ($\pm$0.074)	0.908 ($\pm$0.049)	0.869 ($\pm$0.067)	0.917 ($\pm$0.032)	0.911 ($\pm$0.034)	0.907 ($\pm$0.048)
Hepatitis	0.771 ($\pm$0.171)	0.813 ($\pm$0.154)	0.813 ($\pm$0.132)	0.872 ($\pm$0.14)	0.850 ($\pm$0.141)	0.864 ($\pm$0.139)	0.868 ($\pm$0.162)	0.842 ($\pm$0.17)

TABLE A.1 – Résumé des résultats numériques de la comparaison de 8 heuristiques TREERANK. Ce Tableau indique la performance $\overline{\text{ASC}}^{(t)}$ de chaque procédure d'apprentissage, son écart-type $\hat{\sigma}^2$ étant indiqué entre parenthèses.

	TRK_{CART}	TRK_{SVM}	$Bagg_{CART}$	$Bagg_{SVM}$	$RF1_{CART}$	$RF1_{SVM}$	$RF3_{CART}$	$RF5_{CART}$
Simulations								
GaussEasy20d	0.258 (±0.034) 0.138 (±0.019) 0.070 (±0.001)	0.295 (±0.022) 0.152 (±0.015) 0.074 (±0.009)	0.295 (±0.023) 0.161 (±0.016) 0.082 (±0.008)	0.314 (±0.019) 0.164 (±0.018) 0.083 (±0.009)	0.302 (±0.025) 0.160 (±0.014) 0.084 (±0.008)	0.311 (±0.020) 0.163 (±0.018) 0.083 (±0.010)	0.291 (±0.024) 0.155 (±0.020) 0.082 (±0.012)	0.286 (±0.015) 0.156 (±0.011) 0.083 (±0.008)
GaussMed20d	0.240 (±0.033) 0.128 (±0.022) 0.065 (±0.013)	0.269 (±0.023) 0.136 (±0.013) 0.066 (±0.009)	0.265 (±0.030) 0.145 (±0.020) 0.08 (±0.010)	0.286 (±0.030) 0.154 (±0.025) 0.08 (±0.014)	0.304 (±0.030) 0.167 (±0.016) 0.087 (±0.010)	0.302 (±0.021) 0.166 (±0.014) 0.088 (±0.008)	0.270 (±0.030) 0.148 (±0.023) 0.082 (±0.011)	0.271 (±0.032) 0.151 (±0.026) 0.079 (±0.013)
GaussHard20d	0.212 (±0.036) 0.111 (±0.023) 0.055 (±0.017)	0.214 (±0.025) 0.115 (±0.016) 0.061 (±0.006)	0.220 (±0.031) 0.113 (±0.014) 0.065 (±0.010)	0.217 (±0.030) 0.111 (±0.020) 0.054 (±0.010)	0.253 (±0.039) 0.132 (±0.017) 0.070 (±0.012)	0.252 (±0.030) 0.132 (±0.016) 0.073 (±0.010)	0.222 (±0.029) 0.128 (±0.015) 0.069 (±0.012)	0.228 (±0.039) 0.120 (±0.029) 0.061 (±0.020)
Données *benchmark* (UCI)								
Congressional Vote	0.261 (±0.060) 0.128 (±0.052) 0.069 (±0.046)	0.595 (±0.105) 0.186 (±0.004) 0.093 (±0.002)	0.359 (±0.049) 0.232 (±0.075) 0.11 (±0.031)	0.650 (±0.112) 0.186 (±0.004) 0.093 (±0.002)	0.473 (±0.061) 0.217 (±0.064) 0.094 (±0)	0.477 (±0.054) 0.201 (±0.042) 0.094 (±0)	0.495 (±0.059) 0.201 (±0.042) 0.094 (±0)	0.474 (±0.053) 0.204 (±0.054) 0.094 (±0)
Australian Credit	0.257 (±0.050) 0.134 (±0.023) 0.068 (±0.011)	0.414 (±0.024) 0.256 (±0.082) 0.115 (±0.028)	0.419 (±0.014) 0.235 (±0.031) 0.141 (±0.041)	0.423 (±0.016) 0.252 (±0.040) 0.115 (±0.012)	0.771 (±0.059) 0.663 (±0.075) 0.543 (±0.081)	0.425 (±0.012) 0.248 (±0.039) 0.111 (±0.002)	0.429 (±0.014) 0.273 (±0.059) 0.122 (±0.031)	0.429 (±0.014) 0.278 (±0.061) 0.129 (±0.038)
German Credit	0.237 (±0.031) 0.123 (±0.019) 0.061 (±0.014)	0.254 (±0.017) 0.133 (±0.018) 0.065 (±0.004)	0.245 (±0.022) 0.128 (±0.020) 0.063 (±0.008)	0.269 (±0.009) 0.146 (±0.016) 0.098 (±0.033)	0.264 (±0.016) 0.146 (±0.021) 0.075 (±0.014)	0.266 (±0.012) 0.148 (±0.020) 0.081 (±0.021)	0.260 (±0.020) 0.144 (±0.022) 0.083 (±0.029)	0.263 (±0.017) 0.147 (±0.027) 0.077 (±0.015)
Japanese Credit	0.373 (±0.043) 0.220 (±0.059) 0.106 (±0.024)	0.423 (±0.050) 0.232 (±0.053) 0.105 (±0.005)	0.406 (±0.024) 0.250 (±0.047) 0.131 (±0.035)	0.416 (±0.026) 0.245 (±0.052) 0.127 (±0.030)	0.416 (±0.015) 0.256 (±0.063) 0.125 (±0.033)	0.414 (±0.020) 0.256 (±0.068) 0.125 (±0.032)	0.417 (±0.023) 0.260 (±0.073) 0.141 (±0.047)	0.418 (±0.027) 0.274 (±0.081) 0.123 (±0.035)
Autos MPG	0.420 (±0.011) 0.229 (±0.081) 0.092 (±0.007)	0.462 (±0.144) 0.184 (±0.010) 0.092 (±0.005)	0.523 (±0.112) 0.190 (±0.002) 0.094 (±0.004)	0.584 (±0.109) 0.190 (±0) 0.095 (±0)	0.418 (±0.039) 0.244 (±0.087) 0.095 (±0)	0.433 (±0.058) 0.238 (±0.079) 0.095 (±0)	0.421 (±0.016) 0.240 (±0.08) 0.095 (±0)	0.418 (±0.046) 0.236 (±0.076) 0.095 (±0)
Ionosphere	0.360 (±0.127) 0.146 (±0.007) 0.074 (±0.003)	0.352 (±0.129) 0.149 (±0.008) 0.074 (±0.004)	0.408 (±0.116) 0.181 (±0.052) 0.078 (±0)	0.477 (±0.072) 0.156 (±0) 0.078 (±0)	0.411 (±0.099) 0.179 (±0.048) 0.078 (±0)	0.494 (±0.062) 0.156 (±0) 0.078 (±0)	0.392 (±0.086) 0.178 (±0.044) 0.078 (±0)	0.391 (±0.032) 0.179 (±0.038) 0.081 (±0.008)

	TRK_{CART}	TRK_{SVM}	$\text{Bagg}_{\text{CART}}$	Bagg_{SVM}	RF1_{CART}	RF1_{SVM}	RF3_{CART}	RF5_{CART}
Breast Cancer Diagnosis	0.503 (±0.015)	0.565 (±0.063)	0.578 (±0.036)	0.680 (±0.071)	0.548 (±0.025)	0.595 (±0.047)	0.528 (±0.005)	0.527 (±0.006)
	0.314 (±0.059)	0.341 (±0.102)	0.319 (±0.106)	0.290 (±0.070)	0.403 (±0.121)	0.317 (±0.103)	0.453 (±0.036)	0.444 (±0.037)
	0.131 (±0.004)	0.133 (±0.003)	0.134 (±0)	0.134 (±0)	0.134 (±0)	0.134 (±0)	0.135 (±0)	0.134 (±0)
Breast Cancer Original	0.568 (±0.024)	0.566 (±0.027)	0.565 (±0.014)	0.570 (±0.027)	0.558 (±0.009)	0.559 (±0.010)	0.556 (±0.012)	0.555 (±0.010)
	0.375 (±0.101)	0.364 (±0.107)	0.441 (±0.106)	0.399 (±0.133)	0.430 (±0.085)	0.442 (±0.076)	0.407 (±0.078)	0.409 (±0.087)
	0.154 (±0.039)	0.139 (±0.006)	0.157 (±0.044)	0.157 (±0.043)	0.148 (±0.016)	0.146 (±0.010)	0.142 (±0.002)	0.156 (±0.042)
Heart Disease	0.376 (±0.054)	0.401 (±0.057)	0.372 (±0.047)	0.407 (±0.050)	0.397 (±0.042)	0.416 (±0.027)	0.415 (±0.026)	0.408 (±0.033)
	0.237 (±0.092)	0.238 (±0.090)	0.221 (±0.056)	0.278 (±0.099)	0.261 (±0.064)	0.273 (±0.070)	0.272 (±0.070)	0.289 (±0.076)
	0.109 (±0.037)	0.101 (±0.011)	0.139 (±0.058)	0.114 (±0.030)	0.118 (±0.033)	0.118 (±0.017)	0.121 (±0.031)	0.120 (±0.032)
Hepatitis	0.405 (±0.146)	0.547 (±0.202)	0.470 (±0.132)	0.629 (±0.213)	0.551 (±0.24)	0.572 (±0.240)	0.625 (±0.230)	0.562 (±0.233)
	0.25 (±0.139)	0.306 (±0.129)	0.279 (±0.172)	0.412 (±0.131)	0.327 (±0.178)	0.413 (±0.138)	0.346 (±0.162)	0.277 (±0.187)
	0.174 (±0.142)	0.188 (±0.137)	0.202 (±0.177)	0.339 (±0.157)	0.251 (±0.196)	0.269 (±0.190)	0.342 (±0.154)	0.277 (±0.187)

TABLE A.2 – Résumé des résultats numériques de la comparaison de 8 heuristiques TREERANK sur les données benchmark. Ce Tableau indique les performances de chaque procédure en termes d'ASC partielle pour les proportions $u_0 \in \{0.2, 0.1, 0.05\}$ respectivement dans cet ordre, leurs écart-types étant indiqués entre parenthèses.

	TreeRank $RF3_{CART}$	RankBoost	TreeRank $RF1_{SVM}$	SVM^{rank}	RLScore
Simulations					
GaussEasy20d	0.739 ($\pm$0.017)	0.747 ($\pm$0.017)	0.786 ($\pm$0.022)	0.724 ($\pm$0.024)	0.739 ($\pm$0.022)
GaussMed20d	0.681 ($\pm$0.025)	0.701 ($\pm$0.030)	0.738 ($\pm$0.026)	0.705 ($\pm$0.028)	0.713 ($\pm$0.028)
GaussHard20d	0.590 ($\pm$0.032)	0.632 ($\pm$0.038)	0.650 ($\pm$0.048)	0.617 ($\pm$0.044)	0.714 ($\pm$0.025)
Données *benchmark* (UCI)					
Congressional Vote	0.992 ($\pm$0.015)	0.937 ($\pm$0.049)	0.987 ($\pm$0.024)	0.935 ($\pm$0.048)	0.934 ($\pm$0.048)
Australian Credit	0.934 ($\pm$0.031)	0.937 ($\pm$0.023)	0.926 ($\pm$0.031)	0.92 ($\pm$0.028)	0.929 ($\pm$0.036)
German Credit	0.769 ($\pm$0.029)	0.781 ($\pm$0.030)	0.793 ($\pm$0.022)	0.737 ($\pm$0.036)	0.775 ($\pm$0.024)
Japanese Credit	0.931 ($\pm$0.042)	0.930 ($\pm$0.040)	0.923 ($\pm$0.038)	0.904 ($\pm$0.046)	0.910 ($\pm$0.047)
Autos MPG	0.972 ($\pm$0.016)	0.955 ($\pm$0.012)	0.978 ($\pm$0.016)	0.95 ($\pm$0.014)	0.942 ($\pm$0.025)
Ionosphere	0.968 ($\pm$0.032)	0.933 ($\pm$0.013)	0.990 ($\pm$0.013)	0.872 ($\pm$0.081)	0.875 ($\pm$0.055)
Breast Cancer Diagnosis	0.988 ($\pm$0.007)	0.982 ($\pm$0.004)	0.994 ($\pm$0.008)	0.98 ($\pm$0.005)	0.982 ($\pm$0.008)
Breast Cancer Original	0.993 ($\pm$0.006)	0.983 ($\pm$0.006)	0.995 ($\pm$0.006)	0.983 ($\pm$0.003)	0.984 ($\pm$0.004)
Heart Disease	0.911 ($\pm$0.034)	0.885 ($\pm$0.037)	0.917 ($\pm$0.032)	0.883 ($\pm$0.034)	0.891 ($\pm$0.032)
Hepatitis	0.868 ($\pm$0.162)	0.798 ($\pm$0.168)	0.864 ($\pm$0.139)	0.812 ($\pm$0.158)	0.803 ($\pm$0.166)

Table A.3 – Résumé des résultats numériques de la comparaison de 5 heuristiques de scoring. Ce Tableau indique la performance $\overline{\mathrm{ASC}}^{(t)}$ de chaque procédure d'apprentissage, son écart-type $\hat{\sigma}^2$ étant indiqué entre parenthèses.

	TreeRank $RF3_{CART}$	RankBoost	TreeRank $RF1_{SVM}$	SVM^{rank}	RLScore
Simulations					
GaussEasy20d	0.291 (±0.024) 0.155 (±0.020) 0.082 (±0.012)	0.291 (±0.025) 0.146 (±0.019) 0.077 (±0.015)	0.311 (±0.020) 0.163 (±0.018) 0.083 (±0.010)	0.279 (±0.021) 0.142 (±0.013) 0.073 (±0.009)	0.281 (±0.015) 0.144 (±0.010) 0.072 (±0.013)
GaussMed20d	0.270 (±0.030) 0.148 (±0.023) 0.082 (±0.011)	0.284 (±0.032) 0.159 (±0.020) 0.079 (±0.013)	0.302 (±0.021) 0.166 (±0.014) 0.088 (±0.008)	0.275 (±0.029) 0.149 (±0.015) 0.076 (±0.013)	0.271 (±0.028) 0.149 (±0.017) 0.076 (±0.011)
GaussHard20d	0.222 (±0.029) 0.128 (±0.015) 0.069 (±0.012)	0.245 (±0.025) 0.132 (±0.015) 0.066 (±0.012)	0.252 (±0.030) 0.132 (±0.016) 0.073 (±0.010)	0.229 (±0.022) 0.123 (±0.012) 0.068 (±0.008)	0.277 (±0.031) 0.146 (±0.015) 0.074 (±0.013)
Données *benchmark* (UCI)					
Congressional Vote	0.495 (±0.059) 0.201 (±0.042) 0.094 (±0)	0.333 (±0.025) 0.169 (±0.012) 0.084 (±0.006)	0.477 (±0.054) 0.201 (±0.042) 0.094 (±0)	0.333 (±0.025) 0.17 (±0.013) 0.084 (±0.006)	0.332 (±0.025) 0.168 (±0.012) 0.084 (±0.007)
Australian Credit	0.429 (±0.014) 0.273 (±0.059) 0.122 (±0.031)	0.412 (±0.014) 0.206 (±0.013) 0.103 (±0.011)	0.425 (±0.012) 0.248 (±0.039) 0.111 (±0.002)	0.404 (±0.024) 0.204 (±0.013) 0.103 (±0.010)	0.405 (±0.024) 0.199 (±0.014) 0.096 (±0.013)
German Credit	0.260 (±0.020) 0.144 (±0.022) 0.083 (±0.029)	0.254 (±0.016) 0.130 (±0.010) 0.067 (±0.005)	0.266 (±0.012) 0.148 (±0.020) 0.081 (±0.021)	0.254 (±0.012) 0.130 (±0.004) 0.065 (±0.005)	0.257 (±0.013) 0.13 (±0.008) 0.065 (±0.003)
Japanese Credit	0.417 (±0.023) 0.260 (±0.073) 0.141 (±0.047)	0.395 (±0.025) 0.199 (±0.017) 0.095 (±0.010)	0.414 (±0.020) 0.256 (±0.068) 0.125 (±0.032)	0.38 (±0.043) 0.188 (±0.023) 0.094 (±0.017)	0.383 (±0.053) 0.188 (±0.026) 0.091 (±0.018)
Autos MPG	0.421 (±0.016) 0.240 (±0.08) 0.095 (±0)	0.354 (±0.005) 0.178 (±0.006) 0.09 (±0.006)	0.433 (±0.058) 0.238 (±0.079) 0.095 (±0)	0.353 (±0.008) 0.177 (±0.005) 0.09 (±0.003)	0.350 (±0.011) 0.177 (±0.006) 0.089 (±0.006)
Ionosphere	0.392 (±0.086) 0.178 (±0.044) 0.078 (±0)	0.288 (±0.005) 0.144 (±0.003) 0.072 (±0.003)	0.494 (±0.062) 0.156 (±0) 0.078 (±0)	0.263 (±0.044) 0.131 (±0.024) 0.065 (±0.014)	0.263 (±0.029) 0.126 (±0.023) 0.062 (±0.016)

	TreeRank $RF3_{CART}$	RankBoost	TreeRank $RF1_{SVM}$	SVM^{rank}	RLScore
Breast Cancer Diagnosis	0.528 (±0.005) 0.453 (±0.036) 0.135 (±0)	0.507 (±0.012) 0.253 (±0.007) 0.125 (±0.008)	0.595 (±0.047) 0.317 (±0.103) 0.134 (±0)	0.502 (±0.013) 0.253 (±0.006) 0.126 (±0.010)	0.506 (±0.013) 0.253 (±0.012) 0.130 (±0.010)
Breast Cancer Original	0.556 (±0.012) 0.407 (±0.078) 0.142 (±0.002)	0.534 (±0.018) 0.265 (±0.012) 0.132 (±0.014)	0.559 (±0.010) 0.442 (±0.076) 0.146 (±0.010)	0.537 (±0.017) 0.271 (±0.009) 0.137 (±0.012)	0.537 (±0.010) 0.265 (±0.008) 0.134 (±0.011)
Heart Disease	0.415 (±0.026) 0.272 (±0.070) 0.121 (±0.031)	0.361 (±0.041) 0.176 (±0.027) 0.089 (±0.017)	0.416 (±0.027) 0.273 (±0.070) 0.118 (±0.017)	0.371 (±0.035) 0.188 (±0.022) 0.094 (±0.011)	0.371 (±0.033) 0.187 (±0.020) 0.094 (±0.01)
Hepatitis	0.625 (±0.230) 0.346 (±0.162) 0.342 (±0.154)	0.504 (±0.225) 0.263 (±0.115) 0.133 (±0.057)	0.572 (±0.240) 0.413 (±0.138) 0.269 (±0.190)	0.526 (±0.248) 0.272 (±0.125) 0.137 (±0.062)	0.543 (±0.231) 0.280 (±0.116) 0.141 (±0.057)

Table A.4 – Résumé des résultats numériques de la comparaison de 5 heuristiques de scoring. Ce Tableau indique les performances de chaque procédure en termes d'ASC partielle pour les proportions $u_0 \in \{0.2, 0.1, 0.05\}$ respectivement dans cet ordre, leurs écart-types étant indiqués entre parenthèses.

Bibliographie

[Agarwal *et al.* 2005] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled et D. Roth. *Generalization bounds for the area under the ROC curve.* Journal of Machine Learning Research, vol. 6, pages 393–425, 2005.

[Allen 1974] D.M. Allen. *The relationship between variable selection and data augmentation and a method for prediction.* Technometrics, vol. 16, pages 125–127, 1974.

[Amit & Geman 1997] Y. Amit et D. Geman. *Shape quantization and recognition with randomized trees.* Neural Computation, vol. 9, pages 1545–1588, 1997.

[Anderson *et al.* 1994] N. Anderson, P. Hall et D. Titterington. *Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates.* Journal of Multivariate Analysis, vol. 50, pages 41–54, 1994.

[Ansaldi 2002] N. Ansaldi. *Contribution des méthodes statistiques à la quantification de l'agrément de conduite.* PhD thesis, Université de Marne La Vallée, 2002.

[Arlot 2007] S. Arlot. *Rééchantillonnage et Sélection de modèles.* PhD thesis, Université Paris-Sud - Faculté des Sciences d'Orsay, 2007.

[Arrow 1951] K. Arrow. Social choice and individual values. John Wiley, 1951.

[Bach *et al.* 2005] F. Bach, D. Heckerman et E. Horwitz. *On the path to an ideal ROC curve : considering cost asymmetry in learning classifiers.* In Tenth International Workshop on Artificial Intelligence and Statistics (Aistats), 2005.

[Bach *et al.* 2006] F. Bach, D. Heckerman et E. Horvitz. *Considering cost asymmetry in learning classifiers.* Journal of Machine Learning Research, vol. 7, pages 1713–1741, 2006.

[Bansal & Fernández-Baca 2009] M. S. Bansal et D. Fernández-Baca. *Computing distances between partial rankings.* Information Processing Letters, vol. 109, no. 4, pages 238–241, 2009.

[Barreno *et al.* 2007] M. Barreno, A.A. Cardenas et J.D. Tygar. *Optimal ROC curve for a combination of classifiers.* In Proccedings of the conference on Neural Information and System Processing (NIPS), 2007.

[Barthélémy & Montjardet 1981] J.P. Barthélémy et B. Montjardet. *The median procedure in cluster analysis and social choice theory.* Mathematical Social Sciences, vol. 1, pages 235–26, 1981.

[Bartlett *et al.* 2002] P.L. Bartlett, S. Boucheron et G. Lugosi. *Model selection and error estimation.* Machine Learning, vol. 48, pages 85–113, 2002.

[Bauer & Kohavi 1998] E. Bauer et R. Kohavi. *An empirical comparison of voting classification algorithms : bagging, boosting and variants.* Machine Learning, pages 1–38, 1998.

[Bertail *et al.* 2008] P. Bertail, S. Clémençon et N. Vayatis. *On bootstrapping the ROC curve*. In Proceedings of the conference on Neural Iinformation and Processing Systems (NIPS), 2008.

[Betzler *et al.* 2008] N. Betzler, M.R. Fellows, J. Guo, R. Niedermeier et F.A. Rosamond. *Computing Kemeny rankings, parameterized by the average KT-distance*. In Proceedings of the 2nd International Workshop on Computational Social Choice, 2008.

[Biau & L.Gyorfi 2005] G. Biau et L.Gyorfi. *On the asymptotic propoerties of a nonparametric l_1-test statistic of homogeneity*. IEEE Transactions on Information Theory, vol. 51(11), pages 3965–3973, 2005.

[Biau *et al.* 2008] G. Biau, L. Devroye et G. Lugosi. *Consistency of Random Forests*. Journal of Machine Learning Research, vol. 9, pages 2039–2057, 2008.

[Bickel 1969] P. Bickel. *A distribution free version of the Smirnov two sample test in the p-variate case*. The Annals of Mathematical Statistics, vol. 40(1), pages 1–23, 1969.

[Birgé & Massart 2006] L. Birgé et P. Massart. *Minimal penalties for gaussian model selection*. Probabilistic Theory Related Fields, vol. 134(3), 2006.

[Borda 1781] J.C. Borda. *Mémoire sur les élections au scrutin*. In Histoire de l'Académie Royale des Sciences. 1781.

[Boucheron *et al.* 2005] S. Boucheron, O. Bousquet et G. Lugosi. *Theory of classification : a survey of recent advances*. ESAIM Probabilities and Statistics, vol. 9, pages 323–375, 2005.

[Bradley *et al.* 1994] A.P. Bradley, B.C. Lovell, M. Ray et G. Hawson. *On the methodology for comparing learning algorithms : a case study*. In Proceedings of the Second Australian and New Zealand Conference on Intelligence Information Systems, pages 37–41, 1994.

[Bradley 1997] A.P. Bradley. *The use of the area under the ROC curve in the evaluation of machine learning algorithms*. Pattern Recognition, vol. 30 (7), pages 1145–1159, 1997.

[Breiman *et al.* 1984] L. Breiman, J. Friedman, R. Olshen et C. Stone. Classification and regression trees. Wadsworth and Brooks, 1984.

[Breiman 1996a] L. Breiman. *Arcing classifiers*. Rapport technique, Statistics Department, University of Berkeley, California, 1996.

[Breiman 1996b] L. Breiman. *Bagging Predictors*. Machine Learning, vol. 26, pages 123–140, 1996.

[Breiman 1996c] L. Breiman. *The heuristics of instability in model selection*. Annals of Statistics, vol. 24, pages 2350–2383, 1996.

[Breiman 1996d] L. Breiman. *Out-Of-Bag Estimation*. Rapport technique 14, Statistics Department, University of Berkeley, California, 1996.

[Breiman 2001] L. Breiman. *Random Forests*. Machine Learning, vol. 45, no. 1, pages 5–32, 2001.

[Burges *et al.* 2005] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton et G. Hullender. *Learning to Rank using Gradient Descent*. In Proceedings of the 22nd International Conference on Machine Learning, pages 89–96, 2005.

[Burges *et al.* 2006] C. Burges, R. Ragno et Q. Le. *Learning to rank with nonsmooth cost functions*. In Proceedings of NIPS, 2006.

[Chabat *et al.* 2002] A. Chabat, R. Goscinny et A. Uderzo. *Monologue d'Otis, sribe, -50 av. JC.* Astérix et Obélix : Mission Cléopâtre, 2002.

[Charon & Hudry 1998] I. Charon et O. Hudry. *Lamarckian genetic algorithms applied to the aggregation of preferences.* Annals of Operations Research, vol. 80, pages 281–297, 1998.

[Cheung & Klotz 1997] Y.K. Cheung et J.H. Klotz. *The Mann-Whitney-Wilcoxon distribution using linked list.* Statistica Sinica, vol. 7, page 805 :813, 1997.

[Clémençon & Vayatis 2007] S. Clémençon et N. Vayatis. *Ranking the best instances.* Journal of Machine Learning Research, vol. 8, pages 2671–2699, 2007.

[Clémençon & Vayatis 2008a] S. Clémençon et N. Vayatis. *Empirical performance maximization for linear rank statistis.* Neural Advances Processing Systems, 2008.

[Clémençon & Vayatis 2008b] S. Clémençon et N. Vayatis. *Overlaying classifiers : a practical approach for optimal ranking.* In In NIPS'08 : Proceedings of the 2008 conference on advances in neural information processing systems, 2008.

[Clémençon & Vayatis 2008c] S. Clémençon et N. Vayatis. *Tree-strutured ranking rules and approximation of the optimal ROC curve.* In In ALT'08 : Proceedings of the 2008 conference on algorithmic learning theory, 2008.

[Clémençon & Vayatis 2009a] S. Clémençon et N. Vayatis. *Adaptive estimation of the optimal ROC curve and a bipartite ranking algorithm.* In Lecture Notes in Computer Science. Springer Berlin, 2009.

[Clémençon & Vayatis 2009b] S. Clémençon et N. Vayatis. *Nonparametric estimation of the precision-recall curve.* In Proceedings of the 26th Annual International Conference on Machine Learning (ICML), 2009.

[Clémençon & Vayatis 2009c] S. Clémençon et N. Vayatis. *On partitioning rules for bipartite ranking.* In Journal of Machine Learning Research : Proceedings of Aistat'09, 2009.

[Clémençon & Vayatis 2009d] S. Clémençon et N. Vayatis. *Tree-based ranking methods.* IEEE Transactions on Information Theory, vol. 55(9), pages 4316–4336, 2009.

[Clémençon *et al.* 2005a] S. Clémençon, G. Lugosi et N.Vayatis. *From Ranking to Classification : a Statistical View.* In Proceedings of the 29th Annual Conference of the German Classification Society (GfKl), 2005.

[Clémençon *et al.* 2005b] S. Clémençon, G. Lugosi et N. Vayatis. *Ranking and scoring using empirical risk minimization.* In Lecture notes in computer science, Proceedings of COLT, 2005.

[Clémençon *et al.* 2008] S. Clémençon, G. Lugosi et N. Vayatis. *Ranking and empirical minimization of U-statistics.* The Annals of Statistics, vol. 36(2), pages 844–874, 2008.

[Clémençon *et al.* 2010] S. Clémençon, M. Depecker et N. Vayatis. *Adaptive Partitioning Schemes for Bipartite Ranking : How to Grow and Prune a Ranking Tree.* Journal of Machine Learning (Accepted for publication), 2010.

[Cohen *et al.* 1999] W. Cohen, R. Schapire et Y. Singer. *Learning to order things.* Journal of Artificial Intelligence Research, vol. 10, pages 213–270, 1999.

[Condorcet 1785] M.J. Condorcet. Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Imprimerie Royale, Paris, 1785.

[Conitzer 2006] V. Conitzer. *Improved Bounds for Computing Kemeny Rankings.* In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), 2006.

[Cornec 2009] M. Cornec. *Inégalités probabilistes pour l'estimateur de validation croisée dans le cadre de l'apprentissage statistique et modèles statistiques appliqués à l'économie et à la finance*. PhD thesis, Université Paris Nanterre, 2009.

[Cortes & Mohri 2004] C. Cortes et M. Mohri. *AUC optimization vs. error rate minimization*. In Advances in Neural Information Processing Systems, 2004.

[Cortes & Mohri 2005] C. Cortes et M. Mohri. *Confidence intervals for the area under the ROC curve*. In Advances in Neural Information and Processing Systems, 2005.

[Cortes & Vapnik 1995] C. Cortes et V. Vapnik. *Support-Vector Networks*. Machine Learning, vol. 20, 1995.

[Critchlow 1985] D.E. Critchlow. Metric methods for analyzing partially ranked data. 1985.

[Davis & Goadrich 2006] J. Davis et M. Goadrich. *The relationship between Precision-Recall and ROC curves*. In Proceedings of the 23rd international conference on Machine learning, 2006.

[Devore & Lorentz 1993] R. Devore et G. Lorentz. Constructive approximation. Springer, 1993.

[Deza & Deza 2009] M. Deza et E. Deza. Encyclopedia of distances. Springer, 2009.

[Diaconis 1988] P. Diaconis. *Groupe representation in probability and statistics*. In IMS Lecture Series 11. IMS, 1988.

[Diaz-Uriarte & de Andrés 2006] R. Diaz-Uriarte et S. Alvarez de Andrés. *Gene selection and classification of microarray data using random forest*. BMC Bioinformatics, vol. 7(3), 2006.

[Dietterich & Bakiri 1991] T. Dietterich et G. Bakiri. *Error-correcting output codes : A general method for improving multiclass inductive learning programs*. In Proceedings of the Ninth National Conference on Artificial Intelligence, 1991.

[Dietterich 1998] T. Dietterich. *An experimental comparison of three methods for constructing ensembles of decision trees : bagging, boosting and randomization*. Machine Learning, pages 1–22, 1998.

[Dodd & Pepe 2003] L.E. Dodd et M. S. Pepe. *Partial AUC estimation and regression*. Biometrics, vol. 59(3), pages 614–623, 2003.

[Dwork *et al.* 2001] C. Dwork, R. Kumar, M. Naor et D. Sivakumar. *Rank aggregation methods for the Web*. In Proceedings of the 10th International WWW conference, pages 613–622. IEEE Publications, 2001.

[Efron & Tisbshirani 1997] B. Efron et R. Tisbshirani. *Improvements on cross-validation : the .632+ bootstrap method*. Journal of the American Statistical Association, vol. 92(438), pages 548–560, 1997.

[Efron *et al.* 2004] B. Efron, T. Hastie, I. Johnstone et R. Tibshirani. *Least Angle Regression*. Annals of Statistics, vol. 32(2), pages 407–499, 2004.

[Efron 1979] B. Efron. *Bootstrap methods : another look at jackknife*. Annals of Statistics, vol. 7(1), pages 1–26, 1979.

[Efron 1983] B. Efron. *Estimating the error rate of a prediction rule : improvement on cross-validation*. Journal of the American Statistical Association, vol. 78(382), pages 316–331, 1983.

[Efron 1986] B. Efron. *How biased is the apparent error rate of a prediction rule ?* Journal of the American Statistical Association, vol. 81(394), pages 461–470, 1986.

[Egan 1975] J. Egan. Signal detection theory and COR analysis. Academic Press, 1975.

[Fagin *et al.* 2003] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar et E. Vee. *Comparing and aggregating rankings with ties.* In Proceedings of the 12-th WWW conference, pages 366–375, 2003.

[Fagin *et al.* 2006] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar et E. Vee. *Comparing partial rankings.* SIAM J. Discrete Mathematics, vol. 20(3), pages 628–648, 2006.

[Fawcett 2006] T. Fawcett. *An introduction to ROC analysis.* Pattern Recognition Letters, vol. 27, pages 861–874, 2006.

[Ferri *et al.* 2002] C. Ferri, P.A. Flach et J. Hernàndez-Orallo. *Learning Decision Trees Using the Area Under the ROC Curve.* In Proceedings of the Nineteenth International Conference on Machine Learning, 2002.

[Fishburn 1973] P. Fishburn. The theory of social choice. University Press, Princeton, 1973.

[Fisher 1936] R. Fisher. *The use of multiple measurements in taxonomic problems.* Annals of Eugenics, vol. 7, pages 179–188, 1936.

[Freund & Schapire 1999] Y. Freund et R.E. Schapire. *A Short Introduction to Boosting.* Journal of Japanese Society for Artificial Intelligence, vol. 14, no. 5, pages 771–782, 1999.

[Freund *et al.* 2003] Y. Freund, R.Iyer, R.E. Schapire et Y. Singer. *An Efficient Boosting Algorithm for Combining Preferences.* Journal of Machine Learning Research, vol. 4, pages 933–969, November 2003.

[Friedman & Rafsky 1979] J. Friedman et L. Rafsky. *Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests.* The Annals of Statistics, vol. 7(4), pages 697–717, 1979.

[Friedman *et al.* 1998] J. Friedman, T. Hastie et R. Tibshirani. *Additive Logistic Regression : a Statistical View of Boosting.* Annals of Statistics, vol. 28(2), pages 337–407, 1998.

[Friedman 2001] J. Friedman. *Greedy function approximation : a gradient boosting machine.* Annals of Statistics, vol. 6, pages 393–425, 2001.

[Fromont 2007] M. Fromont. *Model selection by bootstrap penalization for classification.* Journal of Machine Learning, vol. 66(23), page 165 :207, 2007.

[Geisser 1975] S. Geisser. *The predictive sample reuse method with applications.* Journal of the American Statistical Association, vol. 70, pages 320–328, 1975.

[Germain 2007a] J.F. Germain. *Pampering the client : clalibrating vehicle parts to satisfy customers.* Proceedings of BIGS, vol. 1, pages 164–172, 2007.

[Germain 2007b] J.F. Germain. *A two-steps model selection procedure based on the regularization path of a L_1-penalized logistic likelihood.* In Proceedings of SFDS, 2007.

[Germain 2009] J.F. Germain. *Sélection de modèles via les chemins de régularisation pour l'objectivation mono et multi-prestations. Application à l'agrément de conduite.* PhD thesis, Télécom ParisTech, 2009.

[Gey 2002] S. Gey. *Bornes de Risque, Détection de Ruptures, Boosting : Trois Thèmes Statistiques autour de CART en Régression.* PhD thesis, Université Paris 11, Orsay, 2002.

[Ghattas 2000] B. Ghattas. *Agrégation d'arbres de décision binaires. Application à la prévision de l'ozone dans les Bouches du Rhône.* PhD thesis, Université de la Méditerranée, 2000.

[Giné & Zinn 1984] E. Giné et J. Zinn. *Some limit theorems for empirical processes*. Annals of Probability, vol. 12(4), pages 929–998, 1984.

[Green & Swets 1974] D.M. Green et J.A. Swets. Signal detection theory and psychophysics. Wiley, 1966, 1974.

[Gretton *et al.* 2008a] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf et A. Smola. *A kernel method for the two-sample problem*. CoRR, vol. abs/0805.2368, 2008.

[Gretton *et al.* 2008b] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf et A. Smola. *A kernel method for the two-sample problem*. Rapport technique, Max Planck Institut für biologische Kybernetik, 2008.

[Gyorfi *et al.* 2002] L. Gyorfi, M. Kohler, A. Krzyzak et H. Walk. A distribution free theory of nonparametric regression. 2002.

[Hajek & Sidak 1967] J. Hajek et Z. Sidak. Theory of rank tests. Academic Press, 1967.

[Hajek 1962] J. Hajek. *Asymptotically Most Powerful Rank-Order Tests*. Annals of Mathematics and Statistics, vol. 33(3), pages 1124–1147, 1962.

[Hall & Tajvidi 2002] P. Hall et N. Tajvidi. *Permutation tests for equality of distributions in high-dimensional stettings*. Biometrika, vol. 89(2), pages 359–374, 2002.

[Hanley & McNeil 1982] J.A. Hanley et B.J. McNeil. *The Meaning and Use of the Area Under a ROC curve*. Radiology, vol. 143, pages 29–36, 1982.

[Harvey *et al.* 1992] L.O. Jr. Harvey, K.R. Hammond, C.M. Lusk et E.F. Mross. *Application of signal detection theory to weather forecasting behavior*. Monthly Weather Review, vol. 120, pages 863–883, 1992.

[Hastie & Park 2007] T. Hastie et M.Y. Park. $\mathcal{L}_1$ *Regularization Path Algorithm for Generalized Linear Models*. Journal of the Royal Statistical Society, vol. 69(4), pages 659–677, 2007.

[Hastie & Tibshirani 1990] T. Hastie et R. Tibshirani. Generalized additive models. Chapman & Hall/CRC, 1990.

[Hastie *et al.* 2001] T. Hastie, R. Tibshirani et J. Friedman. The elements of statistical learning. Springer Series in Statistics, 2001.

[Hastie *et al.* 2004] T. Hastie, S. Rosset, R. Tibshirani et J. Zhu. *The entire regularization path for the support vector machine*. Journal of Machine Learning Research, vol. 5, pages 1391–1415, 2004.

[Heath *et al.* 1993] D. Heath, S. Kasif et S. Salzberg. *k-dt : a multi-tree learning method*. In Proceedings of the Second InternationalWorkshop on Multistrategy Learning, 1993.

[Henze & Penrose 1999] N. Henze et M. Penrose. *On the multivariate runs test*. The Annals of Statistics, vol. 27(1), pages 290–298, 1999.

[Herbrich *et al.* 2000] R. Herbrich, T. Graepel et K. Obermayer. *Large margin rank boundaries for ordinal regression*. Advances in Large Margin Classifiers, pages 115–132, 2000.

[Hoeffding 1948] W. Hoeffding. *A class of statistics with asymptotically normal distribution*. Annals of Mathematics and Statistics, vol. 19, page 293 :325, 1948.

[Horvath *et al.* 2008] L. Horvath, Z. Horvath et Zhou. *Confidence bands for ROC curves*. Journal of Statistical Planning and Inference, vol. 138, pages 1894–1904, 2008.

[Howie 2000] J. Howie. *Hyperbolic groups*. Groups and Applications, pages 137–160, 2000.

[Hsieh & Turnbull 1996] D. Hsieh et B.W. Turnbull. *Nonparametric and semiparametric estimation of the receiver operating characteristic curve.* The Annals of Statistics, vol. 24, pages 25–40, 1996.

[Hudry 2004] O. Hudry. *Computation of median orders : complexity results.* Annales du LAMSADE : Vol. 3. Proceedings of the workshop on computer science and decision theory, DIMACS, vol. 163, pages 179–214, 2004.

[Hudry 2008] O. Hudry. *NP-hardness results for the aggregation of linear orders into median orders.* Annals of Operative Research, vol. 163, pages 63–88, 2008.

[Ilyas *et al.* 2002] I. Ilyas, W. Aref et A. Elmagarmid. *Joining ranked inputs in practice.* In Proceedings of the 28th International Conference on Very Large Databases, 2002.

[Jaakkola & Haussler 1999] T. Jaakkola et D. Haussler. *Probabilistic kernel regression models.* In Proceedings of the 7th international workshop on artificial intelligence and statistics, 1999.

[Joachims 2002a] T. Joachims. Learning to classify text using support vector machines. methods, theory, and algorithms. Kluwer Academic Publishers, 2002.

[Joachims 2002b] T. Joachims. *Optimizing Search Engines using Clickthrough Data.* In KDD'02 - Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 133–142, 2002.

[Järvelin & Kekäläinen 2000] K. Järvelin et J. Kekäläinen. *IR evaluation methods for retrieving highly relevant documents.* SIGIR, ACM, pages 41–48, 2000.

[Keerthi *et al.* 2002] S. Keerthi, K. Duan, S. Shevade et A. Poo. *A Fast Dual Algorithm for Kernel Logistic Regression.* Machine Learning, vol. 61, pages 151 – 165, 2002.

[Kemeny 1959] J. G. Kemeny. *Mathematics without numbers.* Daedalus, vol. 88, pages 571–591, 1959.

[Kendall 1945] M. Kendall. *The treatment of ties in ranking problems.* Biometrika, vol. 33, pages 239–251, 1945.

[Koltchinskii 2001] V. Koltchinskii. *Rademacher penalties and structural risk minimization.* IEEE Transactions on Information Theory, vol. 47(5), page 1902 :1914, 2001.

[Koltchinskii 2006] V. Koltchinskii. *Local Rademacher complexities and oracle inequalities in risk minimization.* Annals of Statistics, vol. 34(6), 2006.

[Kotz & Nadarajah 2004] S. Kotz et S. Nadarajah. Multivariate t-distributions and their applications. Cambridge University Press, Princeton, 2004.

[Kwok & Carter 1990] S. Kwok et C. Carter. *Multiple decision trees.* Uncertainty in Artificial Intelligence, vol. 4, pages 327–335, 1990.

[Laguna *et al.* 1999] M. Laguna, R. Marti et V. Campos. *Intensification and diversification with elite tabu search solutions for the linear ordering problem.* Computers and Operations Research, vol. 26(12), pages 1217–1230, 1999.

[L.Devroye *et al.* 1996] L.Devroye, L. Györfi et G. Lugosi. A probabilistic theory of pattern recognition. Springer, 1996.

[Lecué 2007] G. Lecué. *Méthodes d'agrégation : optimalité et vitesses rapides.* PhD thesis, LPMA, Université Paris VII, 2007.

[Lehman & Romano 2005] E.L. Lehman et J.P. Romano. Testing statistical hypotheses. Springer, 2005.

[Lehmann 2006] E.L. Lehmann. Nonparametrics : Statistical methods based on ranks. Springer, 2006.

[Lugosi & Nobel 1996] G. Lugosi et A. Nobel. *Consistency of data-driven histogram methods for density estimation and classification.* Annals of Statistics, vol. 24, pages 687–706, 1996.

[Lugosi & Zeger 1996] G. Lugosi et K. Zeger. *Concept learning using complexity regularization.* IEEE Transactions on Information Theory, vol. 42(1), page 48 :54, 1996.

[Maclin & Opitz 1997] R. Maclin et D. Opitz. *An empirical evaluation of bagging and boosting.* In Fourteenth National Conference on Artificial Intelligence, 1997.

[Macskassy & Provost 2004] S. Macskassy et F. Provost. *Confidence bands for ROC curves : methods and an empirical study.* In Proceedings of the first Workshop on ROC Analysis in AI, 2004.

[Macskassy *et al.* 2005] S. Macskassy, F. Provost et S. Rosset. *Bootstrapping the ROC curve : an empirical evaluation.* In Proceedings of ICML Workshop on ROC Analysis in Machine Learning, 2005.

[Mallat 1990] S. Mallat. A wavelet tour of signal processing. San Diego : Academic Press, 1990.

[Mammen & Tsybakov 1999] E. Mammen et A. Tsybakov. *Smooth discriminant analysis.* The Annals of Statistics, vol. 27(6), pages 1808–1829, 1999.

[Mandhani & Meila 2009] B. Mandhani et M. Meila. *Tractable Search for Learning Exponential Models of Rankings.* In Proceedings of AISTATS, 2009.

[Mann & Whitney 1947] H.B. Mann et D.R. Whitney. *On a test of whether one of two random variables is stochastically larger than the other.* Annals of Mathematics and Statistics, vol. 18, pages 50–60, 1947.

[Marden 1995] J.I. Marden. *Analyzing and modeling rank data.* In Monographs on statistics and applied probability. Chapman & Hall, 1995.

[Mason 1982] I. Mason. *A model for assessment of weather forecasts.* Australian Meteorological Magazine, vol. 30, pages 291–303, 1982.

[Massart 2007a] P. Massart. Ecole d'été de probabilités de saint-flour xxxiii, concentration inequalities and model selection. Springer, 2007.

[Massart 2007b] P. Massart. *Un point de vue non asymptotique pour la sélection de modèle.* Gazette des Mathématiques, vol. 58 (114), pages 5–31, 2007.

[Meila *et al.* 2007] M. Meila, K. Phadnis, A. Patterson et J. Bilmes. *Consensus ranking under the exponential model.* In Conference on Artificial Intelligence (UAI), 2007.

[Mielke & Berry 2001] P.W. Mielke et K.J. Berry. Permutation methods. Springer, 2001.

[Mika *et al.* 1999] S. Mika, G. Rätsch, J. Weston, B. Schölkopf et K.R. Müller. *Fisher discriminant analysis with kernels :.* In Neural Networks for Signal Processing, 1999.

[Moulines *et al.* 2008] E. Moulines, Z. Harchaoui et F. Bach. *Testing for homogeneity with kernel Fisher discriminant analysis.* In Advances in Neural Information Processing Systems, 2008.

[Mozer *et al.* 2001] M. Mozer, R. Dodier, M. colagrosso, C. Guerra-Salcedo et R. Wolniewicz. *Prodding the ROC curve : constrained optimization of classifier performance.* In Proceedings of the conference on Neural Information and System Processing (NIPS), 2001.

[Nemirovski 2000] A. Nemirovski. *Topics in non-parametric statistics.* In Volume 1738 of Lecture Notes in Mathematics. Springer, 2000.

[Neyman & Pearson 1933] J. Neyman et E.S. Pearson. *On the problem of the most efficient tests of statistical hypothesis*. In Philosophical Transactions o f the Royal Society of London, Series A, containing Papers of Mathematical and Physical Character. 1933.

[Nobel 2002] A. Nobel. *Analysis of a complexity-based pruning scheme for classification trees*. IEEE Transactions on Information Theory, vol. 48(8), pages 2362–2368, 2002.

[Obuchowski 2003] N.A. Obuchowski. *Receiver operating characteristic curves and their use in radiology*. Radiology, vol. 229 (1), pages 3–8, 2003.

[Pahikkala *et al.* 2007] T. Pahikkala, E. Tsivtsivadze, A. Airola, J. Boberg et T. Salakoski. *Learning to Rank with Pairwise Regularized Least-Squares*. In Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval, pages 27–33, 2007.

[Peña & Giné 1999] V. De La Peña et E. Giné. Decoupling : from dependence to independence. 1999.

[Pennock *et al.* 2000] D. Pennock, E. Horvitz et C. Giles. *Social choice theory and recommender systems : analysis of the foundations of collaborative filtering*. In Proceedings of the National Conference on Artificial Intelligence, 2000.

[Provost & Domingos 2002] F. Provost et P. Domingos. *Tree induction for probability-based ranking*. Machine Learning, vol. 52(3), pages 199–215, 2002.

[Provost & Fawcett 1997] F. Provost et T. Fawcett. *Analysis and visualization of classifier performance : comparison under imprecise class and cost distributions*. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), 1997.

[Quinlan 1986] J.R. Quinlan. *Induction of decision trees*. Machine Learning, vol. 1, pages 81–106, 1986.

[Rachev 1991] A. Rachev. Probability metrics and the statbility of stochastic models. Wiley, 1991.

[Rakotomamonjy 2004] A. Rakotomamonjy. *Optimizing area under roc curve with svms*. In Proceedings of the first Workshop on ROC Analysis in AI, 2004.

[Rudin 2006] C. Rudin. *Ranking with a p-norm push*. In Proceedings of the Nineteenth Annual Conference on Learning Theory, 2006.

[Schölkopf & Smola 2002] B. Schölkopf et J. Smola. Learning with kernels : Support vector machines, regularization, optimization and beyond. MIT Press, 2002.

[Schwarz 1978] G. Schwarz. *Estimation the Dimension of a Model*. Annal of Statistics, vol. 6, pages 461–464, 1978.

[Serfling 1980] R. Serfling. Approximation theorems of mathematical statistics. John Wiley and Sons, 1980.

[Spackman 1989] K.A. Spackman. *Signal detection theory : valuable tools for evaluating inductive learning*. In Proceedings of the Sixth International Workshop on Machine Learning, Morgan Kaufman, San Mateo, 1989.

[Spall 2003] J.C. Spall. Introduction to stochastic search and optimization : Estimation, simulation, and control. John Wiley and Sons, 2003.

[Stone 1974] M. Stone. *Cross-validatory choice and assessment of statistical predictions*. Journal of the Royal Statistical Society : Series B, vol. 36, pages 111–147, 1974.

[Svetnik *et al.* 2003] V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan et B. Feuston. *Random Forest : A Classification and Regression Tool for Compound Classification and QSAR Modeling.* Journal of Chemical Information and Computer Sciences, vol. 43(6), pages 1947–1958, 2003.

[Swets 1979] J.A. Swets. *ROC analysis applied to the evaluation of medical imaging techniques.* Invest. Radiology, vol. 14, pages 109–121, 1979.

[Taylor & Cristianini 2000] J. Taylor et N Cristianini. Support vector machines and other kernel-based learning methods. Cambridge University Press, 2000.

[Tibshirani 1996] R. Tibshirani. *Regression shrinkage and selection via the lasso.* Journal of th Royal Statistical Society, vol. B 58, pages 229–243, 1996.

[Truchon 1998] M. Truchon. *An extension of the Condorcet criterion and Kemeny orders.* In Cahier 98-15 du Centre de Recherche en Economie et Finance Appliquées. 1998.

[Tsai *et al.* 2007] M.F. Tsai, T.Y. Liu, T. Qin, H.H. Chen et W.Y. Ma. *Frank : a ranking method with fidelity loss.* In Proceedings of the 30th annual international ACM SIGIR conference, 2007.

[Tsybakov 2004] A. Tsybakov. *Optimal aggregation of classifiers in statistical learning.* The Annals of Statistics, vol. 33, pages 135–166, 2004.

[Usunier & ans P. Gallinari 2005] N. Usunier et M. Amini ans P. Gallinari. *A data-dependant generalisation error bound for the AUC.* In ROCML ICML 2005 Workshop, 2005.

[van der Vaart 1998] A. W. van der Vaart. Asymptotics statistics. Cambridge Series in Statistical and Probabilistic Mathematics, 1998.

[van Trees 1968] H.L. van Trees. Detection, estimation, and modulation theory, part i. John Wiley, 1968.

[Vapnik & Chervonenkis 1974] V. Vapnik et A. Chervonenkis. Theory of pattern recognition. 1974.

[Vapnik *et al.* 1992] V. Vapnik, B. Boser et I. Guyon. *A training algorithm for optimal margin classifiers.* In Fifth Annual Workshop on Computational Learning Theory, 1992.

[Vapnik 1982] V.N. Vapnik. Estimation of dependences based on empirical data. Springer Series in Statistics, 1982.

[Vapnik 1996] V. Vapnik. The nature of statistical learning theory. 1996.

[Vapnik 1998] V. Vapnik. Statistical learning theory. 1998.

[Wakabayashi 1998] Y. Wakabayashi. *The complexity of computing medians of relations.* Resenhas, vol. 3(3), pages 323–349, 1998.

[Wilcoxon 1945] F. Wilcoxon. *Individual comparisons by ranking methods.* Biometrics, vol. 1, page 80 :83, 1945.

[Xu & Li 2007] J. Xu et H. Li. *AdaRank : a boosting algorithm for information retrieval.* In Proceedings of SIGIR'07, 2007.

[Yan *et al.* 2003] L. Yan, R. Dodier, M. Mozer et R. Wolniewicz. *Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney.* In Proceedings of the 20th International Conference on Machine Learning (ICML), 2003.

[Yates & Ribeiro-Neto 1999] R.B. Yates et B. Ribeiro-Neto. Modern information retrieval. Addison Wesley, 1999.

[Young 1974] H.P. Young. *An axiomatization of Borda's rule :.* Journal of Economic Theory, vol. 9, pages 43–52, 1974.

[Yue & Finley 2007] Y. Yue et T. Finley. *A support vector method for optimizing average precision.* In Proceedings of SIGIR'07, 2007.

[Zhu & Hastie 2001] J. Zhu et T. Hastie. *Kernel logistic regression and the import vector machine.* In Proceedings of NIPS, 2001.