Inference and evaluation of the multinomial mixture model for unsupervised text clustering

Loïs Rigouste

TELECOM PARIS
école nationale supérieure des télécommunications

École Doctorale d'Informatique, Télécommunications et Électronique de Paris

Thèse

présentée pour obtenir le grade de Docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité : **Signal et Images**

Loïs RIGOUSTE

Méthodes probabilistes pour l'analyse exploratoire de données textuelles

Soutenue le 7 novembre 2006 devant le jury composé de :

Ludovic Lebart	Président
Eric Gaussier	Rapporteurs
Michèle Sebag	
Fabrice Clérot	Examinateurs
Christian Robert	
Olivier Cappé	Directeurs de thèse
François Yvon	

« C'est une bien triste chose qu'il y ait de nos jours si peu d'informations inutiles. »

Oscar Wilde – A Few Maxims for the Instruction of the Over-Educated

« Mais rangez un peu ! Avec le prix qu'on paye, quand même, le minimum c'est qu'il y ait un peu de ménage qui soit fait ! »

Alain Chabat – La Cité de la Peur

Remerciements

Même si l'énumération pourra paraître rébarbative au lecteur non familier, j'ai trouvé que deux pages de remerciements, c'était très court au regard du nombre d'échanges fructueux dont j'ai eu la chance de bénéficier pendant cette période de trois ans. Soyons honnête, ça l'est aussi au regard des 180 pages plus formelles qui suivent[1].

Je remercie Ludovic Lebart d'avoir accepté de présider le jury de thèse. J'ai également apprécié les contacts que nous avons eus lors de la conférence ASMDA 2005 à Brest. Merci à Eric Gaussier et Michèle Sebag d'avoir tous deux conduit la tâche de rapporteur avec beaucoup d'application, suggérant nombre d'améliorations et donnant un éclairage et une interprétation originaux de plusieurs aspects de cette thèse. Les travaux d'Eric Gaussier, dont certains sont proches des thématiques développées ici, ont particulièrement retenu notre intérêt. Merci à lui d'avoir ainsi inspiré diverses pistes de recherche explorées dans ce document. Par ailleurs, Michèle Sebag nous a fait l'honneur d'accepter de présenter la première séance du séminaire d'apprentissage de l'ENST, qu'un groupe de doctorants auquel j'appartenais a initié en 2005. Qu'elle soit remerciée de sa confiance. Je remercie Christian Robert d'avoir également participé à ce jury. Cela a été pour moi un plaisir et une expérience enrichissante de travailler avec lui lors de la traduction de son livre *Le Choix Bayésien*. Enfin, j'adresse toute ma gratitude à Fabrice Clérot pour son implication dans mon doctorat. Ses conseils et les nombreuses discussions particulièrement riches et variées que nous avons eues font officieusement de lui un troisième directeur de thèse, qui a admirablement et agréablement guidé mes travaux de recherche, sans pour autant les contraindre de quelque manière que ce soit.

La croyance populaire veut qu'un doctorat soit généralement une longue période de travail solitaire essentiellement déprimant, dans un bureau sinistre et avec pour seuls compagnons un ordinateur et plusieurs piles de livres. Mon expérience a été opposée en tout point à ce tableau peu reluisant et j'estime que c'est en très grande partie grâce à mes deux directeurs de thèse : Olivier Cappé et François Yvon. Ce fut un réel plaisir de travailler avec eux pendant ces trois ans, de bénéficier de leur expérience et de leurs conseils. Pendant ces quelques années dans le monde de la recherche académique, j'ai pu constater combien le degré d'implication d'un directeur de thèse dans les travaux du doctorant variait d'une thèse à une autre. À cet égard, la participation d'Olivier et François, en particulier dans les tâches les plus concrètes de programmation et de rédaction d'articles, a été extraordinaire. Je conviens que ce compliment, placé en cette page, est sans grande originalité. Pour l'ancrer dans le concret, qu'il me soit donc permis de l'illustrer ainsi :

- En février 2005, à la suite d'incidents de santé mineurs, j'ai dû m'absenter de l'ENST pour une dizaine de jours. Olivier a repris les résultats de mes expériences et a jeté en ces quelques jours toutes les bases de l'algorithme d'inférence itérative présenté dans cette thèse. Le retour au travail a été beaucoup plus facile dans de telles conditions !
- Un mois plus tard, alors qu'une date de retour de résultats pour la participation au DÉfi Fouille de Textes (DEFT) se trouvait placée pendant mes congés, François a repris les programmes que j'avais écrits et les a ajouté aux siens pour me remplacer au pied levé dans le rôle d'expérimentateur et de correspondant que je m'étais engagé à remplir. Plus tard, il a écrit le module implémentant LDA sur lequel se sont construites une grande partie des expériences présentées dans le dernier chapitre.

Qu'ils soient chaleureusement remerciés de leur dévouement peu commun !

[1] Non, s'il-vous-plait, ne refermez pas la thèse tout de suite, il n'y a pas que des calculs, il y a aussi quelques illustrations !

Dans cette liste de remerciements, mes collègues de bureau méritent aussi une place de choix. Sans minimiser la richesse de nos échanges scientifiques, je dois bien dire qu'ils ont été une source inépuisable de rigolades, que ce soit Thomas (ah, les dimanches au boulot, les sushis, les Fatals Picards et Didier Super) ou Zaïd (ah, les dimanches au boulot, les sushis, Madonna et Jean-Jacques Goldman[2]). Et un mot aussi pour cet autre colocataire du bureau DA312, peut-être un peu moins drôle mais d'une fidélité et d'un calme toujours irréprochables : merci au ficcus benjamina à côté de la fenêtre.

Après des expériences en recherche peu orientées vers la théorie, un de mes buts dans ce doctorat était de me familiariser avec l'application de méthodes statistiques au traitement des langues. Et si, aujourd'hui, les expressions *recuit simulé* et *méthodes Monte Carlo* m'évoquent un peu plus qu'une technique de cuisine ou une principauté de la côté méditéranéenne, c'est grâce à l'accueil que m'a réservé l'équipe TSAC, notamment lors des séminaires du même nom. Un grand merci donc à Eric, Jean-François, Céline, François, Gersende, Jamal et Yann, ainsi qu'aux doctorants Julien, Jean-François et Natalya.

Ces trois ans auraient par ailleurs été moins agréables si je n'avais pas trouvé, aux détours de couloirs de l'ENST, quelques compagnons de galère. Je commencerais par remercier Christophe, Teodora et Valentin, compagnons indéfectibles de la dernière ligne droite sur les week-ends d'août 2006. Merci de m'avoir fait oublié par votre entrain que j'aurais préféré être sur les plages (de la côte Atlantique) que sur les pages (de ma thèse). Je voudrais également témoigner de ma gratitude à Grégoire, la gentillesse incarnée et la bible du C++ (comment ne pas se faire exploiter lorsque l'on combine ces deux qualités ?), Mathieu, peut-être celui que j'ai le plus croisé, toujours avec plaisir, de la première année à la soutenance, en passant par le Bureau des Doctorants, et Nicolas, mon binôme préféré de notre époque « cycle ingénieur », une référence précieuse pour mes questions en apprentissage automatique et surtout, simplement, une amitié durable. Difficile de citer exhaustivement toutes les autres personnes qui ont égayé ces trois années à l'ENST. Je me lance tout-de-même dans une liste résolumment aléatoire de mercis à destination du département TSI et/ou d'anciens du BDT, mes excuses par avance pour les inévitables oublis : Nicolas (Saunier), Guilhem, Damien, Sofia, Reda, Nicolas (Dailly), Romain, Jean-Philippe, Guillaume, Nancy, Pierre, Chloé, Cléo, Slim, Simoné, Thibault, Antoine, Miguel, Eduardo, Raphael, Julie, Thomas, Roland, Sarah, Bertrand, Amine, Lionel, Nicolas (Tizon), Sebastien, Maria, Fabrice, Laurence, Sophie-Charlotte, Patricia, Clara et Auguste.

Sans vouloir m'étendre démesurément sur la vie en dehors du bureau, j'ai aussi naturellement une pensée pour tous ceux, amis ou famille, parents et grands-parents, qui ont été présents pendant ces trois ans et en particulier le jour de la soutenance. Deux mentions spéciales à cet égard : l'une à ma grand-mère, qui m'a chaleureusement accueilli durant les deux semaines qui m'ont permis de rédiger la majeure partie du présent document ; l'autre à Anne-Laure, dont le soutien aura été absolument infaillible tout au long de cette thèse. Un grand merci à toi, même si, tout grand qu'il soit, il reste dérisoire par rapport à tout ce que tu m'as apporté et m'apporte encore.

Un dernier remerciement, avant de rentrer dans le vif du sujet, à l'attention des relecteurs de cette thèse, de ceux qui n'ont butiné que quelques pages aux quelques courageux qui ont dépassé l'introduction : outre l'ensemble du jury de thèse, je pense plus particulièrement à Nicolas, Zaïd, Christophe et Slim. Merci à vous, qui avez grandement contribué à l'amélioration de ce manuscrit.

[2] Oui, il y a eu comme un changement d'ambiance !

Résumé

Nous abordons dans cette thèse le problème de la classification non supervisée de documents par des méthodes probabilistes. Notre étude se concentre en particulier sur le modèle de mélange de lois multinomiales avec variables latentes thématiques au niveau des documents.

La construction de groupes de documents thématiquement homogènes est une des technologies de base de la fouille de texte, et trouve de multiples applications, aussi bien en recherche documentaire qu'en catégorisation de documents, ou encore pour le suivi de thèmes et la construction de résumés. Diverses propositions récentes ont été faites de modèles probabilistes permettant de déterminer de tels regroupements. Les modèles de classification probabilistes ont l'avantage de pouvoir également être vus comme des outils permettant de construire des représentations numériques synthétiques des informations contenues dans le document. Ces modèles, qui offrent des facilités pour la généralisation et l'interprétation des résultats, posent toutefois des problèmes d'estimation difficiles, qui sont dûs en particulier à la très grande dimensionnalité du vocabulaire.

Notre contribution à cette famille de travaux est double : nous présentons d'une part plusieurs algorithmes d'inférence, certains originaux, pour l'estimation du modèle de mélange de multinomiales ; nous présentons également une étude systématique des performances de ces algorithmes, fournissant ainsi de nouveaux outils méthodologiques pour mesurer les performances des outils de classification non supervisée. Les bons résultats obtenus par rapport à d'autres algorithmes classiques illustrent, à notre avis, la pertinence de ce modèle de mélange simple pour les corpus regroupant essentiellement des documents monothématiques.

Abstract

In this thesis, we investigate the use of a probabilistic model for unsupervised clustering of text collections. We focus in particular on the multinomial mixture model, with one latent theme variable per document.

Unsupervised clustering has become a basic module for many intelligent text processing applications, such as information retrieval, text classification or information extraction. Recent proposals have been made of probabilistic clustering models, which build "soft" theme-document associations. These models allow to compute, for each document, a probability vector whose values can be interpreted as the strength of the association between documents and clusters. As such, these vectors can also serve to project texts into a lower-dimensional "semantic" space. These models however pose non-trivial estimation problems, which are aggravated by the very high dimensionality of the parameter space.

The contribution of this study is twofold. First, we present and contrast various estimation procedures for the multinomial mixture model, some of which had not been tested before in this context. Second, we propose a systematic evaluation of the performances of these algorithms, thereby defining a framework to assess the quality of unsupervised text clustering methods. The comparison with the performances of other classical models demonstrates, in our opinion, the relevance of the simple multinomial mixture model for clustering corpus mainly composed of monothematic documents.

Table des matières

Chapitre 1

Introduction

1.1 Motivations

1.1.1 Enjeux de société et traitement automatique des langues

La numérisation des ressources textuelles de toutes origines est un enjeu majeur à l'ère de la société de l'information et de la communication. Il n'est pas anodin que les entreprises américaines Microsoft, Yahoo et Google s'y intéressent [Crane, 2006], suscitant à la fois enthousiasme et réticences (de la part notamment de Jean-Noël Jeanneney, président de la Bibliothèque Nationale de France [Jeanneney, 2005]). Le problème dépasse désormais largement le cadre de la recherche et entraîne des réactions de toutes parts, notamment dans le monde politique, jusqu'au président de la République. Pourquoi un tel déchaînement de passions ?

Au-delà des questions relatives au protectionnisme industriel ou à la propriété intellectuelle, retenons qu'en l'espace de quelques décennies, le grand public a pris conscience de l'importance de l'accès à l'information sous forme numérique. Le stockage binaire permet de copier et de conserver plus facilement les ressources, notamment audio et vidéo. Il s'agit donc d'un enjeu pour la préservation du patrimoine mais également pour son accessibilité au plus grand nombre. En complément des services locaux que sont les bibliothèques et les médiathèques, le développement très rapide d'Internet (et notamment du *World Wide Web*) permet de prendre connaissance en quelques clics d'une information stockée à l'autre bout du monde. Nous ne sommes probablement qu'au début du processus global de numérisation, en particulier pour ce qui est des ressources audiovisuelles (films, archives audio), mais, d'ores et déjà, la profusion de données textuelles a accéléré de façon mécanique la demande pour des outils de Traitement Automatique des Langues (TAL) rapides et performants. Ces techniques ont changé notre façon de concevoir la recherche d'information : l'indexation numérique offre un accès plus souple, robuste et rapide à la connaissance que le classement méticuleux des livres dans des rayonnages. Nul ne nie aujourd'hui la valeur ajoutée des services de recherche automatisés. Le recours à Internet est devenu incontournable pour trouver une réponse rapide à divers types de questions, souvent jusqu'à faire oublier toute solution alternative.

Cet avènement est dû, pour une part importante, à l'augmentation exponentielle de la puissance des ordinateurs et donc de la quantité de données qu'ils peuvent traiter. Alors que les premiers développement de l'intelligence artificielle cherchaient plutôt des méthodes ayant vocation à imiter les comportements humains, les algorithmes tirant pragmatiquement partie de la puissance de calcul des ordinateurs, sans lien avec les sciences cognitives,

ont aujourd'hui prouvé leur efficacité, notamment dans un certain nombre d'applications du TAL (classification, traduction, résumé, etc) [Jurafsky and Martin, 2000]. En particulier, le parti-pris du *TAL statistique* est que les traits pertinents du langage relatifs à une application donnée peuvent être *appris* sur de larges bases de documents, les *corpus*. Les connaissances structurées ajoutées, sous forme par exemple de bases de concepts, ou *ontologies*, ou de synthèses d'avis d'experts, sont en général peu utilisées en TAL statistique, le paradigme étant que toute information spécifique peut être apprise statistiquement moyennant une base documentaire suffisamment vaste et un modèle assez précis. Un exemple particulièrement éloquent à ce sujet est la recherche d'informations sur Internet, comme, par exemple, la date d'un événement historique. La réponse correcte se trouve dans un nombre de pages si grand que le fait que, localement, quelques sites puissent comporter des erreurs ne gêne pas l'utilisateur pour trouver, par redondance, la date exacte [Cohen, 2006].

Les méthodes statistiques ont l'avantage d'être, dans leur principe général, plus ou moins indépendantes de la langue et du domaine considérés. En effet, à partir d'un grand nombre d'exemples (qui sont, eux, naturellement dépendants du contexte), les algorithmes appliqués sont souvent suffisamment généraux pour fonctionner de la même manière dans des cas très différents. Ils ont également une capacité à tolérer et exploiter de façon satisfaisante des données comportant nombre d'erreurs et d'imperfections (*données bruitées*), permettant, notamment, de combiner plusieurs applications au sein d'une même chaîne de traitement. Ainsi, les systèmes de TAL prennent parfois en entrée le résultat de systèmes de reconnaissance optique ou vocale (par exemple, pour établir automatiquement un compte-rendu de réunion, il faut pouvoir associer reconnaissance du locuteur, reconnaissance vocale et résumé automatique) et la recherche d'information[1] est de plus en plus multilingue, comme l'est Internet (c'est le sujet des campagnes *Cross Language Evaluation Forum* [CLEF, 2000–2006]). Enfin, et ce n'est pas le moindre de leurs succès, les méthodes statistiques ont su prouver leur efficacité dans de nombreuses tâches proches de la recherche d'information lors de projets de recherche et de challenges entre plusieurs équipes, tels que ceux organisés par le NIST (institut national des standards et de la technologie américain), et notamment les *Text REtrieval Conferences* [TREC, 1992–2006], ou, en France, le DÉfi Fouille de Textes [Alphonse et al., 2005, et annexe B].

1.1.2 Objectifs

Nous nous intéressons, dans cette étude, à des méthodes permettant de constituer des groupements (*clusters* en anglais) par *similarité* dans un corpus de documents. Définir la similarité attendue n'est pas évident car elle peut se situer à bien des niveaux : similarité de genres, d'auteurs, de styles, de langues, tout dépend en réalité de la nature du corpus et de l'application finale.

Nous avons ici cherché plus spécifiquement des proximités ou des cohérences intra-groupes se situant au niveau du vocabulaire, et en particulier des proximités *sémantiques*, c'est-à-dire que nous avons souhaité regrouper des textes qui « parlent de la même chose ». Être plus précis est difficile car, contrairement à la tâche de catégorisation où les thèmes à rattacher aux documents sont définis a priori, nous espérons ici les voir émerger des

[1] La recherche d'information (*information retrieval*) est la tâche du TAL consistant à trouver dans un ensemble de documents ceux qui sont le plus pertinents vis-à-vis d'une *requête* (*query*) constituée d'un mot ou d'un nombre réduit de mots, comme le font les moteurs de recherche.

données elles-mêmes, avec un niveau de granularité contrôlé par l'utilisateur[2].

Le cadre général pour cette étude est celui de la *classification non supervisée*. Un des objectifs privilégiés est l'*analyse exploratoire* de corpus qui consiste, en quelque sorte, à utiliser les méthodes de classification non supervisée comme des « loupes » permettant d'examiner globalement le contenu d'un corpus, sans avoir à le lire de façon détaillée. Le produit de ce type d'étude est un *plan de classement*, c'est-à-dire une façon d'affecter chaque texte du corpus à un ou plusieurs groupes. Suivant les méthodes, ce plan de classement est ou pas applicable à des données reçues ultérieurement à son élaboration.

Les spécificités des données textuelles, et notamment les questions de polysémie et de synonymie, font échouer dans cette tâche les stratégies les plus immédiates. C'est la raison pour laquelle un grand nombre de méthodes ont été proposées pour l'analyse exploratoire, faisant de la classification non supervisée de documents un sujet de recherche passionnant au croisement de plusieurs domaines mais avec ses problématiques particulières, telles que les choix de représentation, de mesures de similarité ou de procédures d'évaluation, par exemple. Dans cette thèse, nous nous sommes tout spécialement intéressés à l'apport des méthodes probabilistes dans ce contexte. Nous expliquerons pourquoi en commençant par description générale du domaine dans la section 1.2.

1.2 Classification non supervisée

1.2.1 Limites de la classification supervisée

La classification non supervisée s'oppose à la catégorisation de documents ou classification supervisée, qui consiste à assigner une étiquette à de nouveaux textes à partir d'exemples [Cheeseman and Stutz, 1996, Mitchell, 1997]. L'application la plus classique de la catégorisation est l'*indexation*, consistant à chercher au sein d'un ensemble de catégories existantes (les rayonnages ou sections d'une bibliothèque) celle(s) qui correspond(ent) le mieux à chaque nouveau document. Cette tâche prend beaucoup de temps lorsqu'elle doit être effectuée manuellement et l'intérêt de l'automatisation est évident dans ce contexte [Sebastiani, 2002]. Parmi les méthodes ayant reçu le plus d'attention, citons, entre autres, le classifieur bayésien naïf [Lewis, 1998] ou les séparateurs à vaste marge (*support vector machines* en anglais) [Joachims, 1998]. Il y a aujourd'hui un large consensus dans la communauté apprentissage/traitement automatique des langues sur le fait que le problème de la classification supervisée de documents est en grande partie résolu, atteignant les niveaux de performances de la classification manuelle [Sebastiani, 2002].

En classification supervisée, les données d'apprentissage étant naturellement *définies à l'avance, préalablement à l'application de l'algorithme*, il en va de même des catégories constituant les étiquettes : elles sont en nombre fini et en général non modifiables. Dans les applications de classification non supervisée, les étiquetages ne sont pas disponibles ou bien sont trop coûteux pour pouvoir être effectués de façon substantielle.

[2]Par exemple, si notre tâche est de classer les cinq ouvrages suivants : *La critique de la raison pure*, *La phénoménologie de l'esprit*, *Oui-Oui et la chasse au trésor*, *Oui-Oui et la gomme magique* et *Les aventures de Pinocchio*, il est raisonnable, dans un premier temps, et pour protéger l'innocence de l'enfance, de séparer les deux premiers des trois autres. Cependant, après avoir retiré Kant et Hegel, il est tout aussi raisonnable d'exiger que la classification soit adaptée et sache distinguer Oui-oui de Pinocchio au moyen, par exemple, de la petite voiture jaune et rouge (il est bien connu que Pinocchio n'a pas le permis de conduire).

1.2.2 Applications de la classification non supervisée

Des applications pratiques de la classification non supervisée sont, par exemple, les tâches de détection de nouveauté ou de veille technologique, dans lesquelles on peut être amené à recevoir de nombreux textes parlant de sujets divers, nouveaux et trop pointus pour disposer d'une catégorisation préalable. L'*analyse exploratoire* a ainsi pour vocation de structurer l'objet d'étude, avec ou non la volonté de pouvoir généraliser les résultats à des données futures. Il faut noter que l'expression n'est pas nouvelle puisque déjà employée il y a quelques décennies dans le cadre de l'analyse de données [Benzécri et al., 1981]. Elle désignait alors la démarche consistant à rechercher dans les données des régularités non attendues au moment de leur collecte, avec une ambition de visualisation et d'interprétation des résultats. Les ambitions sont donc similaires, même si les applications et objectifs finals sont, dans notre cas (classification non supervisée), moins liés à l'étude de la langue que pour Benzécri (analyse littéraire, lexicométrie).

Par ailleurs, les algorithmes de classification non supervisée étant des outils permettant de regrouper divers types de données, ils sont également utiles en tant qu'éléments de systèmes plus complexes, notamment en recherche d'information et en catégorisation. Dans les problèmes de grande dimensionnalité, ils permettent de réduire le nombre d'attributs en les regroupant en sous-ensembles cohérents. En classification supervisée, lorsque les catégories initiales sont fortement hétérogènes, il y a un avantage clair à les scinder en sous-catégories plus cohérentes [Vinot and Yvon, 2003]. Dans ce type d'application, le regroupement de documents est considéré uniquement comme un moyen de réaliser plus efficacement un autre objectif. La connaissance du corpus en soi n'a pas d'importance.

Enfin, une autre application de la classification non supervisée est l'étude de la langue. Pour obtenir de bonnes capacités de prédiction sur l'ensemble d'un corpus, un modèle de langage particulier n'est pas toujours suffisant et il peut être avantageux de construire différents modèles sur des groupes homogènes de documents. Dans une démarche de type *diviser-pour-régner*, le partitionnement est donc ici un moyen de se simplifier une tâche de modélisation difficile en considérant des sous-corpus plus réguliers (et donc plus simples à prédire).

Les applications de la classification non supervisée sont donc importantes, et présentes à de nombreux niveaux. Dans la section suivante, nous présentons les grandes familles d'algorithmes dédiés à cette tâche avant de proposer une discussion sur les critères de sélection entre ces méthodes.

1.2.3 Deux paradigmes concurrents

Parmi les méthodes de classification non supervisée de documents dotées de justifications théoriques solides et ayant reçu une large attention dans la communauté de recherche, il est possible de distinguer deux grandes tendances : les méthodes *vectorielles* et les méthodes *probabilistes*.

Pour expliquer ce qui les distingue, il est possible de partir d'un problème sous-jacent, celui du choix de la représentation :

- Il est possible de modéliser chaque texte comme un vecteur. Il peut s'agir d'un vecteur d'entiers de la taille du vocabulaire contenant les comptes de mots dans le document en question (modèle du « sac de mots », section 2.1.1) ou d'un vecteur de réels obtenu par diverses transformations (par exemple, la représentation *fréquence inversée sur les documents*, ou *idf*, section 2.2.2). Faire le choix d'une représentation revient implicitement à choisir un espace, dans lequel les documents sont supposés

conformément à la classification recherchée. Ce problème peut être également vu comme le choix d'une distance pertinente entre les textes.

- Dans la vision probabiliste, le corpus est vu comme le résultat d'un processus génératif selon un modèle sous-jacent. Les observations indépendantes et identiquement distribuées sont, par exemple, suivant les points de vue, les documents ou les occurrences mais la caractéristique commune des modèles probabilistes est qu'ils font intervenir un certain nombre de paramètres, dont la détermination ou *inférence* est une étape-clé de la résolution du problème de classification.

Pour les méthodes vectorielles, une fois le travail de modélisation effectué, il est possible d'appliquer un grand nombre d'algorithmes classiques en analyse de données fondés sur les proximités entre les documents. L'un des plus utilisés est probablement les K-moyennes [MacQueen, 1967], sur lequel nous reviendrons en sections 2.2.1 et 5.1. Le résultat des K-moyennes, comme d'autres algorithmes de classification, est que chaque document appartient à un et un seul thème de façon non ambigüe. Nous parlons dans ce cas de *classification déterministe*. À l'inverse, il est envisageable que chaque document ne soit pas affecté à un thème mais ait une certaine probabilité d'appartenir à chaque groupe. Un partitionnement est alors caractérisé par les vecteurs (de somme 1) donnant les probabilités pour chaque document d'appartenir aux thèmes. Il s'agit alors de *classification probabiliste*[3]. Les méthodes probabilistes produisent par définition des classifications probabilistes. En revanche, il est faux de dire que toutes les méthodes vectorielles résultent en des partitionnements déterministes. De fait, que cela soit dû à une opération de projection sur des sous-espaces ou à la définition des modèles, il existe des méthodes vectorielles qui produisent une classification probabiliste :

- l'analyse sémantique latente (LSA) [Deerwester et al., 1990] est l'application au cas particulier de la matrice de comptes d'un algorithme plus général de décomposition matricielle l'analyse en composantes principales (ACP). L'ACP consiste à représenter une matrice de façon réduite et robuste, en effectuant une décomposition en valeurs propres (ou singulières) et en ne conservant que les plus significatives ;
- la décomposition en matrices non négatives (NMF) [Lee and Seung, 2001] relie le problème de la classification non supervisée à une autre décomposition matricielle particulière, qui a, par rapport à LSA, la particularité que tous les éléments soient positifs ou nuls (ce qui rend en principe les résultats plus faciles à interpréter) ;
- un autre algorithme non probabiliste, se fondant sur la théorie de l'information, a trouvé de nombreuses applications, en particulier dans la classification non supervisée textuelle : il s'agit de la méthode de réduction de redondance par « goulot d'information » [Slonim and Tishby, 2000].

Nous aborderons plus en détail ces méthodes *vectorielles* en section 2.2.

Les méthodes probabilistes que nous allons étudier reposent toutes sur l'existence d'une matrice de paramètres β qui associe le vocabulaire aux « thèmes » recherchés dans le corpus. Informellement, elle représente l'importance de chaque mot dans chaque thème. Les différentes méthodes se distinguent en revanche par le niveau auquel est déterminé le thème latent.

- pour le modèle de mélange de multinomiales [Nigam et al., 2000], qui est celui qui a principalement retenu notre attention dans le reste de la thèse, le tirage du thème est effectué une fois par document. On parle pour cette raison de représentation

[3]Une classification déterministe est en fait un cas particulier de classification probabiliste dans lequel toutes les probabilités d'appartenance sont 0, sauf celle qui indique le thème associé au document, qui vaut 1.

monothématique ;
- l'analyse probabiliste sémantique latente (PLSA) [Hofmann, 2001] choisit de considérer les couples mot/document comme objet d'étude privilégié. PLSA est le premier modèle probabiliste faisant un lien avec les modèles vectoriels, LSA dans ce cas ;
- l'allocation Dirichlet latente (LDA) [Blei et al., 2002], est probablement aujourd'hui le modèle le plus étudié dans le domaine de la fouille de textes probabiliste. Une variable thématique est tirée pour chaque occurrence, selon une distribution dépendante du document. Le mécanisme de génération en est rendu plus complexe et, par conséquent, la phase d'inférence pose également plus de difficultés ;
- notons également l'existence d'un modèle *Gamma-Poisson* (GaP) [Canny, 2004] qui se situe dans la lignée de LDA, mais propose de modéliser les données par d'autres distributions de probabilité.

Nous passerons en revue ces méthodes *probabilistes* en section 2.3. Le modèle de mélange sera d'autre part abordé plus longuement au chapitre 4 et une étude plus détaillée de LDA sera présentée au chapitre 5.

Devant le nombre et la variété des méthodes de classification non supervisée applicables, il nous semble à présent nécessaire de formuler des critères de sélection.

1.2.4 Critères de choix

Possibilité de généralisation de la classification L'objectif premier d'un algorithme de classification non supervisée est, par définition, de proposer un regroupement cohérent des textes considérés. Cependant, une autre question, pas nécessairement moins importante, que nous pouvons nous poser est : une fois établi ce classement de référence sur un certain nombre d'exemples, saurons-nous rapidement affecter un thème à un nouveau document, reçu postérieurement à la phase de conception de la classification ? En d'autres termes, les résultats que nous avons obtenus sur un ensemble d'apprentissage donné sont-ils extrapolables ? Le pire cas est naturellement d'avoir à relancer l'ensemble de l'algorithme pour une simple observation supplémentaire. Les méthodes que nous présenterons dans le chapitre 2 sont plus ou moins adaptées à cette contrainte. Pour le modèle de mélange de multinomiales (section 2.3.2), ce problème est réglé de façon simple car, pour un nouveau vecteur de comptes donné, il est possible de déterminer explicitement la distribution conditionnelle de la variable définissant le thème. Pour les autres modèles, les solutions sont généralement moins justifiées théoriquement. Ainsi les méthodes vectorielles préconisent-elles de projeter le nouveau vecteur document sur « l'espace sémantique » (voir en particulier LSA, section 2.2.3). Pour intuitives qu'elles soient, ces solutions paraissent un peu « ad hoc » par opposition au cadre statistique où la frontière est claire entre l'estimation des paramètres lors de la phase d'apprentissage et l'utilisation de ces estimés pendant la phase de test.

Cette problématique est liée à une autre question, souvent importante pour les applications industrielles mais à laquelle nous avons accordé moins de poids dans le cadre de cette thèse. Elle porte sur la possibilité d'effectuer un apprentissage *incrémental*, c'est-à-dire de pouvoir accepter les textes un à un et d'être capable de raffiner le modèle à chaque nouvelle observation. Par défaut, la plupart des algorithmes requièrent l'ensemble des observations avant exécution de la phase d'apprentissage, mais il existe dans bien des cas des versions incrémentales. Citons par exemple [Brand, 2002] pour la décomposition en valeurs singulières (LSA) et [Neal and Hinton, 1993] pour l'algorithme espérance-maximisation (modèle de mélange de multinomiales).

Interprétabilité des résultats L'analyse exploratoire vise à donner à l'utilisateur une meilleure connaissance du corpus d'étude. Dans ce contexte, il n'est pas suffisant d'obtenir un partitionnement des documents, mais il est aussi crucial de comprendre à quoi fait référence chacun des thèmes, de déterminer son importance dans le corpus et sa cohérence interne. Les modèles probabilistes semblent disposer de ce point de vue d'un avantage sur les autres, dans la mesure où ces problèmes peuvent s'exprimer en termes statistiques clairs. Par exemple, l'étude du paramètre β liant thèmes et vocabulaire est d'une aide considérable pour la détermination des mots les plus importants dans chaque groupe. Cependant, la connaissance de ce paramètre, même si elle plaide pour l'adoption des modèles probabilistes, ne résout pas tous les problèmes. Nous y reviendrons en détail dans la section 5.3.

Autres critères La vitesse est traditionnellement un critère important pour évaluer un algorithme. Sous ce terme, se cachent en réalité deux critères corrélés mais non identiques : l'ordre théorique de complexité d'une part et les performances pratiques d'autre part, qui peuvent être influencées par bien d'autres caractéristiques (structure des données, compromis mémoire utilisée/vitesse d'exécution, langage de programmation, etc). Dans notre cas, les calculs de complexité théoriques ne sont pas très informatifs compte tenu de la variabilité des représentations et du nombre de grandeurs variables à considérer (nombre de documents en apprentissage, en test, taille du vocabulaire, longueur des documents, nombre de thèmes, etc). Par ailleurs, du point de vue de la vitesse d'exécution réelle, aucun algorithme ne semble se distinguer sensiblement dans nos expériences. Nous n'avons donc pas choisi ce critère parmi nos priorités. Il faut toutefois noter que, pour les applications industrielles où les documents peuvent se compter en centaines de millions et les tailles de vocabulaire en milliards de mots, des différences qui ne nous ont pas paru significatives peuvent se révéler importantes et, l'optimisation du code, qui n'a ici pas retenu l'essentiel de notre attention, est également une étape décisive.

Une autre question importante est la dépendance de l'algorithme de classification vis-à-vis de l'initialisation ou, en d'autres termes, la stabilité de ses performances. Il est difficile de présenter une étude systématique de stabilité pour toutes les méthodes étudiées. Nous reviendrons néanmoins sur ces questions pour le modèle de mélange de multinomiales dans le chapitre 4 et pour LDA dans le chapitre 5. Le problème de la sélection du nombre de thèmes est également un problème traditionnellement difficile en classification non supervisée. L'existence d'une stratégie naturelle pour le choix de cet ordre de modèle serait donc un argument fort en faveur d'une méthode. Néanmoins, ce problème se pose dans les mêmes termes pour toutes les méthodes étudiées au chapitre 2 et les solutions, qui relèvent du cadre général de *la sélection de modèles*, ne seront pas présentées dans cette thèse. Enfin, un dernier critère potentiellement pertinent pour la sélection de méthodes est celui de *symétrie mots/documents*. Les méthodes considérées permettent-elles d'opérer une classification sur les termes aussi facilement que sur les textes ? La réponse est plus naturellement oui pour les méthodes vectorielles, et en particulier celle du goulot d'information (section 2.2.6), mais notre objectif étant en priorité la classification non supervisée de *documents*, nous n'avons pas privilégié particulièrement cette vision symétrique.

Les critères que nous avons considérés prioritaires, la possibilité de classer un document non vu et l'interprétabilité des résultats, orientent donc le choix vers les modèles probabilistes, au détriment d'autres méthodes de classification non supervisée pour lesquelles les étapes post-apprentissage ne sont pas aussi directes.

1.3 Contributions

Le point de vue développé dans cette thèse est donc que les méthodes probabilistes permettent un traitement plus élégant et théoriquement cohérent que leurs homologues non probabilistes. Nous présentons dans l'état de l'art un ensemble de notations unique permettant de mettre plus facilement en évidence les liens et différences entre les modèles. Nous proposons par ailleurs dans le cadre probabiliste de réfléchir sur les lois statistiques les mieux appropriées pour modéliser les comptes de mots, un sujet souvent absent des analyses concernant PLSA, LDA et les modèles similaires.

Dans le but d'établir un cadre expérimental d'évaluation, nous avons mené l'étude la plus exhaustive possible sur l'ensemble des mesures de performance utilisée en classification non supervisée, en mettant l'accent sur celles qui sont le plus pertinentes pour les données textuelles.

Par nos expériences sur le modèle de mélange de multinomiales et LDA, nous montrons que :

- la sensibilité de l'algorithme espérance-maximisation aux conditions initiales est due à la taille du vocabulaire et qu'une heuristique simple d'inférence itérative permet d'améliorer la stabilité et les performances ;
- la tendance de l'algorithme espérance-maximisation à produire une classification presque déterministe est un effet conjoint de la taille du vocabulaire et des longueurs des documents ;
- l'algorithme d'échantillonnage de Gibbs rao-blackwellisé, qui avait jusqu'à présent été testé uniquement pour LDA, peut être également utilisé pour le modèle de mélange de multinomiales ;
- la réponse généralement apportée aux performances médiocres du modèle de mélange de multinomiales, notamment par LDA, d'adopter un processus de génération plus compliqué n'est pas l'unique solution. De fait, conserver un modèle simple et concentrer ses efforts sur la phase d'inférence est également une stratégie raisonnable, l'utilisation de LDA n'étant manifestement pas toujours une garantie de bons résultats de classification.

En revanche, nous n'avons pas la prétention d'aborder le problème dans sa plus grande généralité. Les aspects importants de la classification non supervisée de documents qui ne sont pas ou peu abordés dans le document sont les suivants :

- nous avons essentiellement travaillé sur un corpus et peu étudié la généralisation à d'autres situations. La section 4.5 présente quelques tests sur d'autres jeux de données mais notre étude demeure dans un cadre où les documents sont plutôt monothématiques et les catégories mutuellement recouvrantes sont peu nombreuses ;
- parmi les questions également évoquées en section 4.5, les différences de comportement de l'algorithme EM pour de très grands nombres de thèmes n'ont pas été étudiées, pas plus que l'adaptabilité du cadre d'évaluation à cette situation ;
- en dehors de l'état de l'art, nous n'avons pas abordé les extensions hiérarchiques des modèles probabilistes. Des méthodes théoriquement et pratiquement satisfaisantes restent selon nous à découvrir pour ce problème ;
- nous avons toujours considéré disposer du « bon » nombre de thèmes, une hypothèse naïvement optimiste, particulièrement en analyse exploratoire lorsque le corpus est majoritairement inconnu. Une piste privilégiée dans notre contexte serait d'appliquer des résultats de la théorie statistique de sélection de modèles au problème pratique du choix du nombre de thèmes.

1.4 Organisation de la thèse

Le cœur de la thèse est l'étude du modèle de mélange de multinomiales, sous divers aspects et avec la préoccupation récurrente de chercher des voies pour l'améliorer. Nous aurons également l'occasion de présenter d'autres modèles, d'introduire un cadre d'évaluation ou d'aborder les problèmes spécifiques d'inférence des paramètres.

Dans le chapitre 2, la première section aborde les problèmes de prétraitement des données, des questions pratiques de formation du vocabulaire à celles de la modélisation statistique des comptes et pose les notations utilisées dans la suite de la thèse. Le reste du chapitre est consacré à l'état de l'art proprement dit, en commençant par les méthodes les plus utilisées pour la classification de textes non supervisée. La seconde section regroupe de façon un peu arbitraire les méthodes non probabilistes : K-moyennes, LSA, NMF, regroupement spectral et goulot d'information principalement, suivis de présentations plus brèves d'autres méthodes (section 2.2.7). Dans la troisième partie, nous nous intéressons par opposition aux algorithmes reposant sur un modèle probabiliste, quelle que soit la méthode d'inférence des paramètres. Nous y abordons le modèle de mélange de multinomiales, PLSA, LDA, GaP, ainsi que des extensions hiérarchiques (MASHA/HPLSA) et une étude sur les difficultés spécifiques liées à la modélisation des comptes de mots. En fin de chapitre, nous consacrons une section aux liens entre les différents modèles avant de justifier notre choix de nous intéresser plus particulièrement au modèle de mélange de lois multinomiales.

Le chapitre 3 est consacré à l'évaluation dans le domaine de la classification non supervisée. Nous aborderons les difficultés inhérentes à la tâche. Elles proviennent pour la plupart du fait que l'objectif de la classification non supervisée, bien qu'ayant un sens, ne peut être défini aisément en termes clairs et non subjectifs. Nous développerons ensuite les mesures que nous avons utilisées dans la suite de la thèse, essentiellement la perplexité et score de coocurrences avec appariement par la méthode hongroise.

Nous revenons plus en détail sur le modèle de mélange de multinomiales dans le chapitre 4 et en particulier l'inférence via l'algorithme Espérance-Maximisation (EM) puis l'échantillonnage de Gibbs. Nous présentons les premières expériences conduites sur le modèle, portant en particulier sur le lissage. Nous nous interrogeons par ailleurs sur la nature réellement probabiliste de la classification obtenue en montrant que, dans ce cas, la différence de résultats avec un algorithme de partitionnement déterministe est faible. Nous montrons en quoi nous sommes de façon typique dans une situation connue en apprentissage sous le nom de *malédiction de la dimensionnalité* (ou *curse of dimensionality*, voir, par exemple, [Duda et al., 2000]). La mise en évidence de ces difficultés inspire ensuite de nouvelles méthodes d'inférence. L'importance de l'initialisation dans un espace de grande dimensionnalité est contournée par le biais d'un algorithme itératif. Nous montrons également comment appliquer des méthodes numériques (MCMC, *chaînes de Markov Monte Carlo*) pour l'inférence, en particulier l'échantillonnage de Gibbs. Nous concluons ce chapitre sur une brève note relative à l'apprentissage supervisé.

Le chapitre 5 est également assez orienté vers l'expérimentation, mais avec l'ambition de sortir du cadre strict de l'étude du modèle de mélange de multinomiales pour le comparer à d'autres méthodes de classification non supervisée : l'algorithme classique des K-moyennes et l'allocation Dirichlet latente (LDA). La deuxième partie du chapitre aborde le problème de l'interprétation des thèmes et propose quelques méthodes simples pour déterminer les mots les plus représentatifs de chaque regroupement. Dans un souci de cohérence, nous avons mis l'accent sur les méthodes endogènes aux modèles probabilistes, c'est-à-dire ne nécessitant pas d'appliquer en supplément des techniques non liées à la théorie statistique.

Chapitre 2

État de l'art

Parmi les nombreux modèles existants pour la classification non supervisée, nous avons choisi de privilégier dans ce chapitre ceux qui ont fait leurs preuves pour l'analyse de données textuelles. Le domaine d'application du texte a en effet certaines spécificités, notamment dans la modélisation des documents, qui font que des algorithmes pourtant performants par ailleurs y sont manifestement moins adaptés. C'est le cas par exemple de tous ceux qui se fondent sur une distance euclidienne sur la matrice de comptes, notamment les algorithmes de type hiérarchique agglomératif et la version la plus élémentaire de l'algorithme des K-moyennes[1] [Jurafsky and Martin, 2000].

Tous les articles que nous présentons ici adoptent l'hypothèse du « sac de mots » (comme vecteur sur l'espace du vocabulaire) comme point de départ du traitement des documents, sans que cela soit un choix délibéré de notre part dans la sélection bibliographique. Il faut certainement y voir l'influence considérable du modèle d'espace vectoriel (*Vector Space Model*) [Salton et al., 1975] en fouille de textes et en recherche d'information au cours des dernières décennies. Avant l'étude bibliographique proprement dite, nous consacrerons une première section au prétraitement des données, en mettant l'accent sur la phase de construction du vocabulaire, avant d'introduire les notations utilisées dans la suite de la thèse.

Notre présentation des méthodes non probabilistes commence par les méthodes les plus classiques en classification non supervisée, avant de détailler les spécificités d'application aux données textuelles. Nous nous intéresserons ensuite en détail aux quatre méthodes de classification non supervisée de documents qui nous semblent les plus importantes (analyse sémantique latente, regroupement spectral, factorisation en matrices non-négatives et goulot d'information) et en citerons plus brièvement d'autres dans une cinquième sous-section. Les troisième et quatrième sections du chapitre sont dédiées aux modèles probabilistes, d'abord par une succession de présentations des modèles, du plus simple (le mélange de multinomiales, étudié plus en détail dans le chapitre 4) au plus évolué (allocation Dirichlet latente, repris dans le chapitre 5), puis par une mise en évidence des relations entre eux. Enfin, nous expliquons dans la dernière section les choix de la thèse, justifiant, d'une part, le biais assumé de cette bibliographie en faveur des modèles probabilistes et, d'autre part, la volonté de s'intéresser en priorité au mélange de multinomiales au détriment des modèles plus complexes.

[1] Il suffit de considérer la comparaison d'un texte avec le même document dupliqué deux fois pour constater que la distance euclidienne est inadaptée à la représentation des textes sous forme de comptes : dans un espace à très grande dimension comme le vocabulaire, il est plus important d'avoir des valeurs non nulles sur les mêmes indices que d'avoir exactement la même valeur de compte pour chaque mot.

2.1 Représentation des documents

2.1.1 Prétraitements

Pour opérer un traitement statistique sur des données, il faut trouver une représentation du corpus sur laquelle construire un modèle. De nombreux choix qualitatifs ou quantitatifs sont possibles mais le plus répandu en fouille de textes et en recherche d'information est celui d'associer chaque document à un vecteur de comptes sur le vocabulaire. Cette hypothèse est connue sous le nom d'hypothèse du *« sac de mots »* (*« bag of words »*) [Salton et al., 1975]. Elle peut paraître outrageusement simplificatrice puisqu'il s'agit de s'affranchir de l'ordre des mots et de ne se préoccuper que de leur nombre d'occurrences dans le texte[2]. Nous supposons toutefois que les occurrences de mots vont, à elles seules, être suffisamment informatives pour la tâche qui nous intéresse.

Pour extraire les mots du texte, différents traitements sont possibles, avec des techniques d'analyse linguistique plus ou moins poussées et des coûts en temps et en ressources très variables. Ainsi, l'étape de segmentation en mots ou de *tokenisation*, c'est-à-dire l'identification de chaque forme[3], fait intervenir une certain nombre de décisions, plus ou moins arbitraires, que nous détaillons ci-dessous.

Filtrage des signes La ponctuation entre les mots est en général ignorée et l'intuition que son apport sémantique est limité par rapport aux mots eux-mêmes semble justifiée. Le problème est en revanche un peu plus compliqué pour les chiffres et dates. Leur apport n'est parfois pas négligeable mais à condition de les normaliser correctement et de savoir reconnaître que, par exemple, les chaînes *11-09-2001*, *11 septembre 2001* ou *2001 9/11* doivent être regroupées sous le même indice. Une difficulté analogue se retrouve pour les nombres (*cent mille*, *100000*, *100 000*, etc).

Filtrage des *anti-mots* Il est également d'usage de retirer les mots de liaison et d'articulation du texte (articles, conjonctions de coordination, prépositions), appelés *anti-mots* (*stop words* en anglais), car leur pouvoir discriminant d'un document par rapport à un autre est souvent faible et ils ne sont pas d'un grand secours, dès lors que l'hypothèse du « sac de mots » est acceptée. D'autre part, le fait d'éliminer les mots présentant les plus fortes fréquences produit un effet d'« écrêtage », c'est-à-dire diminue l'amplitude totale de la bande des fréquences considérées et donc, potentiellement, les difficultés de stockage et de calcul. Cependant, le filtrage des anti-mots repose souvent sur une liste pré-établie inévitablement arbitraire. Et, par ailleurs, il existe des cas dans lesquels les anti-mots sont effectivement utiles ([Sebastiani, 2002] cite notamment les travaux d'attribution de « paternité » à des textes d'auteurs inconnus, dans lesquels les critères stylistiques sont déterminants), ce qui rend la décision de filtrer les anti-mots éminemment dépendante du corpus et de l'application.

[2]Ce faisant, il est évident qu'une partie significative de l'information est perdue. Par exemple, sous l'hypothèse du « sac de mots », la phrase *l'éléphant est plus fort que l'hippopotame* devient strictement équivalente à *l'hippopotame est plus fort que l'éléphant*. En fait, il se trouve que la plupart des méthodes que nous présentons pourraient fonctionner sur des modèles plus complexes. Par exemple, le problème de différencier les deux phrases ci-dessus serait réglé en considérant comme unité de base les couples de mots, ou *bigrammes*, pour un traitement théorique souvent assez analogue mais au prix d'une complexité de calcul accrue.

[3]Pour la définition de forme/mot/occurrence, nous renvoyons le lecteur au glossaire p.139.

Filtrage des *hapax* Le modèle du « sac de mots » est lié de façon inhérente à des problèmes de grande dimensionnalité : la taille du vocabulaire, c'est-à-dire le nombre de formes différentes, dépasse toujours le nombre de documents (de l'ordre de 40000 pour 5000 documents, dans nos expériences). Sur l'ensemble de ce vocabulaire, une majorité des mots apparaît très peu (il s'agit d'un effet de la loi de Zipf, voir la section 2.3.7.1) et ne représente qu'un nombre réduit d'occurrences. Ainsi, un prétraitement courant est d'ignorer les termes qui n'ont qu'une occurrence dans le corpus (*hapax*) ou moins d'un compte-seuil, ou bien encore d'écarter ceux qui n'apparaissent que dans un nombre limité de documents distincts. La réduction du vocabulaire peut avoir le double avantage d'accélérer le traitement informatique et de rendre les données moins bruitées (les données textuelles n'étant jamais parfaites, un certain nombre d'hapax peuvent notamment correspondre à des mots courants mal orthographiés). Nous aurons recours à ce type de méthodes et analyserons leur effet dans la section 4.3.1.

Racinisation, lemmatisation Dans certaines situations, il peut sembler souhaitable de regrouper certaines formes, considérant, par exemple, que qu'un mot au singulier et sa forme plurielle désignent fondamentalement le même concept. Pour que leurs comptes soient additionnés, il faut pouvoir les détecter comme étant identiques. La technique la plus basique (*racinisation*, ou *stemming* en anglais) fait simplement appel à des heuristiques consistant à tronquer certains suffixes (typiquement la lettre *s* finale pour le cas singulier/pluriel). Des mécanismes linguistiques plus évolués et reposant sur une analyse du texte plus poussée (*lemmatisation*) permettent d'obtenir les *lemmes* constituant la base canonique de chaque forme (infinitif pour un verbe, masculien singulier pour un nom). Cependant, les résultats de ces outils, bien qu'ils soient statistiquement satisfaisants, ne sont pas parfaits et sont susceptibles d'aggraver ponctuellement des problèmes d'homonymie (par exemple, en regroupant sous le même lemme les mots *(se) reposer* et *poser*). L'utilisation de ces techniques a fait l'objet de longs débats en analyse des données textuelles (voir par exemple [Lebart and Salem, 1988]). Pour le cas de la classification (supervisée ou non supervisée), la pertinence de ces méthodes demeure incertaine [Sebastiani, 2002]. En outre, une recommendation générale à l'analyse statistique de données, au-delà de l'application textuelle, est de ne pas appliquer de simplifications de représentation a priori, tant qu'il n'y a pas de certitude qu'elles soient bénéfiques [Cheeseman and Stutz, 1996].

Groupement des unités L'unité de base choisie est le mot. On parle dans ce cas de modèle *unigramme* pour dire que la dépendance entre les différentes occurrences n'est pas considérée directement. Il est possible de travailler sur des unités de base plus grandes telles que les couples de mots (*bigrammes*). La taille du vocabulaire devient, dans le pire des cas, ${n_W}^2$. Bien que ce traitement permette de retenir plus d'information relative au texte original, l'augmentation de dimensionnalité qui en résulte est souvent impossible à gérer en pratique, dans la mesure où n_W est déjà lui-même très grand[4]. En outre, le nombre d'occurrences l restant identique alors que la taille du vocabulaire augmente, il est probable que l'estimation des paramètres soit moins bonne. Une autre possibilité pour regrouper les mots est de chercher les termes qui font souvent partie du même syntagme (par exemple, *Eiffel* est souvent associé à *tour*) pour les compter sous le même indice. Là encore, il n'y a pas unanimité sur l'impact positif de ces idées en classification [Sebastiani, 2002].

[4]Pour un exemple d'étude sur les bigrammes, voir [Wallach, 2006] qui adapte le modèle de l'allocation Dirichlet latente à ce cas.

La discussion qui précède indique les questions à considérer pour former le vocabulaire à partir de textes bruts, mais ne dit pas quels textes considérer. Nous supposons disposer de deux corpus, l'un étant utilisé pour la phase d'apprentissage (l'estimation des paramètres) et l'autre pour la phase de test (l'évaluation du modèle). Une solution consiste à construire le vocabulaire à partir de l'ensemble des formes des corpus d'apprentissage et de test. Cependant, cette stratégie équivaut à incorporer des informations issues du corpus de test dans la phase d'apprentissage, ce qui n'est pas acceptable pour une procédure d'évaluation objective.

Dès lors, une solution plus acceptable est simplement d'ignorer les mots du corpus de test absents du vocabulaire d'apprentissage. Cette décision suppose implicitement que les mots écartés ne sont pas nécessaires pour élaborer un plan de classement adéquat. Or, dans les cas réels, les textes non vus contiennent forcément des formes importantes ne figurant pas dans le corpus d'apprentissage, ne serait-ce que dans les noms propres. Une autre stratégie pour prendre en compte ces nouvelles formes consiste par exemple à compter ces occurrences ensemble sous l'appellation *« out of vocabulary »*. Cette méthode a l'effet de regrouper sous un même indice des mots potentiellement très différents et de créer un mot très fréquent dans l'ensemble de test (et ne respectant pas les distributions naturelles des comptes de mots) et dont on ne sait rien puisqu'il n'a jamais été vu dans l'ensemble d'apprentissage. Cette situation ne semble finalement pas régler de façon définitive le problème de prise en comptes des nouvelles formes, et pose de nouvelles questions, notamment pour l'estimation des paramètres. Devant ces difficultés, nous n'avons donc pas retenu cette solution.

2.1.2 Notations

Notons n_T le nombre de groupes formés par l'algorithme de classification non supervisée, qui correspondront plus intuitivement, dans les cas que nous étudierons, aux thèmes.

Nous notons n_D le nombre de documents dans le corpus d'apprentissage. Comme dit dans la section précédente, le vocabulaire est construit uniquement à partir des formes apparaissant au moins une fois dans le corpus d'apprentissage. Nous notons n_W le nombre de formes différentes, c'est-à-dire la taille du vocabulaire.

Nous supposons qu'en sortie de la phase de prétraitement, nous obtenons une matrice de comptes ou matrice mot/document C, de terme général C_{wd} avec $w \in \{1, \ldots, n_W\}$, $d \in \{1, \ldots, n_D\}$ et représentant le nombre d'occurrences du mot d'indice w dans le document d'indice d du corpus d'apprentissage.

$$C = \begin{array}{c} \\ \text{termes} \end{array} \begin{array}{c} \overset{\textit{documents}}{} \\ \begin{pmatrix} 10 & 6 & \ldots & 8 \\ 3 & 0 & \ldots & 2 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \ldots & 3 \end{pmatrix} \end{array}$$

Pour $d = 1, \ldots, n_D$, on note

$$l_d = \sum_{w=1}^{n_W} C_{wd}$$

le nombre d'occurrences dans le texte d et

$$l = \sum_{d=1}^{n_D} l_d$$

le nombre total d'occurrences dans le corpus d'apprentissage, somme de tous les termes de la matrice de comptes.

De façon similaire à ce qui précède, nous définissons pour le corpus de test : le nombre de documents $n_D{}^\star$, la matrice termes/documents $C^\star$, de terme général $C^\star_{wd^\star}$ (avec $d^\star \in \{1, \ldots, n_D{}^\star\}$), la longueur $l^\star_{d^\star}$ du document $d^\star$ et le nombre total d'occurrences dans le corpus de test $l^\star$.

2.2 Modèles non probabilistes

Le problème du partitionnement de données existe bien au-delà de la fouille de textes et de très nombreux modèles, éventuellement adaptés d'autres domaines, peuvent servir à regrouper des documents similaires. D'une façon générale, il suffit de définir une représentation des documents induisant une distance et une méthode pour diviser l'espace à partir de cette matrice de proximité. Ce point de vue engage à dissocier le problème de la classification de celui de la représentation des données. Par conséquent, dans un premier temps, nous présenterons les modèles de classification non supervisée les plus courants : algorithmes hiérarchique agglomératif et K-moyennes, avant de montrer dans quel espace il est judicieux de les appliquer par le biais de transformations sur les comptes.

Nous nous intéressons ensuite à quatre méthodes qui ont reçu beaucoup d'échos ces dernières années en apprentissage automatique et/ou en fouille de textes : l'analyse sémantique latente (LSA), le regroupement spectral, la factorisation en matrices non-négatives (NMF) et le goulot d'information (*information bottleneck*). La dernière sous-section présente brièvement d'autres idées relatives à la classification non supervisée et s'inscrivant dans des cadres non probabilistes.

2.2.1 Modèles généraux de classification non supervisée

Les modèles de classification non supervisée (voir [Jain et al., 1999] pour une synthèse) sont couramment divisés en deux classes : modèles hiérarchiques ou modèles de partitionnements.

Modèles hiérarchiques Le principe des modèles hiérarchiques est de procéder itérativement, en recherchant à chaque étape un groupe à diviser selon un certain critère qui dépend de la distance entre deux groupes [Cutting et al., 1992, Jain et al., 1999]. Supposons dans un premier temps que nous sachions calculer une telle distance. Il est alors possible de procéder par regroupement *agglomératif* :

- le point de départ est de considérer qu'il y a autant de groupes que d'exemples ($n_T = n_D$) ;
- à chaque étape, il s'agit de calculer les distances entre chaque couple de groupes et de fusionner les deux plus proches.

L'algorithme peut s'arrêter une fois qu'un nombre de thèmes déterminé a été atteint ou continuer jusqu'à obtention d'un groupe unique, ce qui permet d'exhiber une suite de partitionnement imbriqués les uns dans les autres pour un nombre de groupes quelconque

entre 1 et n_D. Le résultat peut se représenter de façon synthétique sous la forme d'un arbre dont les feuilles sont les documents. On appelle ce résultat un *dendrogramme* [Jain et al., 1999].

L'application de ces algorithmes repose donc entièrement sur la définition d'une distance entre deux groupes de documents τ_t et $\tau_{t'}$. Si nous supposons disposer d'une distance entre deux documents $d(C_d, C_{d'})$, par exemple l'opposé du produit scalaire entre les deux vecteurs de comptes (nous verrons dans la section suivante comment définir de meilleures représentations), les distances suivantes sont classiquement utilisées :

- *lien simple* (*single linkage*) : $d(\tau_t, \tau_{t'}) = \min_{d\in\tau_t, d'\in\tau_{t'}} d(C_d, C_{d'})$
- *lien complet* (*complete linkage*) : $d(\tau_t, \tau_{t'}) = \max_{d\in\tau_t, d'\in\tau_{t'}} d(C_d, C_{d'})$
- *lien moyen* (*average linkage*) : $d(\tau_t, \tau_{t'}) = \frac{1}{|\tau_t||\tau_{t'}|} \sum_{d\in\tau_t, d'\in\tau_{t'}} d(C_d, C_{d'})$

En classification non supervisée en général, les algorithmes de regroupement hiérarchique donnent de bonnes performances si des distances (entre observations, et entre groupes) pertinentes vis-à-vis des données et du problème sont sélectionnées, mais au prix d'un temps de calcul important (en l'absence d'heuristique astucieuse, il faudra calculer les distances entre chaque couple de points) [Jain et al., 1999].

K-moyennes Les algorithmes de partitionnement se distinguent des algorithmes hiérarchiques par le fait qu'ils produisent en général un regroupement unique pour un nombre de thèmes donnés et non un dendrogramme. Leur famille est plus générale que celles des algorithmes hiérarchiques et il est de ce fait difficile d'en donner une caractérisation précise. Des traits communs aux algorithmes de partitionnements sont qu'ils optimisent en général un critère sur l'ensemble des données de façon itérative et qu'ils sont souvent sensibles à la procédure d'initialisation [Jain et al., 1999]. Une discussion sur les algorithmes de partitionnement dans leur plus grande généralité dépasse le cadre de cette thèse et nous avons donc choisi de nous intéresser plus particulièrement à l'un d'entre eux : l'algorithme des *K-moyennes*.

De par sa simplicité et ses bons résultats sur des problèmes variés, l'algorithme des K-moyennes [MacQueen, 1967] reste l'une des méthodes les plus utilisées en classification non supervisée. Le principe des K-moyennes est que les documents et les thèmes sont des points d'un espace multidimensionnel, sur lequel nous supposons donnée une distance d. Les documents sont dans un premier temps affectés aléatoirement à chaque groupe. Puis le centroïde de chaque thème τ_t est calculé comme étant le barycentre de l'ensemble des individus de ce groupe. Il s'agit donc de vecteurs dans l'espace du vocabulaire, que nous notons $\beta_{t\bullet}$.

Les centroïdes et assignation des individus aux thèmes sont ensuite ajustés en alternance, comme le montre la figure 2.1, dans laquelle chaque document est représenté par son vecteur de comptes.

Comme pour les algorithmes de classification hiérarchiques, le choix de la représentation et de la distance sont primordiaux pour l'efficacité des K-moyennes. Pour les données textuelles représentées sous forme de comptes, la distance euclidienne étant manifestement inadaptée, il est d'usage de faire appel à la distance cosinus $d(C_d, \tau_t)$, calculée par :

$$d(C_d, \tau_t)^{-1} = \frac{\sum_{w=1}^{n_W} C_{wd}\beta_{tw}}{\sqrt{\sum_{w=1}^{n_W} C_{wd}^2}\sqrt{\sum_{w=1}^{n_W} \beta_{tw}^2}},$$

selon la définition classique du cosinus entre deux vecteurs, qui mesure leur similarité dans un espace euclidien. Cependant, il a été également observé que de meilleurs résultats

- Tirer selon une loi uniforme un thème pour chaque document dans l'ensemble $\{\tau_1, \ldots, \tau_{n_T}\}$
- Pour un certain nombre d'itérations, répéter :
  - Calcul des nouveaux barycentres :

$$\forall t \in \{1, \ldots, n_T\}, \forall w \in \{1, \ldots, n_W\}, \beta_{tw} = \frac{\sum_{d \in \tau_t} C_{wd}}{|\tau_t|}$$

  - Réaffectation des documents, pour $d \in \{1, \ldots, n_D\}$:

$$\arg\min_{t=1,\ldots,n_T} d(C_d, \tau_t)$$

FIG. 2.1 – Principe de l'algorithme des K-moyennes.

pouvaient être obtenus en choisissant d'autres représentations que les vecteurs de comptes non modifiés [Jurafsky and Martin, 2000, pages 646 à 654]. Nous abordons à présent ces questions de transformation des comptes.

2.2.2 Transformation des comptes

L'application des algorithmes de regroupement décrits dans les sections précédentes aux vecteurs de comptes donne des résultats mitigés [Jurafsky and Martin, 2000]. C'est la raison pour laquelle il est légitime de s'interroger sur l'opportunité d'utiliser d'autres représentations et d'autres distances. L'une des raisons des mauvaises résultats sur les vecteurs de comptes bruts, semble être un mauvais équilibre entre la représentation des mots les plus fréquents et celle des mots plus rares. Nous présentons dans cette section deux idées modifiant cet équilibre, l'une consistant à ignorer les comptes, l'autre à les pondérer de façon adaptée.

La représentation la plus simple, que nous avons jusqu'à présent négligé, fait abstraction de la longueur des textes : un document peut être vu comme un ensemble de n_W indicateurs binaires, chacun d'entre eux correspondant à la présence ou l'absence d'un mot du vocabulaire dans le document. Néanmoins, [McCallum and Nigam, 1998] montre en classification supervisée que ce modèle, connu sous le nom *Bernoulli multivarié*, donne systématiquement de moins bonnes performances, pour des vocabulaires de la taille du nôtre, qu'une représentation fondée sur le « sac de mots ». Cette analyse confirme l'intuition que le nombre d'occurrences d'un terme dans un texte est importante, au-delà du simple fait qu'il soit présent. Conjecturant que ces conclusions sont transposables au cas non supervisé, nous avons, par la suite, privilégié les modéles fondés sur les comptes.

Une autre remarque sur la représentation sous forme de comptes bruts est que la longueur des documents a une grande importance. De fait, si la distance considérée est le simple produit scalaire entre vecteurs de comptes, un texte contenant beaucoup d'occurrences obtiendra nécessairement des scores de similarité plus grands qu'un texte contenant exactement les mêmes mots mais avec des comptes plus petits. L'intuition dicte que cet effet de la longueur doit être compensé et c'est d'ailleurs ce que nous avons fait en section précédente : l'usage de la distance cosinus normalisée sur les comptes bruts est exactement équivalente à l'usage du produit scalaire sur les vecteurs de comptes normalisés par la norme $\mathcal{L}_2$. Notons ici que la normalisation par la norme $\mathcal{L}_1$ a elle pour effet de diviser les

comptes par la longueur du document, pour obtenir des *profils* sur le vocabulaire. Nous n'avons cependant pas utilisé cette normalisation, moins naturelle lorsque l'on choisit la distance cosinus.

Les normalisations ne suffisent pas à compenser les effets de déséquilibre entre les comptes évoqués ci-dessus. À cet effet, nous introduisons à présent la *fréquence inversée sur documents*, appelée plus couramment *idf* (pour *inverse document frequency*). Ce facteur modificateur, qui connaît de nombreuses variantes, corrige une lacune de la représentation sous forme de vecteurs de comptes : les mots qui présentent les comptes les plus forts ne sont pas forcément les plus pertinents. Ainsi, les anti-mots, qui apparaissent dans tous les documents de façon très fréquente, sont en général tout-à-fait dénués d'intérêt pour la classification thématique. L'idée est de pénaliser ces termes apparaissant dans tous les documents (et, parallèlement, de récompenser les mots plus rares) par l'idf égale à :

$$\forall w \in \{1, \ldots, n_W\}, \mathrm{idf}(w) = \log \frac{n_D}{\sum_{d=1}^{n_D} \mathbb{1}_{\{C_{wd}>0\}}}$$

Ce facteur connaît de nombreuses autres définitions que celle que nous présentons ici mais le principe est toujours le même : pour chaque mot, il est d'autant plus grand que le terme apparaît dans peu de documents, ou, en d'autres termes, à nombres d'occurrences égaux, il privilégie les mots pour lesquelles les apparitions sont les plus concentrées dans un nombre limité de documents. Il faut en particulier noter que, pour les mots figurant dans tous les documents, nous avons $\sum_{d=1}^{n_D} \mathbb{1}_{\{C_{wd}>0\}} = n_D$, et donc $\mathrm{idf}(w) = 0$.

L'usage de cette représentation, connue sous le nom de *TFIDF* est courante en classification non supervisée, en particulier avec l'algorithme des K-moyennes associé à une distance cosinus [Jurafsky and Martin, 2000]. [Steinbach et al., 2000] rapporte des expériences selon lesquelles l'algorithme des K-moyennes est plus rapide et obtient de meilleures performances que les algorithmes hiérarchiques. L'algorithme des K-moyennes avec TFIDF et la distance cosinus constitue donc une bonne référence pour juger des performances d'un algorithme de classification non supervisée de documents. Dans la section 5.1, nous présenterons une telle comparaison.

2.2.3 Analyse sémantique latente (LSI/LSA)

L'indexation sémantique latente (Latent Semantic Indexing, LSI) [Deerwester et al., 1990] s'inscrit dans une optique traditionnelle en analyse de données. Il s'agit de la décomposition de matrices selon leurs directions propres (ou singulières) pour conserver un maximum d'« information » sur un nombre minimum de dimensions. *Analyse des correspondances* [Benzécri et al., 1981, Lebart and Salem, 1988], *analyse en composantes principales* ou *échelonnement multidimensionnel* (*multidimensional scaling* [Kruskal, 1964] sont autant de techniques faisant appel à des décompositions en valeurs singulières (voir [Chatfield and Collins, 1980] pour une synthèse), elles-mêmes équivalentes à des décompositions en valeurs propres d'une matrice carrée associée (matrice de similitude, de covariance empirique, etc). Nous allons voir que c'est le même principe qui est à l'œuvre ici, même si les motivations premières des auteurs de l'indexation sémantique latente viennent du domaine de la recherche d'information. Deerwester et al. constatent que la technique la plus simple, consistant à répondre à une requête d'utilisateur en retournant les documents qui contiennent le plus d'occurrences des mots contenus dans la requête, se heurte à des limites liées à des problèmes de polysémie et de synonymie :

- il est possible que dans certains documents du corpus, les termes de la requête soient présents mais employés dans un autre sens que celui recherché par l'utilisateur (renvoi de documents non pertinents, dégradation du score de précision[5]) ;
- il arrive à l'inverse que le mot demandé ne se trouve pas dans un document pourtant pertinent pour le thème car c'est un synonyme qui est employé (oubli de documents pertinents, dégradation du score de rappel).

L'hypothèse de Deerwester et al. est que trouver un moyen de considérer le contexte d'un mot et les liens sous-jacents entre des termes dans le corpus permet de régler en partie ces problèmes, au moins celui de la synonymie. Si, par exemple, l'utilisateur recherche le mot *voiture* et qu'un document utilise uniquement le terme *automobile*, le fait que d'autres termes du même champ lexical tels que *moteur* ou *carosserie* soient, d'une part, présents en nombre dans ce texte et, d'autre part, en cooccurrence fréquente avec le terme de la requête *voiture* ailleurs dans le corpus, permet d'affirmer que le texte est probablement pertinent.

Pour découvrir une telle structure *latente* dans le corpus, l'idée est de faire appel à une décomposition en valeurs singulières (*Singular Value Decomposition*, SVD) de la matrice de comptes C. On suppose que C est de rang m. C est alors décomposable en valeurs singulières, c'est-à-dire qu'il existe des matrices orthonormales U $(n_W \times m)$ et V $(n_D \times m)$ et une matrice diagonale S $(m \times m)$, de valeurs propres strictement positives et rangées en ordre décroissant, telles que :

$$C = USV^T$$

L'avantage de cette représentation est qu'elle fournit un nouvel espace de dimension m pour représenter C, les valeurs singulières sur la diagonale de S illustrant l'« importance » des différentes dimensions. Nous savons que m est inférieur à $\min(n_W, n_D)$ mais cela ne garantit en rien qu'il soit petit. Or remplacer la représentation sous forme de comptes par une autre n'a pas grand intérêt si les dimensions des matrices sont sensiblement les mêmes[6]. Par conséquent, l'étape suivante est de réduire substantiellement la dimension de l'espace latent, tout en perdant le moins d'« information » possible. Une solution consiste à supprimer les dimensions associées aux valeurs propres les plus faibles.

Plus formellement, pour tout $k = 1, \ldots, m$, on note $C^{(k)} = U^{(k)} S^{(k)} {V^{(k)}}^T$, avec :

- $U^{(k)}$ $(n_W \times k)$, k premières colonnes de U ;
- $S^{(k)}$ $(k \times k)$, matrice diagonale correspondant à S sans ses $m-k$ plus petites valeurs propres ;
- $V^{(k)}$ $(k \times n_D)$, k premières lignes de V.

La matrice $C^{(k)}$ est une approximation de C dont on peut montrer qu'elle est la meilleure de rang k au sens de la norme $\mathcal{L}^2$, c'est-à-dire encore celle qui retient le mieux la variance des données initiales.

La décomposition en valeurs singulières est liée à la diagonalisation des matrices carrées symétriques réelles CC^T et C^TC, ce que l'on constate assez facilement avec les formules suivantes :

$$C^TC = VS^2V^T$$

$$CC^T = US^2U^T$$

[5] Voir le chapitre 3 pour la définition des mesures de précision et de rappel.

[6] Au contraire, nous y verrions plutôt, intuitivement, des inconvénients, notamment le risque de perdre le précieux caractère creux (*sparse*) de la matrice de comptes, c'est-à-dire le fait qu'elle affiche beaucoup plus de zéros que d'éléments non nuls.

Or, C^TC est une matrice de similarité entre les documents alors que CC^T est une matrice de similarité entre les mots (vus comme des vecteurs de documents). Avec les approximations ci-dessus, ${C^{(k)}}^T C^{(k)}$ et $C^{(k)}{C^{(k)}}^T$ nous donnent donc un moyen de comparer deux éléments dans le nouvel espace. [Deerwester et al., 1990] explique aussi comment projeter un nouveau vecteur de mots (requête ou document) C_d dans le nouvel espace : sa représentation est déterminée par ${S^{(k)}}^{-1}{U^{(k)}}^T C_d$[7].

L'évaluation porte sur une tâche de recherche d'information : la décomposition en valeurs singulières est appliquée à la matrice de comptes d'un corpus pour lequel sont disponibles des couples requête/liste de documents pertinents. Les requêtes sont ensuite projetées dans l'espace latent et les documents retenus sont ceux qui sont les plus proches dans ce nouvel espace. Les résultats sont présentés sous forme de courbes précision/rappel.

Ils sont modérément encourageants. LSI semble toujours faire au moins aussi bien qu'une méthode basique de renvoi par mots communs (*term matching*) mais se comporte de façon inégale par rapport à d'autres techniques plus évoluées (*SMART* [Salton et al., 1975] et *cluster-based retrieval* [Voorhees, 1985]). Les auteurs soulignent que les différences entre les étapes de prétraitement (filtrage, document fréquence inverse, racinisation...) peuvent avoir une influence non-négligeable sur la performance finale. Ils suggèrent, par conséquent, que LSI tel qu'ils le présentent (avec un simple filtrage des mots trop rares et communs) devrait être précédé et/ou suivi d'autres techniques de recherche d'information pour obtenir un système complet avec des performances supérieures à l'état-de-l'art sur tous les corpus de test.

Certains problèmes ne sont évoqués que brièvement et restent à approfondir :

- introduction de nouveaux documents dans la base sans ré-effectuer la SVD entière[8] ;
- longueurs inhomogènes des différents documents (ou comment relativiser l'importance de textes très longs) ;
- choix du nombre k de thèmes à conserver.

Les développements ultérieurs seront beaucoup plus positifs que les résultats présentés dans cette première étude. Ainsi, [Dumais, 1990] montre qu'un effort sur l'étape de prétraitement permet de gagner sensiblement en performances (une analyse proche du constat que nous réalisons en section 5.1 à propos de l'algorithme des K-moyennes). En particulier, l'utilisation de comptes modifiés par la fonction `log` et une pondération par un coefficient d'entropie[9] permettent de rendre LSI compétitif avec l'état de l'art en recherche d'information. Ces bons résultats seront confirmés dans les campagnes d'évaluation TREC (voir, par exemple, [Dumais, 1994]).

L'utilisation de la décomposition en valeurs singulières de la matrice de comptes peut être réalisée dans d'autres buts que l'indexation. Par exemple, il s'agira pour nous d'obtenir les thèmes dominants dans le corpus, chacun étant associé à un sous-espace singulier. Nous parlerons donc plutôt dans la suite de LSA (*Latent Semantic Analysis*), ce qui désigne

[7] Ce résultat est lié au fait à la recherche d'une correspondance entre les colonnes de C, c'est-à-dire les vecteurs de termes, et les colonnes de V^T, représentations des textes dans l'espace singulier. La formule découle immédiatement de la décomposition : $V = (US)^{-1}C = S^{-1}U^TC$, U étant orthonormale et S supposée de rang plein.

[8] Si l'on note C_d le vecteur colonne à ajouter, le problème consiste à décomposer la matrice $(C|C_d)$ $(n_W \times n_D + 1)$ (qui est en fait égale à $CC^T + C_dC_d^T$) en utilisant les résultats obtenus sur C. Des méthodes pour résoudre ce problème baptisées *SVD incrémentale* ou *ajustement de rang 1* consistent à séparer l'espace selon l'hyperplan orthogonal à C_d et à utiliser le fait que, dans cet espace, la matrice de comptes est toujours C. Voir [Brand, 2002] pour des avancées récentes sur le sujet.

[9] En pratique, ce terme pénalise les mots qui apparaissent dans de nombreux documents, à la manière de la *fréquence des documents inversée* (idf) (voir section 5.1).

simplement l'opération de décomposition en valeurs singulières sans but d'indexation.

Il est surprenant de constater que l'étude de LSA a profité à bien d'autres domaines, parfois éloignés de l'indexation. Citons notamment :

- en psychologie/sciences cognitives, l'application de LSA peut être reliée à l'énigme dite « de Platon », qui est que la vitesse d'acquisition du vocabulaire par les individus, et en particulier les enfants, semble nettement supérieure au nombre de nouveaux mots rencontrés par jour. [Landauer and Dumais, 1997] montre, par des résolutions automatiques de tests de type TOEFL avec des corpus censés imiter l'ensemble des textes auxquels a pu être exposé un enfant à un âge donné, qu'une grande partie de l'information sur des mots inconnus peut être inférée de façon contextuelle au moyen d'un modèle comme LSA ;
- pour la résolution de problèmes (*problem solving*), les mots étant remplacés par des actions dans un monde clos et les documents devenant des séquences de résolution du problème issues de l'ensemble d'apprentissage, l'apport de LSA est discuté dans [Quesada et al., 2002].

L'article [Landauer et al., 1998] recense l'ensemble des applications de LSA, des questionnaires de synonymie à la correction automatique de rédactions. D'autre part, l'université du Colorado à Boulder a développé une page web `http://lsa.colorado.edu` sur laquelle il est possible de tester en ligne l'analyse sémantique latente sur divers corpus (dont certains en français) notamment pour trouver des associations de mots ou effectuer des comparaisons de termes ou de documents.

[Papadimitriou et al., 1998] essaie de justifier théoriquement l'utilisation de LSA pour la recherche d'information. En proposant un modèle génératif de corpus, le théorème principal de l'article prouve que, lorsque les vocabulaires propres à chaque thème contiennent suffisamment de mots, les vecteurs associés à deux documents donnés forment, dans l'espace latent, un angle très faible s'ils appartiennent au même thème et presque droit s'ils appartiennent à deux thèmes différents et ce, avec une probabilité d'autant plus forte que le nombre de documents considéré est grand.

La preuve est d'abord faite dans le cas où les thèmes sont complètement disjoints puis étendue à un cas plus général grâce à un théorème concernant la stabilité de la décomposition en valeurs singulières à de faibles perturbations (c'est-à-dire dont la norme est contrôlée). Mais, même dans ce cas, les hypothèses requises par l'article :

- disjonction des vocabulaires associées à chaque thème ;
- étalement d'un thème sur un nombre suffisamment grand de mots dominants ;
- grand nombre de textes dans le corpus

semblent encore très fortes par rapport à la qualité connue des performances de LSA dans des cas bien moins favorables. Pour assouplir un peu la dernière condition, les auteurs montrent dans la dernière partie de l'article qu'en appliquant LSA sur un sous-ensemble de documents choisis au hasard (*random projection*), on obtient une approximation satisfaisante de la classification obtenue avec le corpus entier.

2.2.4 Regroupement spectral, méthodes à noyaux

L'analyse sémantique latente conduit à effectuer une décomposition en valeurs propres (ou analyse en composantes principales) sur la matrice des produits scalaires des vecteurs de comptes (C^TC). Le *regroupement spectral* part d'un point de vue plus général, en supposant donnée une matrice W de dimensions $n_D \times n_D$ d'*affinité* entre éléments. Dans le cas de LSA, l'affinité entre deux textes d et d' était donc $\sum_{w=1}^{n_W} C_{wd}C_{wd'}$ alors que le regroupe-

ment spectral choisit plus volontiers des définitions telles que $W_{dd'} = e^{-||C_d - C_{d'}||^2/2\sigma^2}$, plus adaptées à la segmentation d'image, application d'origine de cette famille d'algorithmes. Le principe général est de grouper les données à partir de l'étude des vecteurs et valeurs propres de W. Les différentes méthodes de regroupement spectral diffèrent par la matrice considérée, les étapes de normalisation et le ou les sous-espaces propres utilisés [Weiss, 1999].

L'idée générale est décrite de façon synthétique dans [Kannan et al., 2000]. La première étape consiste à procéder à une décomposition de W en composantes principales et à garder autant de vecteurs propres $u_1, \ldots, u_{n_T}$ que de groupes souhaités, en les classant naturellement par valeurs propres décroissantes. Si l'on note ensuite w_d la d-ième ligne de W, le regroupement consiste à projeter chaque ligne dans les sous-espaces propres et à affecter l'élément d au thème t qui maximise le produit scalaire $u_t^T w_d$. Un prétraitement classique est de normaliser la matrice d'affinité par la somme des poids affectés à chaque document. Cette opération permet de tenir compte de l'aspect des données dans leur globalité, pour éviter les groupes trop petits, résultant de particularités locales [Shi and Malik, 1997]. D'autres variantes [Ng et al., 2002] suggèrent d'appliquer des méthodes d'affectation plus évoluées aux données projetées dans les sous-espaces propres, par exemple un algorithme des K-moyennes (voir section 5.1).

Lorsque le choix de l'affinité est particulièrement bien adapté aux données, l'intérêt de la démarche est très clair : alors que les regroupements n'étaient pas visuellement apparents dans l'espace de départ, ils peuvent devenir évidents dans un espace d'arrivée judicieusement choisi [Ng et al., 2002]. Cette idée, connue sous le nom d'« astuce du noyau » (*kernel trick*), et popularisée dans l'application des séparateurs à vaste marge (*support vector machines*) [Boser et al., 1992], permet d'exporter avec succès à des cadres non linéaires des algorithmes a priori conçus pour des données linéairement séparables [Shawe-Taylor and Cristianini, 2004].

L'idée générale des méthodes à noyaux est que certains algorithmes n'ont besoin d'accéder aux données que par le biais des produits scalaires entre exemples. Ainsi la matrice symétrique des produits scalaires entre chaque paire d'exemples (dite *matrice de Gram*) regroupe l'ensemble des informations à propos du corpus d'apprentissage dont ces méthodes ont besoin. Par conséquent, si nous sommes capables de « déformer » la structure de l'espace en proposant de nouveaux produits scalaires entre exemples (c'est-à-dire un nouveau *noyau*), plus pertinents que le produit scalaire classique, nous pouvons également transposer tous ces algorithmes initialement dédiés au cas linéaire. L'astuce du noyau a été utilisée dans un très grand nombre d'applications en apprentissage statistique.

[Cristianini et al., 2001] décrit une autre façon d'utiliser les résultats de LSA dans le cadre des méthodes à noyaux que le regroupement spectral. Alors que, dans ce dernier, l'enjeu est de trouver un espace pertinent dans lequel appliquer l'analyse en composantes principales, l'idée de [Cristianini et al., 2001] est d'utiliser LSA comme un prétraitement permettant de définir un noyau sémantiquement pertinent à d'autres fins. Les auteurs décrivent d'abord brièvement certains modèles utilisés en analyse textuelle sous l'angle « noyaux », notamment le *modèle vectoriel* [Salton et al., 1975] et des variantes plus évoluées prenant en compte les cooccurrences entre termes ou les relations synonymiques. Puis ils remarquent que le processus de décomposition en valeurs singulières de LSA permet de calculer une nouvelles similarité entre documents (dans l'espace latent) et donc un nouveau noyau : le *noyau latent sémantique*. L'intuition de ce travail est déjà présente dans [Deerwester et al., 1990] mais elle est ici expliquée de façon plus systématique puis utilisée en composition (noyaux polynomiaux ou gaussiens) dans des séparateurs à vaste marge

(SVM). Conscients des difficultés numériques liées à l'application de la décomposition en valeurs singulières en grande dimension, les auteurs proposent un algorithme approximatif utilisable dans des cas réels, c'est-à-dire lorsque le nombre de documents est grand. Cet algorithme est ensuite modifié dans les dernières sections par une heuristique accordant plus d'importance aux documents les plus pertinents.

Les résultats sont mitigés : bien que l'utilisation de noyaux spécifiques donne des performances en général au moins aussi bonnes que le noyau linéaire, elles ne font beaucoup mieux que dans certains cas très précis et la lecture de l'article ne permet pas de déterminer clairement dans quelles conditions. Cette étude demeure intéressante car elle présente une utilisation originale de LSA et, plus généralement, démontre en quoi une méthode d'analyse non supervisée peut fournir un cadre de travail plus général pour l'apprentissage, en préambule à d'éventuelles méthodes supervisées (voir aussi [Slonim and Tishby, 2001, Vinot and Yvon, 2003] à ce sujet).

Pour conclure sur le regroupement spectral et les méthodes à noyaux, signalons que, malgré quelques tentatives de noyaux plus évolués, il est difficile d'obtenir pour le texte des performances significativement meilleures que le noyau linéaire [Joachims, 1998]. Voir [Renders, 2004] pour une étude des méthodes à noyaux orientée vers l'application qui nous intéresse ici. Devant les difficultés pour choisir un noyau, la solution d'en apprendre un parmi une famille de façon supervisée à partir de corpus a été proposée pour d'autres domaines [Bach and Jordan, 2004]. Mais, là encore, l'applicabilité de la méthode à l'analyse de données textuelles semble compliquée à établir.

2.2.5 Factorisation en matrices non-négatives (NMF)

L'application de la factorisation en matrices non-négatives [Lee and Seung, 2001] ne produit pas directement un partitionnement mais consiste à trouver une approximation matricielle optimale (caractéristique qu'elle partage avec LSA) sous une contrainte de positivité. Il s'agit de déterminer deux matrices W et H de dimensions respectives $n_W \times n_T$ et $n_T \times n_D$ et de termes tous positifs ou nuls telles que :

$$C \approx WH$$

Nous détaillons plus loin l'application à la classification non supervisée de documents mais donnons d'ores et déjà l'intuition derrière cette décomposition. Chaque colonne de la matrice de comptes C_d est approchée par une somme pondérée (avec uniquement des poids positifs) de vecteurs sur le vocabulaire, les colonnes de W, qui représentent les « thèmes ». La matrice H, qui contient les pondérations, représente l'« influence » de chaque thème dans les documents.

Le symbole $\approx$ est volontairement vague ici puisque la notion d'approximation est équivalente à celle de minimisation d'une fonction de coût et que Lee et Seung en proposent deux pour deux matrices A et B :

- La distance euclidienne : $\|A - B\|^2 = \sum_{i,j}(A_{ij} - B_{ij})^2$
- La divergence de Kullback-Leibler : $D(A\|B) = \sum_{i,j}(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij})$ avec les prolongements par continuité classiques et en admettant que cette divergence puisse être infinie.

Le fait de proposer deux distances permet de choisir celle qui semble le mieux convenir au problème, ou, le cas échéant d'en rejeter une qui est manifestement inadaptée. Par exemple,

une critique à l'encontre de LSA, formulée notamment par Hofmann [Hofmann, 2001, et section 2.3], est que la minimisation de la distance euclidienne repose sur des hypothèses sous-jacentes de normalité sur les données qui ne semblent pas forcément justifiées dans le cadre de matrices de comptes d'occurrences de mots. C'est pourquoi il lui préfère la divergence de Kullback-Leibler.

Les théorèmes fondamentaux de l'article donnent des règles multiplicatives desquelles se déduisent des algorithmes itératifs pour minimiser localement[10] chacune des distances.

- La distance euclidienne $\|C - WH\|$ décroît si l'on applique itérativement les règles de mise à jour :

$$H_{td} \leftarrow H_{td} \frac{(W^T C)_{td}}{(W^T W H)_{td}}$$

$$W_{wt} \leftarrow W_{wt} \frac{(C H^T)_{wt}}{(W H H^T)_{wt}}$$

- La divergence $D(C\|WH)$ décroît si l'on applique itérativement les règles de mise à jour :

$$H_{td} \leftarrow H_{td} \frac{\sum_{w=1}^{n_W} W_{wt} C_{wd}/(WH)_{wd}}{\sum_{w=1}^{n_W} W_{wt}}$$

$$W_{wt} \leftarrow W_{wt} \frac{\sum_{d=1}^{n_D} H_{td} C_{wd}/(WH)_{wd}}{\sum_{d=1}^{n_D} H_{td}}$$

Lee et Seung remarquent que ces règles de mise à jour, qui en apparence ne sont pas liées à l'algorithme de descente de gradient, consistent en fait à suivre le gradient selon un pas garantissant les contraintes de positivité et de croissance de la fonction d'objectif.

Le reste de l'article établit les preuves de convergence. On remarque d'abord que l'on peut minimiser itérativement en H, à W constant, et en W, à H constant. On définit ensuite une fonction auxiliaire à deux variables G telle que, si F est la distance à minimiser,

$$\forall h, h', \begin{cases} G(h,h') \geq F(h) \\ G(h,h) = F(h) \end{cases},$$

avec G telle que $G(h,h') - F(h)$ se réduit à une fonction quadratique positive de $(h - h')$.

On peut constater que, dans la multiplication matricielle de W par H, pour $t = 1, \ldots, n_T$, les éléments la t-ième colonne de W sont toujours multipliés par ceux de la t-ième ligne de H. On peut multiplier la ligne t de H par n'importe quel réel non nul λ_t sans changer le produit final, pour peu que l'on multiplie la colonne correspondante de W par $\frac{1}{\lambda_t}$. Il est possible de normaliser soit les lignes de H soit les colonnes de W pour qu'elles somment à 1. Dans le premier cas, on peut supposer que les éléments de H représentent la probabilité d'appartenance à un thème pour un document donné[11]. Dans le second cas, il s'agit de la probabilité pour chaque mot de représenter un thème. D'où une interprétation assez intuitive des résultats dans le cadre de la fouille de textes : la matrice H définit un partitionnement probabiliste des documents sur n_T dimensions.

[Lee and Seung, 2001] reste peu disert sur les aspects pratiques de mise en œvre de l'algorithme, que ce soit sur les spécificités des divers domaines d'application, les stratégies d'initialisation, de normalisation et de choix du nombre de dimensions latentes (ici la dimension commune de W et H).

[10] Le problème n'étant pas simultanément convexe en W et H, il ne faut pas s'attendre à trouver un minimum global.

[11] La formulation du problème élément par élément évoque alors un modèle de mélange sur les documents.

[Xu et al., 2003] aborde directement l'application de NMF à la classification non supervisée de textes, en mettant l'accent sur une différence importante avec LSA : les sous-espaces latents ne sont pas forcément orthogonaux et sont susceptibles de recouvrements, ce qui est souhaitable pour de nombreux corpus. Les performances de l'algorithme sont comparées à des méthodes de regroupement spectral sur deux corpus au moyen d'un score de cooccurrences (après une étape d'appariement des groupes, voir section 3.5.3) et d'un score d'information mutuelle. Les capacités de classification de NMF semblent plus ou moins équivalentes à celles des algorithmes de regroupement spectral retenus, les résultats étant améliorés par une normalisation analogue à celle que nous avons évoquée pour la matrice d'affinité .

[Vinokourov, 2002] donne également des résultats pratiques des performances de NMF sur une tâche de recherche d'information, montrant sa supériorité sur LSI en termes de précision/rappel de documents pertinents. Une des raisons avancées par Vinokourov en est que NMF encourage les représentations creuses (c'est-à-dire contenant beaucoup de zéros)[12] et cette représentation est plus fidèle à la matrice de comptes, qui est également creuse. Le même article propose une façon probabiliste de retrouver les équations de mise à jour de NMF, avec l'hypothèse d'un bruit poissonien et de distributions a priori laplaciennes pour W et H, par une approximation variationnelle de la distribution initiale. Cette première tentative d'analogie entre NMF et d'autres algorithmes de classification non supervisée sera suivie de preuves de similarités fortes avec, notamment, le regroupement spectral [Ding et al., 2005] et PLSA [Gaussier and Goutte, 2005]. Ces relations seront étudiées en section 2.4.

NMF a fait l'objet d'une attention considérable depuis son introduction en 1999, en particulier dans le domaine du traitement de l'image, où la décomposition d'un objet en plusieurs « parties » est intuitivement attractive. Un problème reste néanmoins préoccupant, il s'agit de l'absence théorique d'unicité de la décomposition. Il est facile de vérifier que toute permutation identique des colonnes de W et des lignes de H laisse le produit inchangé, tout comme la multiplication simultanée de la colonne t de W par une constante $\alpha_t > 0$ et de la ligne t de H par $\frac{1}{\alpha_t}$. Cette dernière remarque offre en fait une marge de manœuvre pour normaliser H, ou W, suivant l'application et l'interprétation souhaitées [Xu et al., 2003]. Mais, plus généralement, à partir d'une solution W et H, il suffit de trouver une matrice A ($n_T \times n_T$) inversible et telle que $\tilde{W} = WA$ et $\tilde{H} = A^{-1}H$ soient encore positives pour obtenir une autre solution, potentiellement non trivialement différente, et telle que $\tilde{W}\tilde{H} = WH$. [Donoho and Stodden, 2004] s'interroge sur ce problème d'unicité de la décomposition (aux permutations et à l'échelle près) d'un point de vue théorique et propose une preuve dans un cas particulier de décomposition « par morceaux » qui semble très orienté vers le traitement de l'image. Un autre résultat d'unicité concerne une extension de l'algorithme NMF pour l'obtention de résultats dans le cas de matrice creuse (*sparse NMF* [Hoyer, 2004]). Les conditions requises sont plus faibles que la restriction à une famille mais restent relativement limitées ($n_T = 2$ ou $\tilde{W}^{-1}W \geq 0$ par exemple) [Theis et al., 2005]. Cette non-unicité de la décomposition reste donc un problème partiellement ouvert : dès lors que la solution préconisée est celle d'un algorithme itératif convergeant vers un minimum local et qu'il est difficile de proposer des stratégies d'initialisation exemptes de tout reproche, la présence de différents points de convergence possibles pose de sérieuses questions de stabilité et d'interprétabilité des résultats.

[12] Du fait des contraintes de positivité, les solutions obtenues au moyen d'algorithmes itératifs ont des chances de se trouver sur certaines frontières du domaine de recherche, c'est-à-dire là où des éléments s'annulent.

2.2.6 Goulot d'information

La méthode du « goulot d'information » (*information bottleneck*) [Slonim and Tishby, 2000] tire son inspiration de la théorie de l'information et semble donner de bons résultats pour l'analyse exploratoire. Le goulot d'information suit deux idées principales :

- les problèmes de sélection d'attributs sur le vocabulaire et de classification non supervisée de documents sont fondamentalement similaires : il s'agit de réduction de dimensionnalité dans les deux cas, selon les deux directions de la matrice mots/documents ;
- l'information mutuelle permet de définir un critère mesurant la perte d'information et, donc, la qualité d'un partitionnement (sur les documents comme sur les mots).

Le principe est d'effectuer un regroupement sur la base d'un critère de maximisation d'information mutuelle d'une variable-cible. Il s'agit ainsi de conserver autant d'information que possible sur la distribution des mots en remplaçant la classification triviale consistant à placer un seul document par groupe (n_D groupes au total) par un regroupement plus restreint de thèmes. La méthode permet, notamment, d'estimer les probabilités des thèmes conditionnellement aux documents qui définissent une classification probabilisée des textes parmi n_T groupes.

Notons W, D et T des variables aléatoires indicatrices respectivement des mots, des documents ou des thèmes (définis au niveau des documents) et à valeurs dans $\{1, \ldots, n_W\}$, $\{1, \ldots, n_D\}$ et $\{1, \ldots, n_T\}$. Dans un souci de cohérence avec les modèles probabilistes présentés en section 2.3, nous introduisons également des notations spécifiques pour certaines probabilités :

$$\begin{aligned} \alpha_t &= P(T = t) \\ \beta_{tw} &= P(W = w | T = t) \\ \mu_{dt} &= P(T = t | D = d) \end{aligned}$$

D'autre part, la loi jointe $P(W = w, D = d)$ est supposée connue (ce qui donne aussi, de fait, l'expression des lois marginales et conditionnelles concernant W et D). Elle sera en pratique estimée par échantillonnage sur la matrice de comptes.

L'information mutuelle entre T et W est une quantité symétrique mesurant l'information que la connaissance de la variable aléatoire « thème » apporte sur la variable « mot ». Elle est définie par :

$$I(T; W) = \sum_{w=1}^{n_W} \sum_{t=1}^{n_T} P(T = t) P(W = w | T = t) \log \frac{P(W = w | T = t)}{P(T = t)}$$

Le problème est donc de trouver le regroupement thématique défini par les probabilités relatives à T : α, β et μ telles que l'information mutuelle $I(T; W)$ soit maximisée, tout en contrôlant l'information mutuelle par rapport à D, $I(T; D)$[13]. Plus précisément, la quantité minimisée par rapport aux distributions conditionnelles de T dans [Tishby et al., 1999] est définie par :

$$\mathcal{Q} = I(T; D) - \lambda I(T; W) \,,$$

λ étant un paramètre analogue à un « taux de compression » opérant un compromis entre un regroupement des documents le plus synthétique possible ($I(T; D)$ petit) et la conservation d'un maximum d'information concernant la distribution des mots ($I(T; W)$ grand).

[13] Il s'agit de s'assurer que le nombre de groupes n_T est significativement plus petit que le nombre de documents n_D et que, par conséquent, l'opération de classification a un intérêt.

Cette minimisation est accomplie par un algorithme itératif. À α_t et β_{tw} fixés, μ_{dt} se déduit par l'expression :

$$\mu_{dt} = \frac{\alpha_t}{Z(\lambda, d)} e^{-\lambda D_{KL}(P(W=w|D=d)||P(W=w|T=t))} \tag{2.1}$$

avec $Z(\lambda, d) = \sum_{t=1}^{n_T} \alpha_t e^{-\lambda D_{KL}(P(W=w|D=d)||P(W=w|T=t))}$ constante de normalisation et $D_{KL}(P(W = w|D = d)||P(W = w|T = t))$ divergence de Kullback-Leibler :

$$\sum_{w=1}^{n_W} P(W = w|D = d) \log \frac{P(W = w|D = d)}{P(W = w|T = t)} .$$

À μ_{dt} fixé, α_t et β_{tw} sont donnés par :

$$\alpha_t = \sum_{d=1}^{n_D} \mu_{dt} P(D = d) \tag{2.2}$$

$$\beta_{tw} = \frac{1}{\alpha_t} \sum_{d=1}^{n_D} P(W = w, D = d) \mu_{dt} \tag{2.3}$$

L'obtention de ces équations est détaillée en annexe A.2. On obtient donc un ensemble de mises à jour (2.1), (2.2) et (2.3) qui convergent vers un minimum local de la fonction objectif. Le lecteur familier de l'algorithme Espérance-Maximisation (EM) aura noté une certaine similarité, nous verrons en section 2.4.1, avec l'étude de [Slonim and Weiss, 2003], que ce n'est pas seulement un air de famille. Concernant l'application pratique de l'algorithme, une hypothèse contestable est l'estimation des probabilités relatives à W et D par échantillonnage sur la matrice de comptes et, plus précisément,

$$\widehat{P}(W = w, D = d) = \frac{C_{wd}}{l}$$

Cette estimation grossière, sans aucune forme de lissage, est probablement mauvaise pour les paramètres associés aux mots rares, pour lesquels on ne dispose que d'un nombre réduit d'observations.

La deuxième remarque importante concernant la méthode du goulot d'information est que les rôles joués par W et D sont absolument symétriques d'un point de vue formel et qu'il est donc possible d'utiliser un algorithme itératif analogue pour former des groupes sur les mots. C'est ainsi que [Slonim and Tishby, 2000] introduit la méthode du *double partitionnement* (*double clustering*) : dans une première étape de l'algorithme, il s'agit d'établir des groupes de mots préservant le plus d'information possible (par rapport au vocabulaire initial) à propos des documents et, par la suite, de former des partitions de documents qui minimisent la perte d'information par rapport à ces groupes de mots. Intuitivement, il s'agit de réduire alternativement les deux dimensions de la matrice de comptes en préservant à chaque étape le plus d'information mutuelle possible vis-à-vis de l'autre dimension.

Dans la mesure où les optimums trouvés ne sont que locaux, l'initialisation joue un rôle crucial, ce que confirment [Slonim and Tishby, 2000, Slonim et al., 2002]. Les auteurs ont surtout travaillé dans le cas de la classification « déterministe » et proposent différentes stratégies, reposant sur le tirage de partitions aléatoires et les transferts d'éléments améliorant la fonction objectif. Le caractère « ad hoc » de ces heuristiques d'initialisation

rend l'application de la méthode moins convaincante. Par ailleurs, le problème du choix de nombre de thèmes n'est pas abordé dans l'application de la méthode et les outils de théorie de l'information ne donnent pas, a priori, de moyen naturel de le régler.

Sur le plan pratique, les performances sont évaluées par similarité avec une classification non supervisée, l'appariement thème/catégorie étant effectué « manuellement » (ce qui n'est possible que lorsque le nombre de thèmes est réduit, et est de toute façon moins satisfaisant que la méthode hongroise développée en section 3.5.3). La méthode du goulot d'information produit les meilleurs résultats mais les algorithmes (fondés sur des distances $\mathcal{L}_1$ et $\mathcal{L}_2$) auxquels elle est comparée semblent de toute façon peu performants sur les applications textuelles, en témoignent les bons classements de méthodes simples telles que l'algorithme des K-moyennes et une heuristique fondée sur la représentation TFIDF [Slonim and Tishby, 2000]. L'idée du double partitionnement n'en est pas moins digne d'intérêt et les auteurs le démontrent en l'appliquant à tous les algorithmes utilisés. Il en résulte systématiquement un gain en qualité de classification.

[Slonim and Tishby, 2001] teste les bénéfices d'une étape de classification non supervisée des termes en vue d'un objectif final de classification supervisée de documents. L'amélioration de performances sur le classifieur bayésien naïf est manifeste mais il est légitime de se demander si l'apport de l'étape de regroupement serait aussi fort sur d'autres méthodes de classification. En effet, les faiblesses du classifieur bayésien naïf résident précisément dans son hypothèse simplificatrice d'indépendance entre les mots. En ce sens, l'étape de regroupement permet de moduler cette indépendance en rajoutant à moindre coût un lien entre les termes. Pour un autre algorithme qui exploiterait de façon constructive les liens entre les attributs, il n'est pas évident qu'une réduction de dimensionnalité par le biais de la classification non supervisée serait aussi bénéfique.

L'un des grands atouts de la méthode du goulot d'information est de présenter un cadre unificateur aux deux problèmes a priori éloignés que sont la classification non supervisée et la sélection d'attributs (c'est-à-dire, ici, de vocabulaire). En voyant les mots et les documents comme deux dimensions d'une même matrice de comptes, et en cherchant à réduire chaque dimension en minimisant la perte d'information mutuelle par rapport à l'autre, le goulot d'information propose une solution combinée et originale à deux problèmes classiques d'apprentissage statistique. Néanmoins, outre le problème d'estimation des probabilités relatives à W et D par échantillonnage, les méthodes d'inférence présentées restent peu satisfaisantes car très sensibles aux optimums locaux. Ce type de problème n'est pas spécifique à la méthode du goulot d'information et nous en ferons une analyse détaillée pour le cas de l'algorithme EM dans le chapitre 4. Les similarités fortes que nous développerons en section 2.4.1 entre l'EM pour le mélange de multinomiales et l'algorithme du goulot d'information itératif laissent penser qu'ils sont probablement sujets à des difficultés de même nature.

2.2.7 Autres modèles de classification non supervisée

2.2.7.1 Cartes auto-organisatrices

Les cartes auto-organisatrices (*Self-Organising Maps*) de Kohonen [Kohonen, 1997] peuvent être utilisées pour l'analyse exploratoire [Lagus et al., 1999]. Il s'agit de produire en dimension 2 (ou 3) un maillage représentatif des données. Cette grille contient un nombre d'unités restreint, typiquement de l'ordre de quelques centaines, et, dans tous les cas, très inférieur à la dimensionnalité de l'espace de départ. Une carte auto-organisatrice est donc un réseau de neurones particulier dans lequel chaque élément de la grille est associé

à l'espace de départ par un vecteur de poids. Ces poids sont modifiés itérativement en présentant les exemples d'apprentissage les uns après les autres : les vecteurs de poids des cellules sont ainsi ajustés pour être « rapprochés » des exemples d'apprentissage. Les modifications appliquées sont progressivement diminuées pour converger vers une topologie particulière.

Le résultat obtenu est facile à visualiser puisqu'il correspond à une carte. Pour l'application qui nous intéresse, chaque texte est placé en un point de l'espace et l'on peut identifier rapidement des groupes thématiques. Les critiques essentielles à l'encontre de cette méthode sont qu'elle est très exigeante en temps de calcul[14] et que le domaine de l'analyse exploratoire, intrinsèquement multi-dimensionnel et lacunaire, ne se prête pas aussi bien que d'autres à une représentation graphique en dimension 2.

2.2.7.2 Classification en hypergraphes

L'article [Han et al., 1998] se place dans le contexte des hypergraphes pour proposer un partitionnement de l'ensemble des documents. Un hypergraphe est l'extension d'un graphe dans laquelle une arête peut relier plus de deux sommets. Il s'agit donc plus généralement de la donnée d'un ensemble de points et d'un ensemble de groupements de ces sommets. La classification recherchée est un hypergraphe particulier où chaque point apparaît dans un groupement et un seul.

Le regroupement d'hypergraphe consiste dans un premier temps à faire émerger des associations au sein des données pour définir les hyper-arêtes. Dans le cas du texte, il pourrait s'agir de déterminer les ensembles de termes le plus souvent en cooccurrences au sein du corpus. Une fois obtenu ce premier hypergraphe, faire de la classification consiste simplement à réduire le nombre d'hyper-arêtes. [Han et al., 1998] utilise à cet effet un algorithme de coupe d'hypergraphes, HMETIS, fondé sur une approche de type « diviser-pour-régner ». Cependant les premières applications de cette méthode ne sont pas textuelles et il n'est pas évident que les heuristiques utilisées donneraient des résultats satisfaisants dans l'espace de très grande dimension que constitue le vocabulaire.

2.2.7.3 Reconstruction localement linéaire

La reconstruction localement linéaire (*Locally Linear Embedding, LLE*) [Saul and Roweis, 2001] repose sur une décomposition en valeurs propres comme un certain nombre de méthodes d'analyse de données. L'idée spécifique ici est de préserver au mieux les distances locales entre un point et ses voisins. Ce principe de reconstruction locale est en cela assez proche d'une autre technique classique d'analyse de données : l'échelonnement multidimensionnel (*multidimensional scaling, MDS*) [Kruskal, 1964], qui vise à préserver les distances entre couples d'observations.

Dans LLE, un plus grand nombre de voisins est considéré, grâce à deux étapes d'analyse : la première vise à construire une matrice carrée W de dimensions le nombre d'observations et de diagonale nulle, approximant chaque individu par une somme pondérée d'un nombre limité d'autres individus ; une fois obtenu cette matrice W, il s'agit de résoudre un autre problème d'optimisation : reconstruire les données dans un espace de dimension nettement inférieure, en respectant au mieux les liens entre les plus proches voisins codés

[14] En outre, les méthodes proposées dans l'article pour accélérer chaque itération sont discutables, puisqu'elles sont équivalentes à réduire l'ensemble de départ quasi-aléatoirement.

dans W (c'est cette deuxième étape qui fait intervenir un problème de déterimination de vecteurs propres).

L'efficacité de LLE repose grandement sur la confiance dont on dispose envers l'estimation des plus proches voisins, et, donc, envers la distance sur l'espace de départ utilisée à cet effet. Nous avons déjà insisté dans les sections précédentes sur le fait qu'il était difficile dans notre application de déterminer une mesure de similarité satisfaisante entre deux vecteurs de document C_d et $C_{d'}$, le choix d'une représentation nécessitant en général d'appliquer des heuristiques arbitraires.

2.2.7.4 Regroupement discriminant

L'intuition du *regroupement discriminant* (*discriminative clustering*) [Peltonen et al., 2002] est que la tâche de classification non supervisée est subjective dans une large mesure et qu'il est parfois possible de l'orienter au moyen de données externes donnant des informations sur la similarité entre les données. Dans l'exemple traité, la variable auxiliaire est une liste de mots-clés associée à chacun des documents à classer. L'intérêt de la méthode est qu'il est possible d'étendre le regroupement à de nouveaux textes, même si ceux-ci ne sont pas accompagnés de mots-clés. Pour le reste, le regroupement discriminant consiste à minimiser une fonction de distortion entre la distribution conditionnelle des mots-clés connaissant les documents et des profils associés à chaque thème. En cela, la méthode présente des ressemblances avec le goulot d'information (section 2.2.6) notamment dans la résolution par un algorithme itératif d'un problème de minimisation d'une moyenne de divergences de Kullback-Leibler.

Si l'algorithme du regroupement discriminant s'appuie sur une idée originale dans un contexte entre classification supervisée et non supervisée, il s'éloigne un peu de la problématique étudiée dans cette thèse qui vise à proposer une classification sans aucune information extérieure aux documents eux-mêmes. Néanmoins, sur des cas d'application réels, il est courant d'avoir au moins quelques indications sur la classification à obtenir et de ne pouvoir les représenter concrètement sous la forme de classes de documents. Dans ce cas, des modèles plus souples, comme celui du regroupement discriminant, peuvent constituer un compromis intéressant.

2.2.7.5 Analyse en composantes indépendantes

L'*analyse en composantes indépendantes* [Hyvarinen et al., 2001] est également appelée *séparation de sources*, d'après son domaine d'application classique : il s'agit dans ce cas de décomposer les voix mélangées de plusieurs locuteurs au sein de signaux provenant d'un certain nombre de microphones placés dans une pièce. Alors que dans l'analyse en composantes principales l'accent est mis sur l'orthogonalité des composantes et l'explication de la variance au sein des données, l'analyse en composantes indépendantes choisit comme critère privilégié l'indépendance des sources.

L'application aux données textuelles, que décrit [Kolenda and Hansen, 1999], consiste formellement à décomposer la matrice de comptes en trois matrices : $C = AS + U$, où A de dimensions $n_W \times n_T$ est appelée matrice de mélange, S de dimensions $n_T \times n_D$ est la matrices des sources (en nombre n_T) et U est une matrice de bruit, avec une distribution à préciser. L'application d'un algorithme particulier dépend ensuite des choix particuliers des distributions sous-jacentes. Celles qui sont introduites dans [Kolenda and Hansen, 1999] sont dérivées d'autres domaines et ne semblent pas être particulièrement pertinentes pour la décomposition de la matrice de comptes. Dans la section suivante, nous nous intéressons

plus amplement à ce choix de modèles probabilistes pour les données textuelles. D'autre part, la section 2.4 montrera différentes équivalences entre les formulations du problème en termes de décomposition matricielle et celles qui s'appuient sur des modèles génératifs. La décomposition de l'analyse en composantes indépendantes présentée ci-dessus, qui s'avère être la même que celle de NMF, sera naturellement abordée.

2.3 Modèles probabilistes

Nous abordons maintenant la fouille de textes dans le cadre des statistiques paramétriques, en présentant des modèles de mélange qui diffèrent sur le choix et l'organisation de la structure sémantique au moyen de variables cachées.

En guise d'introduction, nous présentons le modèle unigramme, probablement la modèlisation la plus simple pour des données textuelles. Nous abordons ensuite les modèles de classification thématique proprement dits, en commençant par le modèle de mélange de multinomiales qui n'autorise qu'une variable latente de thème par document. L'analyse sémantique latente probabiliste (PLSA) propose plus de flexibilité en modélisant les paires (mot,document) mais ne parvient pas à décrire un modèle complètement génératif. Les modèles d'allocation Dirichlet latente (LDA) et Gamma-Poisson (GaP) corrigent ce défaut par l'introduction de modèles théoriquement mieux justifiés mais peut-être moins intuitifs.

Dans les dernières sections, nous décrivons des problématiques différentes, concernant des extensions hiérarchiques ou des modélisations différentes des comptes.

2.3.1 Modèle unigramme

Nous considérons à présent le corpus comme un ensemble de n_D observations multivariées sur l'espace du vocabulaire et nous cherchons une loi adéquate pour modéliser ces vecteurs de comptes C_d. Deux choix courants sous l'hypothèse du « sac de mots » sont :

- la loi multinomiale de paramètres $(l_d, (\beta_1, \ldots, \beta_{n_W}))$:

$$P(C_d; l_d, (\beta_1, \ldots, \beta_{n_W})) = \frac{l_d!}{\prod_{w=1}^{n_W} C_{wd}!} \prod_{w=1}^{n_W} \beta_w^{C_{wd}} \tag{2.4}$$

  avec $\forall w \in \{1, \ldots, n_W\}, \beta_w \in [0,1]$ et $\sum_{w=1}^{n_W} \beta_w = 1$;
- n_W lois de Poisson indépendantes de paramètres $(\lambda_1, \ldots, \lambda_{n_W})$:

$$P(C_d; \lambda_1, \ldots, \lambda_{n_W}) = \prod_{w=1}^{n_W} \frac{e^{-\lambda_w} \lambda_w^{C_{wd}}}{C_{wd}!} \tag{2.5}$$

Malgré leur différence apparente, ces représentations sont étroitement liées asymptotiquement, si nous examinons le phénomène mot par mot. En effet, dans l'hypothèse multinomiale, le compte d'un mot particulier de probabilité β_w suit une loi binomiale. Or, sous les hypothèses (raisonnables ici) que le nombre de tirages l tende vers l'infini, que β_w tende vers 0 et que le produit $l\beta_w$ tende vers une constante λ_w, la loi binomiale tend vers une loi de Poisson : pour $k \ll l$,

$$\mathrm{P}_{\text{binomiale}}(k) \quad = \quad \binom{l}{k} \beta_w^k (1-\beta_w)^{l-k}$$

$$
\begin{aligned}
&= \frac{l(l-1)\dots(l-k+1)}{k!}\frac{l^k}{l^k}\beta_w^k\left(\frac{1}{1-\beta_w}\right)^k e^{l\log(1-\beta_w)} \\
&= \frac{l}{l}\frac{l-1}{l}\cdots\frac{l-k+1}{l}\left(\frac{1}{1-\beta_w}\right)^k\frac{(l\beta_w)^k}{k!}e^{l\log(1-\beta_w)} \\
&\sim 11\dots11\frac{\lambda_w^k}{k!}e^{-l\beta_w} \\
&\sim \frac{\lambda_w^k}{k!}e^{-\lambda_w} \\
&\sim \mathrm{P}_{\mathrm{Poisson}}(k)
\end{aligned}
$$

Vue la similarité asymptotique de ces deux modélisations, nous pouvons supposer que ce qui est vrai pour l'une l'est également pour l'autre et choisir la représentation sur d'autres critères que l'adaptation aux données. En l'occurrence, nous allons privilégier la représentation multinomiale, pour son traitement plus convaincant de la longueur des documents (voir le détail de cette critique à l'encontre de la modélisation par loi de Poisson dans la section 2.3.5).

Si nous supposons que le corpus est généré par une loi multinomiale sur le vocabulaire, de paramètres $(l, (\beta_1, \dots, \beta_{n_W}))$ et que nous souhaitons estimer les β par maximum de vraisemblance, nous obtenons la formule suivante :

$$\forall w \in \{1, \dots, n_W\}, \widehat{\beta}_w = \frac{\sum_{d=1}^{n_D} C_{wd}}{l}$$

Ce modèle de tirage indépendant et identiquement distribué des mots selon la même distribution multinomiale est connu sous le nom de *modèle unigramme*. Avec cette méthode, le nombre d'observations qui contribuent à l'estimation d'un coefficient β_w est le nombre d'apparitions du mot d'indice w dans le corpus.

Or le vocabulaire a une distribution très particulière (nous y reviendrons en section 2.3.7.1) qui fait que les paramètres liés aux premiers mots sont très bien estimés (les formes les plus fréquentes apparaissent plusieurs dizaines de milliers de fois dans un corpus de 5000 documents) alors que ceux qui sont associés aux hapax sont tous égaux et considérablement mal estimés à partir d'une unique observation pour chaque terme. Le recours à des techniques de *lissage*, indispensable en pratique, permet d'atténuer en partie ce problème. La technique la plus simple, dite lissage de Laplace, consiste à ajouter une constante $\lambda_\beta > 0$ à chaque compte, l'estimateur devenant alors :

$$\forall w \in \{1, \dots, n_W\}, \widehat{\beta}_w = \frac{\sum_{d=1}^{n_D} C_{wd} + \lambda_\beta}{l + n_W\lambda_\beta}$$

Il existe de nombreuses autres techniques de lissage en modélisation du langage [Gale, 1994, Chen and Goodman, 1996], mais nous ne les avons pas considérées dans le cadre de cette thèse.

2.3.2 Mélange de multinomiales

Il s'agit d'un des modèles probabilistes les plus simples pour générer un corpus multithématique. Il a été présenté dans [Nigam et al., 2000] dans un contexte un peu différent, dans le but d'améliorer les performances d'un classifieur bayésien par l'ajout massif de données non-étiquetées. [Clérot et al., 2004] présente un modèle proche (mélange de Poisson),

dans un contexte lié plus directement à la situation qui nous intéresse. Ici, nous considérons, a priori, que nous ne disposons d'aucune information de classes et nous estimons les paramètres du modèle génératif pour grouper ensuite les documents suivant les différentes composantes du mélange.

Nous décrivons le modèle dans sa forme la plus simple et détaillons l'application de l'algorithme *Espérance-Maximisation (EM)* [Dempster et al., 1977]. Nous reviendrons sur ce modèle tout au long de la thèse, et en particulier dans le chapitre 4, dans une plus grande généralité et dans un cadre bayésien.

On suppose que les textes sont générés indépendamment. Chaque document (numéroté $d \in \{1, \ldots, n_D\}$) résulte de l_d tirages indépendants sur le vocabulaire (hypothèse du « sac de mots ») suivant une distribution liée au thème, lequel est une variable cachée tirée une fois par texte. D'où le modèle génératif pour un document :

1. Tirer un thème $T_d \sim \text{Mult}(1, (\alpha_1, \ldots, \alpha_{n_T}))$ où les α_t sont des paramètres tels que $\sum_{t=1}^{n_T} \alpha_t = 1$.
2. Conditionnellement au thème T_d, tirer un vecteur de l_d mots

$$C_d \sim \text{Mult}(l_d, (\beta_{T_d 1}, \ldots, \beta_{T_d n_W}))\,,$$

β étant une matrice $n_T \times n_W$ de paramètres telle que

$$\forall t \in \{1, \ldots, n_T\}, \sum_{w=1}^{n_W} \beta_{tw} = 1.$$

Les paramètres du modèle[15] sont donc :

$$\Theta = ((\alpha_t)_{t=1,\ldots,n_T}, (\beta_{tw})_{t=1,\ldots,n_T, w=1,\ldots,n_W})$$

La probabilité d'un document est alors :

$$\begin{aligned}
\text{P}(C_d; \Theta) &= \sum_{t=1}^{n_T} \text{P}(T_d = t; \Theta)\, \text{P}(C_{1d}, \ldots, C_{n_W d} | T_d = t; \Theta) \\
&= \sum_{t=1}^{n_T} \text{P}(T_d = t; \Theta) l_d! \prod_{w=1}^{n_W} \frac{\text{P}(w | T_d = t; \Theta)^{C_{wd}}}{C_{wd}!} \\
&= \frac{l_d!}{\prod_{w=1}^{n_W} C_{wd}!} \sum_{t=1}^{n_T} \alpha_t \prod_{w=1}^{n_W} \beta_{tw}^{C_{wd}}
\end{aligned}$$

Ainsi chaque thème contribue à la probabilité globale du document par sa probabilité a priori α_t et, pour chaque occurrence du texte, par la probabilité β_{tw} d'émission du mot w dans le thème en question.

La probabilité du corpus, ou vraisemblance des observations, est obtenue en réalisant le produit de l'expression ci-dessus pour l'ensemble des documents étudiés, par hypothèse d'indépendance. Cependant, il n'est pas possible d'établir directement une expression d'un estimateur de maximum de vraisemblance.

Il est alors d'usage de faire appel à l'algorithme EM. Il s'agit d'un algorithme itératif qui consiste à estimer en alternance les probabilités a posteriori des documents d'appartenir aux thèmes et les paramères α, β du modèle. Pour cela, on s'intéresse à l'espérance

[15] Dans ce modèle, comme dans tous ceux que nous étudions, les longueurs l_d des documents sont considérées comme des variables exogènes et ne seront donc pas modélisées statistiquement.

conditionnellement aux observations de la log-vraisemblance *complète* $\mathcal{L}^c$, c'est-à-dire la log-vraisemblance des couples vecteur de comptes C_d et thème T_d, définie par :

$$\begin{aligned}
\mathcal{L}^c &= \sum_{d=1}^{n_D} \log \mathrm{P}(C_d, T_d) \\
&= \sum_{d=1}^{n_D} \log \mathrm{P}(T_d) + \log \mathrm{P}(C_d|T_d) \\
&= \sum_{d=1}^{n_D} \left(\log \alpha_{T_d} + \sum_{w=1}^{n_W} \log \beta_{T_d w}^{C_{wd}} \right) + K \\
&= \sum_{d=1}^{n_D} \sum_{t=1}^{n_T} \mathbb{1}_{\{T_d=t\}} \left(\log \alpha_t + \sum_{w=1}^{n_W} C_{wd} \log \beta_{tw} \right) + K
\end{aligned}$$

où K est une constante indépendante des paramètres (que nous oublierons par la suite). La notation $\mathbb{1}$ désigne la fonction indicatrice définie par :

$$\mathbb{1}_A = \begin{cases} 1 & \text{si } A \text{ est vrai ;} \\ 0 & \text{sinon.} \end{cases}$$

Par définition, son espérance est la probabilité de A. En calculant l'espérance conditionnellement aux observations et en supposant que les paramètres sont fixés à des valeurs Θ' issues de l'itération précédente, on obtient donc :

$$E[\mathcal{L}^c|C;\Theta'] = \sum_{d=1}^{n_D} \sum_{t=1}^{n_T} \mathrm{P}(T_d = t|C_d;\Theta') \left(\log \alpha_t + \sum_{w=1}^{n_W} C_{wd} \log \beta_{tw} \right)$$

Les probabilités a posteriori sont données par la formule de Bayes, conduisant, pour $t \in \{1,\ldots,n_T\}, d \in \{1,\ldots,n_D\}$ à :

$$\begin{aligned}
\mathrm{P}(T_d = t|C_d;\Theta') &= \frac{\mathrm{P}(C_d|T_d = t;\Theta')\,\mathrm{P}(T_d = t;\Theta')}{\mathrm{P}(C_d;\Theta')} \\
&= \frac{\mathrm{P}(C_d|T_d = t;\Theta')\,\mathrm{P}(T_d = t;\Theta')}{\sum_{t'=1}^{n_T} \mathrm{P}(C_d|T_d = t';\Theta')\,\mathrm{P}(T_d = t';\Theta')} \\
&= \frac{\alpha'_t \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{tw}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{t'w}}
\end{aligned} \tag{2.6}$$

Nous avons déjà vu l'expression du numérateur dans le calcul : il s'agit de la probabilité jointe du document C_d et du thème t, qui se décompose intuitivement en produit de la probabilité a priori du thème t et de l'« importance » dans le thème de chaque occurrence du mot w dans le texte considéré. Le dénominateur vient de l'opération de normalisation correspondant à $\sum_{t=1}^{n_T} P(T_d = t|C_d;\alpha',\beta')$.

Il est alors possible d'établir les équations de ré-estimation des paramètres en maximisant $E[\mathcal{L}^c|C;\Theta']$, avec la technique des multiplicateurs de Lagrange pour normaliser de façon appropriée les paramètres α (le vecteur somme à 1) et β (chaque colonne somme à

1). On obtient, pour $t \in \{1, \ldots, n_T\}$ et $w \in \{1, \ldots, n_W\}$:

$$\begin{aligned} \alpha_t &= \frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(T_d = t | C_d; \Theta') && (2.7) \\ &= \frac{1}{n_D} \sum_{d=1}^{n_D} \frac{\alpha'_t \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{tw}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{t'w}} \\ \beta_{tw} &= \frac{\sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T_d = t | C_d; \Theta')}{\sum_{d=1}^{n_D} l_d \mathrm{P}(T_d = t | C_d; \Theta')} && (2.8) \\ &= \frac{\sum_{d=1}^{n_D} C_{wd} \frac{\alpha'_t \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{tw}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{t'w}}}{\sum_{d=1}^{n_D} l_d \frac{\alpha'_t \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{tw}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{t'w}}} \end{aligned}$$

Ces formules ont elles aussi une interprétation intuitive simple si les probabilités d'appartenance $\mathrm{P}(T_d = t | C_d; \Theta')$ sont exactement 0 ou 1 (classification « déterministe », chaque texte « appartient » alors à un thème et un seul) :

- Nous obtenons α_t en comptant le nombre de documents dans le thème t puis en normalisant.
- Nous déterminons la nouvelle valeur de β_{tw} en dénombrant le nombre d'occurrences du mot w dans les textes correspondant au thème t, puis nous normalisons sur l'ensemble des mots.

Cette interprétation peut être étendue au cas où les probabilités d'appartenance ne sont pas binaires (classification « probabiliste »). Chaque texte contribue alors au renouvellement des paramètres en proportion de son « implication » dans le thème.

L'algorithme EM consiste à appliquer, à partir d'une valeur initiale des paramètres, les formules (2.6), (2.7) et (2.8) de façon itérative jusqu'à la vérification d'un critère de convergence.

Lorsqu'un mot w n'est jamais observé dans un thème t, ces formules conduisent à une estimation nulle pour β_{tw}. De façon similaire au traitement du modèle unigramme, il est alors nécessaire de recourir à des techniques de lissage des estimateurs, pour rendre compte du fait que, même si un mot n'a jamais été vu en conjonction avec un thème donné dans l'ensemble d'apprentissage, son apparition dans ce thème n'est pas totalement impossible (elle est néanmoins de probabilité très faible). Il est d'usage de traiter ce problème par un lissage de Laplace, consistant à augmenter tous les comptes d'une constante (souvent égale à 1). Nous verrons dans le chapitre 4 que ce lissage peut être interprété comme correspondant à l'algorithme EM associé au maximum *a posteriori* (et non plus au maximum de vraisemblance) lorsque les paramètres β sont munis d'une distribution *a priori* de type Dirichlet.

Au final, nous obtenons pour chaque document des probabilités d'appartenance à chaque thème. Même si le modèle est dit « monothématique », il est important de souligner que, dans le cas général, le partitionnement induit n'est pas « déterministe » mais « probabiliste » : chaque texte a une probabilité d'appartenance plus ou moins marquée à chaque thème.

Nous concluons cette présentation en soulignant un des avantages du modèle : il s'adapte de façon très naturelle aux cas supervisé et semi-supervisé [Nigam et al., 2000]. Les variables latentes thématiques de certains documents ne sont alors plus cachées mais observables et ne sont donc pas remises à jour, l'équation (2.6) n'est appliquée que pour les textes dont

l'étiquette est inconnue. Un exemple d'application au contexte semi-supervisé sera étudié en section 4.6.1.

2.3.3 Analyse sémantique latente probabiliste (PLSA)

Nous présentons maintenant le modèle de l'analyse sémantique latente probabiliste (PLSA) [Hofmann, 2001]. Cependant, nous n'adoptons pas tout à fait la formalisation de l'article car elle ne nous semble pas la plus claire pour distinguer les quantités fixes des paramètres et, par conséquent, les étapes de l'algorithme EM. La notation présentée ici est évoquée brièvement par Hofmann comme une paramétrisation équivalente. Nous reviendrons ensuite plus en détail sur les formulations employées dans l'article et le lien avec nos notations.

Dans le modèle PLSA, chacune des l occurrences est une observation indépendante des autres. Le corpus est donc vu comme un ensemble de couples mot/document (W, D), chaque paire apparaissant C_{wd} fois. On suppose maintenant, pour chaque couple, l'existence d'une variable cachée liée au *thème* et à valeurs dans $\{1, \ldots, n_T\}$. Il s'agit ensuite d'un modèle de mélange classique en supposant que les tirages du mot et du document sont indépendants, conditionnellement au thème. Le modèle génératif consiste à tirer l observations de la façon suivante :

1. Tirer un thème $T \sim \text{Mult}(1, (\alpha_1, \ldots, \alpha_{n_T}))$ où les α_t sont des paramètres tels que $\sum_{t=1}^{n_T} \alpha_t = 1$.
2. Conditionnellement à T, tirer un document $D \sim \text{Mult}(1, (\mu_{T1}, \ldots, \mu_{Tn_D}))$, μ étant une matrice $n_T \times n_D$ de paramètres telle que $\forall t \in \{1, \ldots, n_T\}, \sum_{d=1}^{n_D} \mu_{td} = 1$.
3. Conditionnellement à T, tirer un mot $W \sim \text{Mult}(1, (\beta_{T1}, \ldots, \beta_{Tn_W}))$, β étant une matrice $n_T \times n_W$ de paramètres telle que $\forall t \in \{1, \ldots, n_T\}, \sum_{w=1}^{n_W} \beta_{tw} = 1$.

Les paramètres du modèle sont donc :

$$\Theta = ((\alpha_t)_{t=1,\ldots,n_T}, (\mu_{td})_{t=1,\ldots,n_T,d=1,\ldots,n_D}, (\beta_{tw})_{t=1,\ldots,n_T,w=1,\ldots,n_W})$$

Si l'on note (W_i, D_i) les variables aléatoires associées aux numéros du mot et du document du tirage $i = 1, \ldots, l$ et T_i la variable cachée correspondante, la vraisemblance des données $((W_1, D_1), \ldots, (W_l, D_l))$ est alors :

$$\begin{aligned}
\mathcal{V} &= \mathrm{P}((W_i, D_i)_{i=1,\ldots,l}; \Theta) \\
&= \prod_{i=1}^{l} \mathrm{P}((W_i, D_i); \Theta) \\
&= \prod_{i=1}^{l} \sum_{t=1}^{n_T} \mathrm{P}(T_i = t; \Theta)\, \mathrm{P}((W_i, D_i) | T_i = t; \Theta) \\
&= \prod_{i=1}^{l} \sum_{t=1}^{n_T} \mathrm{P}(T_i = t; \Theta)\, \mathrm{P}(W_i | T_i = t; \Theta)\, \mathrm{P}(D_i | T_i = t; \Theta) \\
&= \prod_{i=1}^{l} \sum_{t=1}^{n_T} \alpha_t \beta_{tW_i} \mu_{tD_i}
\end{aligned}$$

La matrice de comptes permet de déterminer le nombre d'occurrences de chaque couple,

ce qui donne par suite une expression de la log-vraisemblance :

$$\mathcal{L} = \sum_{w=1}^{n_W} \sum_{d=1}^{n_D} C_{wd} \log \left(\sum_{t=1}^{n_T} \alpha_t \beta_{tw} \mu_{td} \right)$$

Mais, cette expression n'étant pas maximisable directement en fonction des paramètres, nous considérons l'espérance conditionnellement aux observations de la log-vraisemblance *complète* $\mathcal{L}^c$, c'est-à-dire la log-vraisemblance des triplets $((W_1, D_1, T_1), \ldots, (W_l, D_l, T_l))$, en supposant que le thème correspondant à l'observation (W_i, D_i) est T_i.

$$\begin{aligned}
\mathcal{L}^c &= \sum_{i=1}^{l} \log \mathrm{P}((W_i, D_i, T_i); \Theta) \\
&= \sum_{i=1}^{l} \log \mathrm{P}(T_i; \Theta) + \log \mathrm{P}(W_i | T_i; \Theta) + \log \mathrm{P}(D_i | T_i; \Theta) \\
&= \sum_{i=1}^{l} \log \alpha_{T_i} + \log \beta_{T_i W_i} + \log \mu_{T_i D_i} \\
&= \sum_{i=1}^{l} \sum_{t=1}^{n_T} \mathbb{1}_{\{T_i = t\}} \left(\log \alpha_t + \log \beta_{t W_i} + \log \mu_{t D_i} \right)
\end{aligned}$$

L'espérance conditionnelle par rapport aux paramètres Θ' issus de l'itération précédente se calcule de façon similaire au modèle simple de mélange de multinomiales :

$$\begin{aligned}
E[\mathcal{L}^c | C; \Theta'] &= \sum_{t=1}^{n_T} \sum_{i=1}^{l} \sum_{w=1}^{n_W} \sum_{d=1}^{n_D} \mathbb{1}_{\{W_i = w, D_i = d\}} \mathrm{P}(T_i = t | W_i = w, D_i = d; \Theta') \\
&\qquad \times (\log \alpha_t + \log \beta_{tw} + \log \mu_{td}) \\
&= \sum_{t=1}^{n_T} \sum_{w=1}^{n_W} \sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta') \\
&\qquad \times (\log \alpha_t + \log \beta_{tw} + \log \mu_{td})
\end{aligned}$$

La maximisation de cette expression donne, en ajoutant les conditions de normalisation adéquates avec la technique des multiplicateurs de Lagrange, pour $t = 1, \ldots, n_T, w = 1, \ldots, n_W, d = 1, \ldots, n_D$:

$$\alpha_t = \frac{1}{l} \sum_{w=1}^{n_W} \sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta') \tag{2.9}$$

$$\begin{aligned}
\beta_{tw} &= \frac{\sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta')}{\sum_{w=1}^{n_W} \sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta')} \\
&= \frac{\sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta')}{l \alpha_t}
\end{aligned} \tag{2.10}$$

$$\begin{aligned}
\mu_{td} &= \frac{\sum_{w=1}^{n_W} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta')}{\sum_{w=1}^{n_W} \sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta')} \\
&= \frac{\sum_{w=1}^{n_W} C_{wd} \mathrm{P}(T = t | W = w, D = d; \Theta')}{l \alpha_t}
\end{aligned} \tag{2.11}$$

Pour calculer les probabilités a posteriori, il faut utiliser la formule de Bayes, ainsi, pour $t = 1, \ldots, n_T, w = 1, \ldots, n_W, d = 1, \ldots, n_D$:

$$\begin{aligned}
\mathrm{P}(T = t | W = w, D = d; \Theta') &= \frac{\mathrm{P}(W = w, D = d | T = t; \Theta') \, \mathrm{P}(T = t; \Theta')}{\mathrm{P}(W = w, D = d; \Theta')} \\
&= \frac{\mathrm{P}(W = w, D = d | T = t; \Theta') \, \mathrm{P}(T = t; \Theta')}{\sum_{t'=1}^{n_T} \mathrm{P}(W = w, D = d | T = t'; \Theta') \, \mathrm{P}(T = t'; \Theta')} \\
&= \frac{\alpha'_t \beta'_{tw} \mu'_{td}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \beta'_{t'w} \mu'_{t'd}}
\end{aligned}$$

Sous cette forme, le modèle est symétrique dans les rôles joués par les mots et les documents, ce qui semble relativement contre-intuitif. C'est pourquoi Hofmann présente le modèle génératif d'une autre façon lorsqu'il l'introduit dans [Hofmann, 2001], en inversant les tirages du thème et du document :

1. Tirer un document $D \sim \mathrm{Mult}(1, \mathrm{P}(D))$.
2. Conditionnellement au document D, tirer un thème $T \sim \mathrm{Mult}(1, \mathrm{P}(T|D))$.
3. Conditionnellement au thème T, tirer un mot $W \sim \mathrm{Mult}(1, \mathrm{P}(W|T))$.

Alors que le nombre de paramètres semble différent, il suffit d'appliquer la formule de Bayes pour vérifier que la probabilité jointe d'un document et d'un terme particulier obéit à la même équation, ainsi que, par conséquent, la log-vraisemblance et les formules de ré-estimation. Il est intéressant de remarquer que la probabilité de tirer un document n'est en fait pas ré-estimée car la valeur maximisant la vraisemblance est le nombre de mots dans le document divisé par le nombre de mots dans le corpus, qui est identique d'une itération sur l'autre. Il n'est pas difficile de prouver que cette probabilité est égale à la même constante avec l'autre paramétrisation, en décomposant les probabilités a posteriori en fonction des probabilités à l'étape précédente, à l'aide des formules de ré-estimation :

$$\begin{aligned}
\forall d \in \{1, \ldots, n_D\}, \quad \mathrm{P}(D = d; \Theta) &= \sum_{t=1}^{n_T} \mathrm{P}(D = d | T = t; \Theta) \, \mathrm{P}(T = t; \Theta) \\
&= \sum_{t=1}^{n_T} \mu_{td} \alpha_t \\
&= \sum_{t=1}^{n_T} \frac{\sum_{w=1}^{n_W} C_{wd} \, \mathrm{P}(T = t | W = w, D = d; \Theta')}{l} \\
&= \frac{1}{l} \sum_{w=1}^{n_W} C_{wd} \\
&= \frac{l_d}{l}
\end{aligned}$$

Et une formule équivalente est calculable pour les mots, ce qui est cohérent avec la nature d'un estimateur du maximum de vraisemblance.

Les autres formules de ré-estimation sont exactement identiques pour β et équivalentes pour p et α par la formule de Bayes. La présentation de l'article permet, en pratique, de ré-estimer moins de paramètres mais, étant donné que les formules sont plus compliquées, la complexité est sensiblement la même.

Le lien entre LSA et PLSA est ensuite explicité, la formule clé étant la décomposition convexe de la probabilité d'un mot pour un document précis donné dans « l'espace latent »

des thèmes, c'est-à-dire sur les probabilités conditionnellement à un thème donné[16] pour $w = 1, \ldots, n_W, d = 1, \ldots, n_D$:

$$\begin{aligned}
\mathrm{P}(W = w|D = d;\Theta) &= \sum_{t=1}^{n_T} \mathrm{P}(W = w|T = t;\Theta)\,\mathrm{P}(T = t|D = d;\Theta) \\
&= \sum_{t=1}^{n_T} \mathrm{P}(W = w|T = t;\Theta)\,\mathrm{P}(T = t;\Theta)\frac{\mathrm{P}(D = d|T = t;\Theta)}{\mathrm{P}(D = d;\Theta)} \\
&= \sum_{t=1}^{n_T} \beta_{tw}\alpha_t \frac{\mu_{td}}{\frac{l_d}{l}}
\end{aligned}$$

Soit :

$$l_d\,\mathrm{P}(W = w|D = d;\Theta) = \sum_{t=1}^{n_T} \beta_{tw}(l\alpha_t)\mu_{td}$$

Sous cette forme, nous reconnaissons en effet une analogie avec la décomposition en valeurs singulières de la matrice de comptes, que l'on cherche à approximer par $l_d\,\mathrm{P}(W = w|D = d;\Theta)$ avec U correspondant à β, V à p et $l\alpha_t$ aux n_T valeurs singulières. Dans les deux cas, les valeurs singulières et proportions du mélange représentent l'« importance » de chaque thème dans le corpus alors que les matrices de passage font le lien entre mots/documents et thèmes.

Toutefois, la différence fondamentale réside dans la quantité optimisée lors du processus d'approximation : pour LSA, il s'agit de minimiser la distance euclidienne alors que PLSA cherche le maximum de vraisemblance[17]. Hofmann énumère également plusieurs avantages de PLSA sur LSA : l'interprétation plus facile des éléments du mélange et la possibilité de trouver le nombre de thèmes optimal grâce à la théorie de sélection de modèles et de contrôle de complexité. Sur le plan pratique, en revanche, il semble que LSA soit plus facile à mettre en œuvre que PLSA. En effet, la décomposition en valeurs singulières est effectuée de manière exacte et permet de trouver un optimum global alors que l'algorithme EM ne converge qu'itérativement et vers un maximum dont rien ne garantit qu'il soit global. Hofmann nuance sérieusement ces inconvénients en signalant que, dans les expériences conduites, les extrema globaux sont souvent « sur-adaptés » aux données d'apprentissage (on parle de *sur-apprentissage* ou *overfitting*) et qu'il est donc bon de ne pas conduire trop d'itérations. D'autre part, la complexité d'une itération de l'algorithme est du même ordre de grandeur ($O(ln_T)$) que LSA[18] et donc les temps de calcul globaux sont comparables si le nombre d'itérations est réduit.

Hofmann compare ensuite PLSA avec des modèles de classification non supervisée qui associent une variable latente thématique à chaque document. Il met l'accent sur la différence notable que, lorsque la formule de Bayes est utilisée pour calculer les probabilités d'appartenance a posteriori d'un document à un groupe, l'appartenance d'un texte à plusieurs thèmes ne réside que dans l'incertitude sur la valeur du thème. Dans le modèle PLSA, en revanche, une variable latente (un thème) différente est associée à chaque observation (mot, document), deux occurrences d'un même mot dans un même document pouvant

[16] Hofmann précise qu'il s'agit d'une projection au sens de la divergence de Kullback-Leibler.

[17] La façon de normaliser les matrices de passages est également différente et est la conséquence du point de vue adopté (changement de base dans un cas et probabiliste dans l'autre).

[18] Le calcul de la complexité d'une décomposition en valeurs singulières avec approximation de rang k sur une matrice creuse est un problème difficile. Voir [Berry et al., 1995] pour une étude détaillée.

même être issues de différents « aspects » ou thèmes. Tous les mots d'un document sont tout de même liés, au sens où ils partagent les probabilités a priori $\mathrm{P}(T = t|D = d; \Theta)$.

L'évaluation des modèles est faite selon la *perplexité* sur des données de test. La définition et les calculs de probabilités correspondants sont détaillés dans le chapitre 3. Les résultats montrent que PLSA est toujours plus fidèle aux données que LSA mais l'obtention des probabilités prédictives est sujette à discussion dans les deux modèles. Il est, par conséquent, difficile de dresser des conclusions, comme le souligne Hofmann. S'intéressant au phénomène de sur-apprentissage, il propose une version modifiée de l'algorithme EM avec un paramètre de température qui contrôle la convergence et que l'on estime sur un autre jeu de données.

L'évaluation est prolongée sur une tâche de recherche d'information sur des corpus annotés. PLSA est comparé en termes de précision/rappel à une technique de base de similarité documents/requêtes et à LSA. Cependant, la similarité utilisée n'est pas exactement celle qui serait prédite par PLSA, mais, apparemment, une moyenne des similarités prédites sur chacun des thèmes cachés. La raison pour laquelle cette technique est utilisée au détriment de l'importance relative des différents thèmes dans le corpus n'est pas très claire dans l'article. Hofmann évoque un problème de robustesse.

2.3.4 Allocation Dirichlet latente (LDA)

L'allocation Dirichlet latente (Latent Dirichlet Allocation, LDA) [Blei et al., 2002] a été élaborée à partir des limites de PLSA. Les deux critiques essentielles de Blei, Ng et Jordan à l'égard de PLSA sont :

- Le modèle n'est pas un modèle génératif des documents, car la variable aléatoire des textes est une multinomiale qui ne peut prendre comme valeurs que les numéros des textes de l'ensemble d'apprentissage. Si une telle stratégie d'indiçage ne semble pas poser de problème sur les mots, elle est plus discutable sur les documents car ne permet pas d'estimer simplement la probabilité d'un document non vu.
- La complexité du modèle est linéaire en fonction du nombre de documents utilisés pour l'estimation des paramètres. Blei, Ng et Jordan voient ce problème comme une cause du phénomène de sur-apprentissage observé par Hofmann (qu'ils confirment d'ailleurs dans leurs expériences).

Le modèle génératif proposé doit donc contenir une variable de valeur commune à tous les mots d'un document mais sans pour autant perdre la pluri-thématicité garantie par PLSA. C'est pourquoi les auteurs suggèrent de tirer chaque mot d'un thème, lui-même tiré dans un mélange pour chaque document $d = 1, \ldots, n_D$:

1. Tirer les paramètres du mélange $\mu_d \sim \mathrm{Dir}(\alpha)$. μ_d est donc un vecteur dont les n_T composantes somment à 1. α est un vecteur de même longueur mais dont les composantes, strictement positives, sont les mêmes pour tous les textes du corpus.
2. Pour chaque mot $i = 1, \ldots, l_d$:
   (a) Conditionnellement à μ_d, tirer un thème $T_i \sim \mathrm{Mult}(1, (\mu_1, \ldots, \mu_{n_T}))$.
   (b) Conditionnellement à T_i, tirer un mot $W_i \sim \mathrm{Mult}(1, (\beta_{T_i 1}, \ldots, \beta_{T_i n_W}))$, β étant une matrice $n_T \times n_W$ de paramètres telle que $\forall t \in \{1, \ldots, n_T\}, \sum_{w=1}^{n_W} \beta_{tw} = 1$.

Les paramètres du modèle sont donc :

$$\Theta = ((\alpha_t)_{t=1,\ldots,n_T}, (\beta_{tw})_{t=1,\ldots,n_T, w=1,\ldots,n_W})$$

Comme le note [Buntine and Jakulin, 2004], on peut également voir ce modèle comme un modèle de mélange de lois discrètes, dont les coefficients de mélange μ_d sont tirés indépendamment pour chaque document. Chaque document résulte alors de l_d tirages indépendants selon une loi discrète de paramètres $(\sum_{t=1}^{n_T} \mu_{dt}\beta_{t1}, \ldots, \sum_{t=1}^{n_T} \mu_{dt}\beta_{tn_W})$, c'est à dire que le profil d'occurrences caractéristique de chaque document s'obtient comme une combinaison barycentrique, contrôlée par la variable latente μ_d, des n_T profils de base $\beta_1, \ldots, \beta_{n_T}$.

Dans tous les cas, la probabilité d'un document est :

$$\begin{aligned}
\mathrm{P}(C_d;\Theta) &= \int_\mu \mathrm{P}(C_d|\mu;\Theta)p(\mu;\Theta)d\mu \\
&= \int_\mu \left(\prod_{i=1}^{l_d} \mathrm{P}(W_i|\mu;\beta)\right) p(\mu;\alpha)d\mu \\
&= \int_\mu \left(\prod_{i=1}^{l_d}\sum_{t=1}^{n_T} \mathrm{P}(W_i|T_i=t;\beta)\,\mathrm{P}(T_i=t|\mu)\right) p(\mu;\alpha)d\mu
\end{aligned}$$

Or, pour μ et α dans le simplexe de dimension $n_T - 1$, pour $w = 1, \ldots, n_W$ et, pour $t = 1, \ldots, n_T$,

$$p(\mu;\alpha) = \frac{\Gamma\left(\sum_{t=1}^{n_T}\alpha_t\right)}{\prod_{t=1}^{n_T}\Gamma(\alpha_t)} \prod_{t=1}^{n_T} \mu_t^{\alpha_t - 1}$$

$$\mathrm{P}(T=t|\mu) = \mu_t$$

$$\mathrm{P}(W=w|T=t;\beta) = \beta_{tw}$$

Donc :

$$\begin{aligned}
\mathrm{P}(C_d;\Theta) &= \frac{\Gamma\left(\sum_{t=1}^{n_T}\alpha_t\right)}{\prod_{t=1}^{n_T}\Gamma(\alpha_t)} \int_\mu \left(\prod_{i=1}^{l_d}\sum_{t=1}^{n_T} \beta_{tW_i}\mu_t\right)\left(\prod_{t=1}^{n_T}\mu_t^{\alpha_t-1}\right)d\mu \\
&= \frac{\Gamma\left(\sum_{t=1}^{n_T}\alpha_t\right)}{\prod_{t=1}^{n_T}\Gamma(\alpha_t)} \int_\mu \left(\prod_{w=1}^{n_W}\left(\sum_{t=1}^{n_T}\beta_{tw}\mu_t\right)^{C_{wd}}\right)\left(\prod_{t=1}^{n_T}\mu_t^{\alpha_t-1}\right)d\mu
\end{aligned}$$

Il s'agit d'une fonction hypergéométrique qui n'est pas maximisable directement. Blei, Ng et Jordan utilisent une technique d'inférence variationnelle pour trouver une bonne approximation[19]. Cette méthode, fondée sur l'inégalité de Jensen, consiste à trouver une distribution calculable, plus simple, et d'ajuster ses paramètres afin de minimiser la divergence de Kullback-Leibler avec la distribution visée. Les auteurs proposent ici de simplifier le modèle en enlevant le lien entre μ et T. Il est alors possible de minimiser la distance entre le modèle approché et la distribution initiale par la méthode de Newton-Raphson. Il s'agit de l'étape E de cet algorithme EM « variationnel ». Dans l'étape M, on peut ensuite maximiser en α et β la borne inférieure de la vraisemblance ainsi obtenue.

LDA est d'abord évalué formellement par une mesure de perplexité (voir le chapitre 3). La méthode est comparée au modèle unigramme simple, à un mélange de multinomiales

[19]Il faut noter que d'autres techniques d'inférence ont été proposées depuis pour estimer les paramètres de LDA. L'échantillonnage de Gibbs notamment semble donner de meilleurs résultats que l'inférence variationnelle, comme le montrent [Griffiths and Steyvers, 2002, Buntine and Jakulin, 2004]. L'algorithme d'espérance-propagation [Minka and Lafferty, 2002] constitue une autre alternative.

(un thème par document) et à PLSA sur deux corpus (AP et CRAN). PLSA sans contrôle de convergence est pire que toutes les autres techniques. Avec contrôle de convergence, elle fait mieux que les modèles unigramme et de mélange de multinomiales, qui présentent des perplexités comparables. LDA est encore nettement meilleur et d'autant plus que le nombre de dimensions augmente.

Sur des applications plus concrètes, comme de la classification supervisée ou du filtrage collaboratif[20], LDA est meilleur que les deux modèles les plus basiques, bien que de façon moins marquée. Malheureusement, le modèle n'est cette fois pas comparé à PLSA.

En conclusion, bien que LDA soit un modèle théoriquement mieux justifié que PLSA, le fait de tirer un thème pour chaque mot semble en contradiction avec une notion intuitive de continuité dans le discours. Comment interprète-t-on une phrase qui contient 10 thèmes différents par exemple, une situation probable si l'on suit la recommandation des auteurs de considérer un grand nombre n_T de thèmes possibles ? La complexité du modèle induit aussi des difficultés de calcul, qui obligent à effectuer plusieurs approximations dans l'estimation des paramètres. Cependant, au final, la qualité des performances obtenues est un argument de poids en faveur de LDA, qui en fait un algorithme à inclure dans une série de tests comparatifs sur la classification textuelle non supervisée. Nous reviendrons plus en détail sur ce modèle en section 5.2.

2.3.5 Gamma-Poisson (GaP)

La motivation de [Canny, 2004] pour proposer le modèle Gamma-Poisson est de présenter un intermédiaire entre les modèles situant le thème au niveau des documents (mélange de multinomiales) et ceux qui le situent au niveau des occurrences (PLSA, LDA) en se focalisant sur des parties de textes. L'intuition semble naturelle de concéder qu'un document n'est pas totalement monothématique sans pour autant accepter qu'il soit composé de tous les thèmes du corpus.

Le modèle génératif de GaP est le suivant :

1. Pour chaque document $d = 1, \ldots, n_D$,
   (a) Pour chaque thème $t = 1, \ldots, n_T$, tirer la longueur totale des passages du document d traitant du thème t : $L_{dt} \sim \text{Gamma}(a_t, c_t/a_t)$
   (b) Pour chaque mot du vocabulaire $w = 1, \ldots, n_W$, $C_{wd} \sim \text{Poisson}(\sum_{t=1}^{n_T} L_{dt}\beta_{tw})$

Les paramètres du modèle sont donc en théorie :

$$\Theta = ((a_t)_{t=1,\ldots,n_T}, (c_t)_{t=1,\ldots,n_T}, (\beta_{tw})_{t=1,\ldots,n_T, w=1,\ldots,n_W})$$

En pratique cependant, a et c sont considérés fixés et seuls les β feront l'objet d'un algorithme d'inférence.

L'originalité de ce modèle réside dans l'intervention de la variable L_{dt}, qui, de même que μ_{dt} dans les sections précédentes, a pour vocation de lier documents et thèmes. Mais il ne s'agit plus ici d'une probabilité pour un thème d'apparaître dans un document mais plutôt de la longueur, en nombre d'occurrences, de l'ensemble des passages du document d traitant du thème t et donc d'une variable entière positive. L'article précise néanmoins

[20] Il s'agit, dans cette expérience, de prédire, à partir d'un ensemble de films donnés pour un utilisateur, un autre film qu'il appréciera. Le modèle est entrainé sur un jeu complet d'utilisateurs avec leurs films favoris et testé sur un autre ensemble d'utilisateurs. Ce problème est analogue à celui de la classification de textes si l'on regarde chaque texte comme un utilisateur et les mots qui le composent comme ses films préférés.

qu'en jouant sur le facteur d'échelle c, il est possible de transformer L_{dt} en une probabilité thématique, ce qui la rapprocherait du rôle des μ_{dt} dans PLSA et LDA. β_{tw} est toujours, en revanche, la probabilité d'apparition du mot w dans le thème t.

Dans la mesure où le paramètre de la loi de Poisson équivaut à l'espérance de la variable, $\sum_{t=1}^{n_T} L_{dt}\beta_{tw}$ peut être vu comme une approximation de C_{wd}. [Canny, 2004] souligne que GaP peut ainsi être rapproché des problèmes de factorisation matricielle type NMF avec l'approximation : $C \approx l\beta$. Cette analogie sera reprise pour les besoins de l'inférence.

Canny justifie le choix de la loi Gamma de manière empirique en estimant les L_{dt} sur un corpus par des méthodes heuristiques. L'importance du facteur de forme a est également évoquée. Canny conjecture que ce facteur, s'il est correctement appris, peut donner une indication sur le nombre moyen de passages concernant un thème donné dans les documents. Néanmoins, dans le contexte de [Canny, 2004], a est supposé fixé et l'expérience suggère qu'il vaut mieux le sous-estimer que le sur-estimer (typiquement, $a = 1.1$ dans les expériences présentées).

Comme dans l'étude des modèles probabilistes précédents, la log-vraisemblance n'est pas maximisable directement en les paramètres. Le calcul de la log-vraisemblance complète pour des longueurs thématiques L_{dt} fixées donne :

$$\begin{aligned}
\mathcal{L}^c &= \sum_{d=1}^{n_D} \log \mathrm{P}(C_d, L_d; a, c, \beta) \\
&= \sum_{d=1}^{n_D} \log \left(\prod_{w=1}^{n_W} \mathrm{P}(C_{wd}|L_d;\beta) \prod_{t=1}^{n_T} p(L_{dt}; a_t, c_t) \right) \\
&= \sum_{d=1}^{n_D} \log \left(\prod_{w=1}^{n_W} \frac{e^{-\sum_{t=1}^{n_T} L_{dt}\beta_{tw}} (\sum_{t=1}^{n_T} L_{dt}\beta_{tw})^{C_{wd}}}{C_{wd}!} \prod_{t=1}^{n_T} \frac{(L_{dt}a_t/c_t)^{a_t-1} e^{-L_{dt}a_t/c_t}}{\Gamma(a_t)c_t/a_t} \right) \\
&= -\sum_{d=1}^{n_D}\sum_{t=1}^{n_T} L_{dt}(\sum_{w=1}^{n_W} \beta_{tw} + \frac{a_t}{c_t}) + \sum_{d=1}^{n_D}\sum_{w=1}^{n_W} C_{wd} \log(\sum_{t=1}^{n_T} L_{dt}\beta_{tw}) \\
&\qquad + \sum_{d=1}^{n_D}\sum_{t=1}^{n_T} (a_t - 1)\log L_{dt} + n_D \sum_{t=1}^{n_T} \log \frac{(a_t/c_t)^{a_t}}{\Gamma(a_t)} - \sum_{w=1}^{n_W}\sum_{d=1}^{n_D} \log(C_{wd}!)
\end{aligned}$$

L'idée initiale de [Canny, 2004] est d'appliquer l'algorithme EM mais il s'avère que le calcul de l'espérance des L_{dt} pose problème et Canny propose donc de remplacer l'étape E par une maximisation des L_{dt} en mettant en avant la forme des distributions expérimentales (la densité est très concentrée autour d'un point).

Une maximisation directe n'est pas possible mais Canny remarque que la minimisation de $-\mathcal{L}^c$ en L_{dt} peut être effectuée à l'aide d'une étape de l'algorithme NMF en reconnaissant dans les termes de $-\mathcal{L}^c$ qui concernent β la divergence de Kullback-Leibler $D(F||WH)$ avec $H \leftrightarrow l$ et des matrices F et W de dimensions respectives $(n_W+n_T)\times n_D$ et $(n_W+n_T)\times n_T$:

$$F \leftrightarrow \begin{pmatrix} & C & \\ a_1 - 1 & & \\ & \ddots & \\ & & a_{n_T} - 1 \end{pmatrix}$$

$$W \leftrightarrow \begin{pmatrix} & \beta & \\ a_1/c_1 & & \\ & \ddots & \\ & & a_{n_T}/c_{n_T} \end{pmatrix}$$

Par conséquent, en s'inspirant de la mise à jour de NMF [Lee and Seung, 2001] :

$$L_{dt} \leftarrow L_{dt} \left(\sum_{w=1}^{n_W} \frac{C_{wd}}{\sum_{t'=1}^{n_T} L_{dt'}\beta_{t'w}} \beta_{tw} + \frac{a_t - 1}{L_{dt}} \right) / \left(\sum_{w=1}^{n_W} \beta_{tw} + \frac{a_t}{c_t} \right)$$

Pour l'étape M, Canny préconise également d'appliquer une mise à jour de NMF, en reconnaissant cette fois directement la divergence de Kullback-Leibler relative à l'approximation $C \approx l\beta$ pour minimiser en β. Nous obtenons alors :

$$\beta_{tw} \leftarrow \beta_{tw} \left(\sum_{d=1}^{n_D} \frac{C_{wd}}{\sum_{t'=1}^{n_T} L_{dt'}\beta_{t'w}} L_{dt} \right) / \left(\sum_{d=1}^{n_D} L_{dt} \right)$$

Bien que nous ayons insisté en section 2.3.1 sur leur similarité, nous terminons cette section en soulignant une différence claire entre les modélisations par loi multinomiale (2.4) et par distributions de Poisson (2.5). Cette divergence tient à la prise en compte de la longueur des textes :

- pour la loi multinomiale, la longueur du texte est un paramètre (qui n'est d'ailleurs pas estimé, mais considéré comme donné dans le modèle de la section 2.3.2), de sorte qu'il est possible, après estimation des β, de générer exactement un texte d'une longueur quelconque ;
- pour les lois de Poisson, en revanche, les longueurs des documents sont prises en compte de façon implicite par les λ : puisque le paramètre de la loi mesure l'espérance des observations, la somme des λ représente la longueur moyenne des textes. Mais il n'est pas possible d'être plus spécifique et de tenir compte d'une éventuelle variabilité de longueur d'un texte à l'autre. Peut-être plus préoccupant, il n'est pas possible non plus à partir de paramètres estimés de simuler un document d'une longueur donnée.

Il n'est pas clair que le traitement de la longueur comme une variable exogène dans le modèle de mélange de multinomiales soit la meilleure façon de procéder[21], mais il nous semble toutefois que cette solution est nettement plus convaincante qu'une modélisation implicite (lois de Poisson, modélisation Bernoulli ou normalisation de chaque colonne de la matrice des comptes par sa somme, pour ne considérer que des fréquences). C'est pourquoi nous avons dans la suite de la thèse préféré la représentation « multinomiale » à la représentation « Poisson ».

En pratique, GaP obtient de meilleurs résultats que LDA avec l'inférence variationnelle tant pour la perplexité que sur une tâche de recherche d'information. L'intuition derrière le modèle est relativement convaincante mais nous relevons les deux problèmes principaux suivants : du point de vue de la modélisation, avec une loi de Poisson, il n'est pas possible de générer un document qui soit exactement de la longueur attendue ; la stratégie d'inférence souffre d'un trop grand nombre d'approximations, pour au final s'inspirer fortement de l'algorithme NMF (auquel le modèle n'est d'ailleurs pas comparé expérimentalement). Pour nuancer ce dernier point, signalons que [Buntine and Jakulin, 2006] montre comment utiliser l'inférence variationnelle et l'échantillonneur de Gibbs dans le cadre de GaP.

[21] Il serait également possible de la modéliser par une loi de Poisson, comme dans [Blei et al., 2002].

2.3.6 Organisation hiérarchique

MASHA Il existe de nombreuses façons d'étendre à des hiérarchies thématiques les modèles probabilistes pour la classification. Dans la mesure où aucune ne semble réellement s'imposer comme étant meilleure que les autres, cette problématique de recherche reste particulièrement active depuis les premières tentatives, par exemple [Bishop and Tipping, 1998, Baker et al., 1999]. Nous nous intéressons à l'extension proposée dans [Vinokourov and Girolami, 2002] pour le modèle de mélange de multinomiales. Cet article repose sur l'hypothèse que les modèles hiérarchiques, même s'ils sont théoriquement équivalents à leurs homologues plats (car un mélange de mélanges est un mélange), sont en pratique plus efficaces car ils suivent un principe de diviser-pour-régner dans l'estimation des paramètres. Ainsi, pour un humain ayant à faire une tâche de regroupement, il est plus facile de diviser grossièrement en un nombre limité de tas dans un premier temps avant éventuellement de se ré-intéresser aux tas les plus hétérogènes. En outre, dans un but d'analyse exploratoire, le fait de pouvoir dynamiquement modifier la hiérarchie de l'arbre pour en explorer un aspect particulier semble attractif.

L'article propose deux modèles : l'un inspiré de PLSA, Hierarchical Probabilistic Latent Semantic Analysis, symétrique dans le rôle que jouent les mots et les documents, et l'autre, Multinomial Asymmetric Hierarchical Analysis (MASHA), asymétrique, dans lequel tous les mots d'un même document appartiennent au même groupe thématique, à relier au modèle de mélange de multinomiales de la section 2.3.2.

On suppose donnée une hiérarchie τ qui a la forme d'un arbre de racine c_1 et qui compte n_C nœuds au total. Présentons d'abord le modèle génératif de MASHA pour un document d de longueur l_d supposée connue.

1. Tirer un thème T_d selon le procédé suivant :
   - On note c_k le nœud courant. $c_k \leftarrow c_1$.
   - Tant que c_k n'est pas une feuille, on note $c_k(1), \ldots, c_k(p)$ les p fils de c_k et on tire $c_{k+1} \sim \text{Mult}(1, (\alpha_{m+1}, \alpha_{m+2}, \ldots, \alpha_{m+p-1}, \alpha_{m+p}))$, α_{m+i} étant la probabilité, en étant au nœud c_k, de choisir le fils $c_k(i)$, avec la relation $\sum_{i=1}^{p} \alpha_{m+i} = 1$, m étant un indice dépendant de la façon de numéroter les branches. Le nombre total de paramètres α libres est le nombre de feuilles n_T moins 1[22]. Pour $t \in \{1, \ldots, n_T\}$, la succession $(c_1, \ldots, c_t)$, menant à la feuille c_t, est le thème du document.
2. Conditionnellement à T_d, tirer les l_d mots du texte $C_d \sim \text{Mult}(l_d, (\beta_{T_d 1}, \ldots, \beta_{T_d n_W}))$, β étant une matrice $n_C \times n_W$[23] de paramètres telle que

$$\forall t \in \{1, \ldots, n_C\}, \sum_{w=1}^{n_W} \beta_{tw} = 1.$$

Remarquons que si la profondeur de l'arbre n'est que de 1, on retrouve un modèle de mélange de multinomiales classique. La spécificité de cette méthode réside dans l'étape

[22] Ceci peut se vérifier en raisonnant couche par couche. Sur les feuilles, il y a autant de coefficients que de feuilles moins le nombre de parents sur la couche supérieure puisque pour chaque parent, les coefficients doivent sommer à 1 et on a donc un paramètre libre de moins. Sur la couche d'au-dessus, on a un nombre de paramètres égal au nombre de nœuds moins le nombre de nœuds de la couche au-dessus, etc... On a donc une simplification jusqu'à la racine de l'arbre, qui ne contient qu'un nœud c_1 et le nombre recherché est $n_T - 1$.

[23] Au final, seules les n_T dernières lignes relatives aux feuilles sont utilisées mais, dans la phase d'apprentissage, il est nécessaire d'associer des probabilités de mots à chaque nœud de l'arbre.

d'inférence des paramètres qui est effectuée nœud par nœud, au moyen de l'algorithme EM.

Pour tout nœud $c \in \tau$, on note τ_c l'ensemble des feuilles de l'arbre construit avec les c premiers nœuds et leurs fils. Dans un premier temps, on procède comme si l'arbre n'avait qu'une couche sous la racine. On retrouve alors les formules classiques d'application de l'EM pour un mélange de multinomiales pour la première couche. L'algorithme est appliqué jusqu'à convergence.

Supposons maintenant que nous soyons dans un nœud τ_l et que nous ayons évalué les α_c et β_{cw} relatifs aux couches supérieures. Nous les considérons à présent fixés et nous nous intéressons uniquement à ceux qui concernent les fils du nœud l. En modifiant légèrement la quantité de l'EM en une forme d'espérance conditionnelle de la log-vraisemblance complète, liée à l'appartenance des documents au nœud l, [Vinokourov and Girolami, 2002] propose d'appliquer l'algorithme pour évaluer les paramètres relatifs à ces nouveaux nœuds sans recalculer tous les autres. Mais l'approche adoptée dans l'article ne semble pas prendre en compte qu'un document puisse se trouver dans n'importe quelle feuille à chaque étape de construction de l'arbre. Il faut donc s'intéresser dans l'estimation de la quantité de l'EM à *toutes* les feuilles de τ_l et pas uniquement à celles qui viennent d'être ajoutée. Si l'on essaie de maximiser la quantité usuelle de l'EM, il faut donc accepter de ré-estimer au moins tous les paramètres liés aux feuilles et les expressions sont sensiblement plus compliquées. Le gain en temps de calcul mis en avant au départ grâce à la simplification du problème est alors loin d'être évident.

Sous MASHA, un texte ne peut appartenir qu'à une seule feuille. C'est la raison pour laquelle Vinokourov et Girolami proposent, en s'inspirant de PLSA, Hierarchical Probabilistic Latent Semantic Analysis (HPLSA) dont le modèle génératif est une combinaison de PLSA et de MASHA. La structure d'arbre est conservée mais les feuilles sont les thèmes des paires mot/document (w, d) et non plus simplement des documents. Le nombre de paramètres est plus important (puisque l'on doit rajouter une matrice α pour lier les documents aux thèmes) mais les équations de ré-estimation sont approximativement les mêmes que pour MASHA.

Nous avons déjà évoqué le problème de ré-estimation locale des paramètres. Une autre des objections que l'on peut formuler à l'encontre de ces modèles est qu'ils supposent l'existence d'un arbre a priori. Sur un problème réel, il paraît extrêmement difficile, à moins d'avoir une connaissance précise des données, de trouver une telle structure. En effet, alors que des choix simples comme le nombre de dimensions latentes ne sont déjà pas évidents, le nombre d'hypothèses à formuler a priori ici est encore bien plus grand.

Conscients que le choix de l'arbre est un problème, Vinokourov et Girolami proposent un critère de sélection de modèles. Ce *critère de complexité stochastique* consiste en fait à calculer la vraisemblance complète du modèle après avoir appliqué l'EM et en supposant que les probabilités a priori sont Dirichlet. Ainsi, après de multiples approximations dont les justifications sont purement empiriques, ils obtiennent une manière de comparer deux structures par rapport à la façon dont elles approchent les données. Cela dit, même si ce critère était théoriquement satisfaisant et aisément calculable, il semble en pratique bien difficile à appliquer : impossible en effet d'estimer les paramètres de l'EM pour tous les arbres possibles. Il faudrait un critère plus simple pour obtenir un modèle a priori, indépendamment des approximations effectuées par la suite, lors de l'estimation des paramètres. Ce problème rejoint d'ailleurs celui du choix du nombre de thèmes dans d'autres modèles.

La méthode d'évaluation repose sur un critère d'information mutuelle. Il s'agit de déterminer la proximité d'un regroupement avec un partitionnement de référence, constitué

de classes établies manuellement. La définition et le calcul de l'information mutuelle seront présentés dans le chapitre 3. Comme cela était prévisible dans l'argumentation de la section « choix du modèle », l'évaluation est faite en testant exhaustivement un sous-ensemble de modèles, à savoir tous les arbres équilibrés de profondeur 2, avec 5 à 7 fils et 2 à 4 petits fils par fils. Les résultats sont comparés avec les performances du modèle plat avec le même nombre de feuilles. Par ailleurs, le critère de complexité stochastique est calculé pour chaque modèle. Sauf exceptions, les résultats sont en général meilleurs pour les modèles hiérarchiques et MASHA est meilleur que HPLSA. Mais le nombre de fois où les modèles plats sont supérieurs n'est pas négligeable et sans corrélation avec le critère de complexité stochastique, qui, par ailleurs, n'identifie pas toujours les meilleures structures en termes d'information mutuelle. Le choix de la structure reste donc un problème critique.

HPLSA Il existe certaines variantes de PLSA, qui diffèrent peu au niveau du modèle mais bien plus au niveau de la méthode d'inférence, consistant à forcer certaines composantes à être commune à tous les documents (« bruit de fond ») [Zhai et al., 2004, Mei and Zhai, 2005]. Ces études constituent une première étape vers l'obtention d'un modèle hiérarchique à partir de PLSA. De façon plus marquée, [Gaussier et al., 2002] étend un modèle hiérarchique initialement présenté dans [Hofmann and Puzicha, 1998] et construit sur les mêmes bases que PLSA, l'idée étant ici que tous les nœuds de l'arbre correspondent à un thème (et sont à ce titre associés à une distribution sur le vocabulaire) et que les feuilles sont en plus considérées comme des classes, c'est-à-dire associées à des distributions sur les indices de documents. Le modèle de génération consiste à tirer les cooccurrences mots/documents, comme PLSA, en tirant successivement :

- un nœud-feuille ;
- un document selon cette classe ;
- un thème parmi l'ensemble des nœuds ancêtres de la classe, y compris elle-même ;
- un mot selon la distribution thématique correspondante.

La structure n'est pas figée mais construite simultanément à l'étape d'inférence, en utilisant les propriétés de l'algorithme choisi, à savoir l'*EM tempéré* (*tempered EM*). Utilisée également dans [Hofmann, 2001], cette méthode de *recuit simulé* (*simulated annealing*) consiste à agir sur l'aspect des probabilités a posteriori, par le biais d'un paramètre de « température ». L'effet recherché est en général d'éviter les optimums locaux. Ici, il s'agit également de tester la stabilité des thèmes pour construire l'arbre hiérarchique.

Cette étude a le mérite de proposer une solution pour l'élaboration de la structure. Néanmoins, se fondant sur PLSA, elle souffre des mêmes défauts de modélisation, notamment pour ce qui est des difficultés de généralisation à des documents non vus. D'autre part, l'algorithme EM tempéré, qui a l'inconvénient d'être plus inspiré par la pratique que par la théorie, rajoute un aspect arbitraire à la procédure d'inférence dans le réglage du paramètre de température.

HLDA Enfin, LDA bénéficie lui aussi d'extensions hiérarchiques.

Pour [Buntine and Jakulin, 2004], chaque nœud est un thème. La structure de l'arbre intervient dans le tirage du thème, pour lequel des paramètres supplémentaires sont introduits : il s'agit des probabilités de rester sur un thème ou de descendre vers ses fils. L'algorithme préconisé pour l'inférence est l'échantillonnage de Gibbs. Cette adaptation hiérarchique semble théoriquement bien justifiée mais souffre d'une certaine rigidité. Ainsi, la profondeur de l'arbre et le nombre de fils de chaque thème, constant, doivent être spécifiés avant application de l'algorithme, la structure n'étant pas modifiable dynamiquement.

Modèle	Référence(s)	Thème latent
Mélange de multinomiales	[Nigam et al., 2000] [Clérot et al., 2004]	Un par document
PLSA	[Hofmann, 2001]	Un par occurrence, par couple (w, d)
LDA	[Blei et al., 2002]	Un par occurrence, répartition des thèmes fixée au niveau du document
Gamma-Poisson	[Canny, 2004]	Répartition des thèmes fixée au niveau du document

Modèle	Inférence	Version hiérarchique
Mélange de multinomiales	EM	[Vinokourov and Girolami, 2002]
PLSA	EM	[Hofmann and Puzicha, 1998] [Gaussier et al., 2002]
LDA	Échantillonnage Gibbs Inférence variationnelle Espérance-propagation	[Buntine and Jakulin, 2004] [Blei et al., 2004]
Gamma-Poisson	Échantillonnage Gibbs Inférence variationnelle	–

TAB. 2.1 – Tableau récapitulatif sur les modèles probabilistes

[Blei et al., 2004] propose de construire l'arbre par un processus particulier (dit « du restaurant chinois »). La profondeur est fixée mais le nombre de fils potentiels de chaque nœud est infini, ce qui donne une grande flexibilité à la structure. Chaque document appartient à une feuille et ses thèmes sont tirés parmi le chemin des nœuds conduisant à cette feuille. LDA est ensuite normalement appliqué. L'échantillonnage de Gibbs est utilisé pour l'inférence, en alternance pour la structure et les thèmes. Il est toutefois légitime de se demander, compte tenu du très grand espace à explorer, si le nombre d'optimums locaux dans lequel peut rester l'échantillonneur de Gibbs ne nuit pas sérieusement à la qualité de l'inférence.

Outre les extensions hiérarchiques, d'autres variantes ont été proposées autour de LDA. Citons notamment le *modèle à thèmes corrélés* (*correlated topic model, CTM*) [Blei and Lafferty, 2006], qui propose d'utiliser un autre a priori sur les thèmes que Dirichlet. Une loi logistique normale permet de modéliser les co-apparitions de thèmes différents. La matrice de variance/covariance correspondante illustre les liens entre thèmes et peut être utilisée pour constituer un graphe sémantique représentatif du corpus. De même que pour LDA, l'estimation des paramètres est conduite par inférence variationnelle.

En guise de conclusion à cette section, nous proposons un tableau récapitulatif sur les modèles probabilistes, synthétisant les points communs et différences entre les méthodes présentées 2.1.

2.3.7 Retour sur la modélisation des comptes de mots

Dans cette section, nous nous interrogeons sur la pertinence de la modélisation des comptes de mots par des lois Poisson ou multinomiale. Nous verrons que cette hypothèse,

acceptée unanimement par les modèles probabilistes précédents, rend très mal compte de certaines spécificités des données textuelles.

2.3.7.1 Loi de Zipf

Avant de proposer une loi pour modéliser les données textuelles, il est nécessaire d'observer attentivement leurs particularités. Les comptes de mots dans un corpus suivent globalement une loi de Zipf, c'est-à-dire que, si nous organisons le vocabulaire par ordre décroissant de fréquence (de sorte que l'indice du mot soit aussi son rang), la probabilité d'apparition du mot d'indice w est proportionnelle à $1/w^{\alpha}$, où α est une constante proche de 1 [Zipf, 1949, Baayen, 2001]. En d'autres termes, et si nous approchons α par 1, le compte du mot w sur l'ensemble du corpus peut être estimé par $\frac{k}{w}$, k étant une constante à déterminer. Nous vérifions cette loi sur un échantillon[24] du corpus Reuters [Reuters, 2000], en fixant k au compte du mot le plus fréquent. Le nombre d'occurrences est $l = 1409016$ et la taille du vocabulaire $n_W = 43320$. La figure 2.2 montre, dans une représentation en double échelle logarithmique, une bonne adéquation entre les comptes observés et la prédiction « zipfienne ».

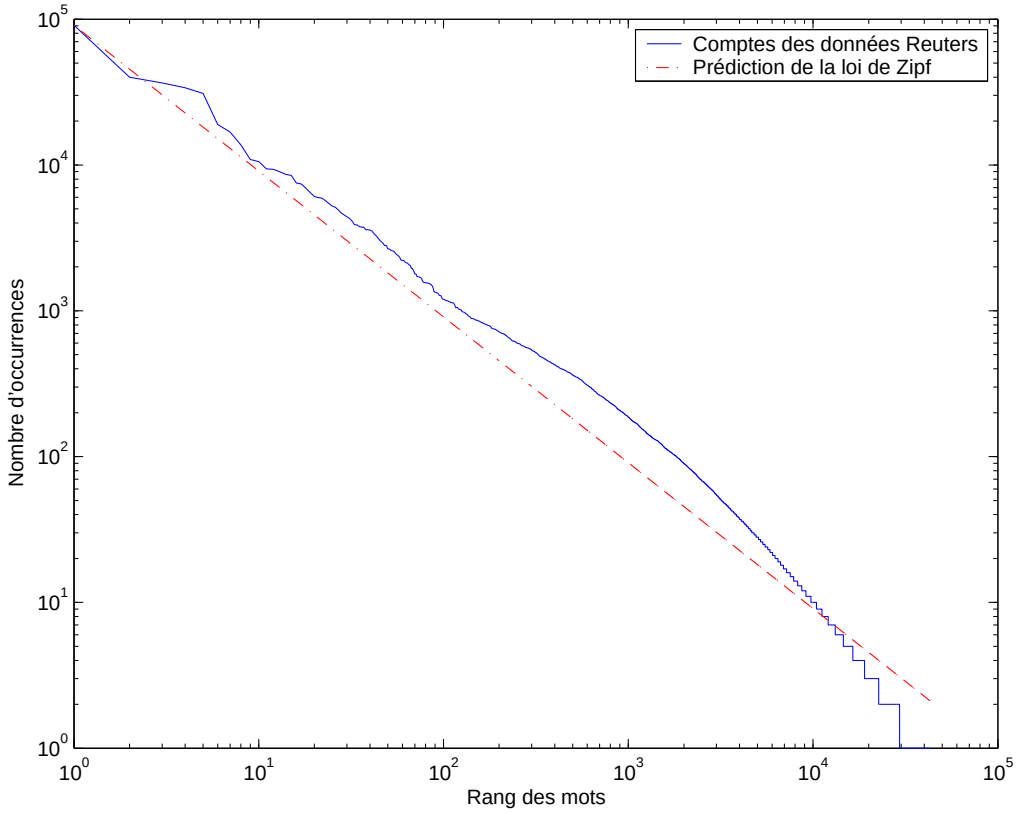

FIG. 2.2 – Nombre d'occurrences des mots dans le corpus en fonction de leur rang.

Il convient de garder à l'esprit certaines conséquences pratiques de cette observation, en particulier sur les relations entre le nombre de mots différents et le nombre d'occurrences dans le corpus. Ainsi, lorsque nous considérons la suppression d'une partie du vocabulaire, comme nous le ferons en section 4.3.1, l'impact en termes de nombres d'occurrences

[24] Voir la section 4.2.1 pour la formation détaillée de cet échantillon.

varie grandement suivant la nature des mots retirés : les 200 mots les plus fréquents ne représentent que 0.46% du vocabulaire mais 48.4% des occurrences. À l'inverse, les mots apparaissant 1 ou 2 fois représentent 47.6% du vocabulaire mais seulement 1.9% des occurrences. Dit autrement, il est tout autant possible de réduire les textes de moitié en ne supprimant que 200 mots sur 43320 que de les laisser quasiment inchangés en retirant la moitié des termes constituant le vocabulaire.

2.3.7.2 Spécificités de la modélisation des comptes de mots

Dans cette section, nous mettons en évidence des lacunes de la loi de Poisson pour la modélisation des comptes [Katz, 1996]. La similarité avec la loi multinomiale, mise en évidence en section 2.3.1, laisse penser qu'elle est sujette aux mêmes imprécisions.

Nous nous intéressons ici plus particulièrement à la critique [Church and Gale, 1995]. La validité de la modélisation des comptes par une loi de Poisson est ici mesurée sur le corpus Brown, assemblage hétéroclite de 500 documents d'origines très diverses (politique, religion, fiction, littérature classique par exemple). La démarche consiste à calculer diverses statistiques sur l'ensemble des textes et à comparer leur valeur à celle que l'on obtiendrait si les comptes suivaient une loi de Poisson. En s'appuyant sur ces résultats, Church et Gale identifient que les comptes observés sont la conjonction de deux phénomènes distincts : l'un relatif à la présence ou à l'absence d'un mot dans un texte et l'autre relatif à son *crépitement* (*burstiness* en anglais), au sens de l'article, c'est-à-dire le nombre de fois où il apparaît lorsqu'il est présent. Cette remarque conduit assez naturellement les auteurs à envisager des modèles alternatifs :

- la loi binomiale négative de paramètres (N, p) qui donne la probabilité d'attendre k échecs avant d'obtenir N succès (de probabilité individuelle p) :

$$\mathrm{P}_{\mathrm{BN}}(k) = \binom{N+k-1}{k} p^N (1-p)^k \,;$$

- le K-mélange dont le premier paramètre $\alpha \in (0,1)$ est spécifiquement consacré à la masse de probabilité en 0 alors que le secon $\beta > 0$ est lié à l'éclatement :

$$\mathrm{P}_{\mathrm{K}}(k) = (1-\alpha)\mathbb{1}_{\{k=0\}} + \frac{\alpha}{\beta+1}\left(\frac{\beta}{\beta+1}\right)^k \,;$$

- plus généralement, les deux lois précédentes peuvent être vues comme des cas particuliers de mélanges infinis de lois de Poisson pour des choix particuliers de densité de mélange Φ :

$$\mathrm{P}_{\Phi}(k) = \int_0^\infty \Phi(\theta) \frac{e^{-\theta}\theta^k}{k!} d\theta \,.$$

Parmi les phénomènes relevés par Church et Katz, une des questions qui a retenu notre attention est : pour un mot donné, combien de documents du corpus ont exactement $k \in \mathbb{N}$ occurrences ? Le résultat est à comparer avec les variations des fréquences $\mathrm{P}(C_{wD} = k)$ prédites par le modèle (en estimant les paramètres par la méthode des moments). Nous avons reproduit les expériences de [Church and Gale, 1995] sur le même échantillon du corpus Reuters que dans la section précédente : en figure 2.3, nous avons tracé un histogramme des documents sur la base du nombre d'occurrences (en abscisses) de quelques mots choisis au hasard. En ordonnée y, figure donc le nombre de documents contenant x fois le mot considéré. Les comptes réels sont représentés par des cercles et ils sont à comparer

avec les prédictions, c'est-à-dire avec la probabilité suivant les différents modèles que le mot apparaisse x fois dans un document, multiplié par le nombre total de textes. La raison pour laquelle ces quantités sont comparables vient de la loi des grands nombres. Ainsi, pour un mot donné $w \in \{1, \ldots, n_W\}$:

$$\begin{aligned} \forall k \in \mathbb{N}, \mathrm{P}(C_{wD} = k) &= E[\mathbb{1}_{\{C_{wD}=k\}}] \\ &= \lim_{n_D \to \infty} \frac{1}{n_D} \sum_{d=1}^{n_D} \mathbb{1}_{\{C_{wd}=k\}} \\ &= \lim_{n_D \to \infty} \frac{\text{Nombre de textes contenant } k \text{ fois le mot } w}{n_D} \end{aligned}$$

Par conséquent, avec un nombre de documents suffisants, l'histogramme devrait représenter parfaitement les probabilités correspondantes. Nous considérons ici un modèle de Poisson simple, dont le paramètre est estimé par la moyenne du nombre d'occurrences sur le corpus. Nous constatons des erreurs de prédiction importantes, notamment sur les grands nombres d'occurrences, où les probabilités sont très nettement sous-estimées. Une hypothèse expliquant cet écart est l'importance du cas $k = 0$: pour la plupart des mots, il y a beaucoup plus de documents dans lesquels le mot ne figure pas que de documents dans lesquels il figure au moins une fois.

D'où l'idée d'introduire un autre modèle de Poisson (« Poisson (0 exclus) ») où nous considérons uniquement les textes dans lesquels le mot apparaît. La probabilité d'absence s'en trouve logiquement sous-estimée mais, malheureusement, le reste de la distribution n'est également approchée que de façon médiocre. En revanche, si nous considérons une loi binomiale négative, en estimant ces paramètres par la méthode des moments comme dans [Church and Gale, 1995], les probabilités obtenues semblent plus fiables, moyennant l'introduction d'un paramètre supplémentaire par mot.

Les auteurs notent que cet effet dépend des thèmes et des mots considérés. Ainsi, les mots qui divergent le plus de la prédiction « Poisson » sont ceux qui, à nombre d'occurrences dans le corpus égal, se trouvent dans le moins de documents. Ils sont donc plus présents dans les textes dans lesquels ils apparaissent et il semble logique de supposer que leur importance en est accrue. L'analyse manuelle de quelques uns de ces termes dans [Church and Gale, 1995] montre que ce sont effectivement des mots porteurs de sens (*content words*).

Ce phénomène peut laisser penser qu'un modèle de mélange thématique de lois de Poisson serait moins sujet aux faiblesses évoquées ci-dessus. En effet, si nous supposons qu'un mot donné est principalement attaché à un thème unique et que nous ne considérons alors que des textes provenant de ce thème, le poids des documents pour lesquels $k = 0$ devrait être moins important. Il n'en est rien. Nous avons représenté en figure 2.4 les mêmes graphes que précédemment, mais en se restreignant aux textes provenant uniquement de la catégorie Reuters dans laquelle le mot est le plus souvent observé (si le terme se retrouve de façon forte dans deux catégories, comme *California*, nous avons représenté les deux). Même sur ces sous-corpus plus homogènes, la loi binomiale négative s'adapte mieux aux données que les autres.

S'il est incontestable que la modélisation par une loi binomiale négative est plus fidèle à la réalité, il faut tout de même nuancer l'importance pratique de cette amélioration. Ainsi, les graphiques précédents ont été conçus pour exagérer le désaccord entre la prédiction et la distribution réelle, notamment avec l'emploi de l'échelle semi-logarithmique. Le nombre de textes dans lesquels un mot donné apparaît plus de 10 fois est très limité (souvent 1 ou

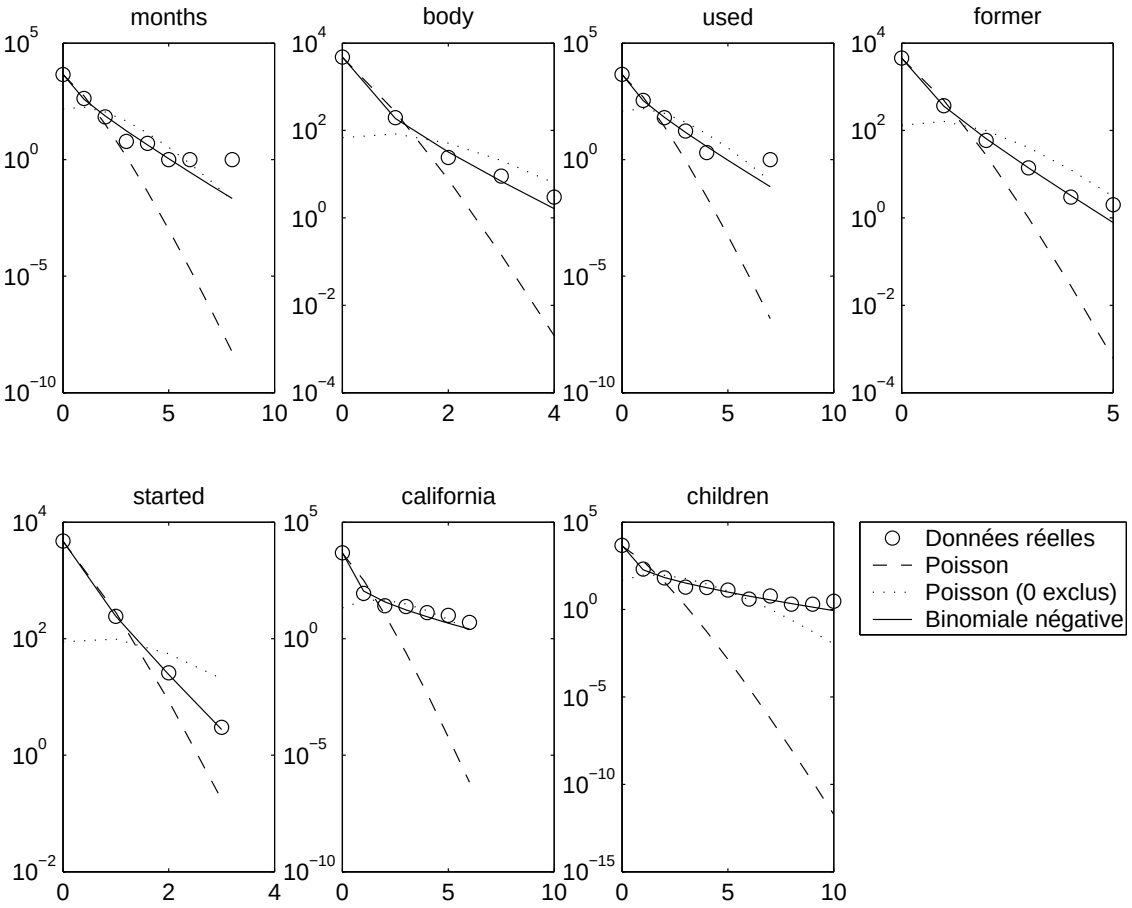

FIG. 2.3 – Ajustement de lois de Poisson et binomiale négative à des données réelles.

2 sur 5000). Même s'il est peut-être justement souhaitable que l'estimation des paramètres liés à ce terme soit la meilleure possible sur ces documents particuliers, il n'est pas sûr que cela justifie entièrement l'introduction de paramètres supplémentaires et, par conséquent, de méthodes d'inférence plus compliquées.

Nous avons, au cours de nos travaux de thèse, testé l'algorithme espérance-maximisation pour un modèle de mélange de lois binomiales négatives. La dérivation, assez similaire à celle du modèle de mélange de lois multinomiales, est développée en annexe (section A.1). Les premiers essais avec ce modèle n'ont malheureusement pas été concluants, le temps de calcul de l'EM modifié étant trop important pour pouvoir réaliser des expériences significatives. Cette voie d'amélioration de la modélisation, bien que prometteuse, nécessiterait donc d'approfondir un peu l'étape d'inférence. Il faut noter que ce modèle a également été utilisé depuis pour la classification supervisée dans [Airoldi et al., 2005].Dans la suite, ce sont les modélisations plus simples qui ont eu notre préférence.

2.4 Relations entre les modèles

De nombreuses relations existent entre les algorithmes présentés en sections 2.2 et 2.3. Outre les similarités de formulation entre PLSA et LSA évoquées par [Hofmann, 2001], il existe des rapprochements encore plus clairs entre les procédures d'inférence de PLSA et

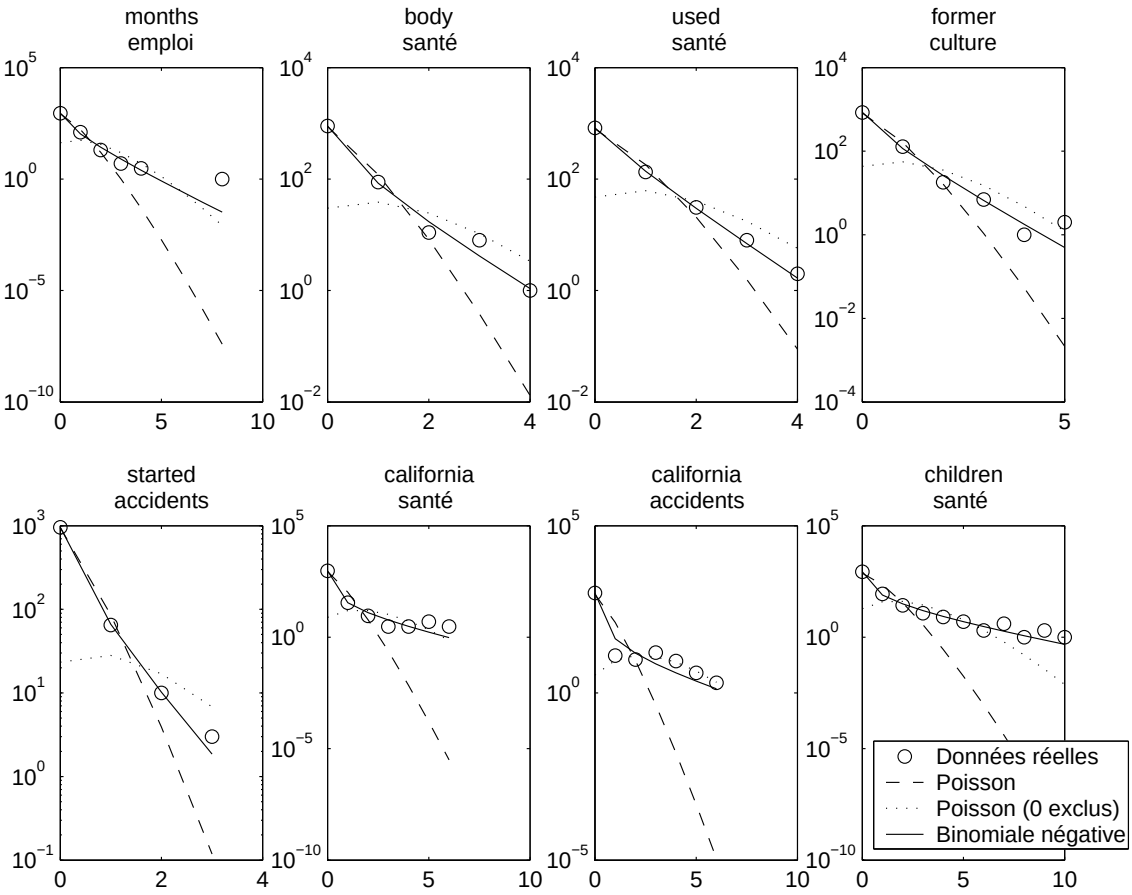

FIG. 2.4 – Ajustement des lois de Poisson et binomiale négative, en se restreignant aux documents issus d'une catégorie.

NMF et entre NMF et certaines versions du regroupement spectral. Par ailleurs, [Hofmann and Puzicha, 1998] analyse les liens entre différentes variantes du modèle de mélange de multinomiales et de PLSA. Les travaux de Wray Buntine [Buntine and Jakulin, 2004, 2006] fournissent un cadre unificateur aux modèles probabilistes, plus spécifiquement centré autour de LDA et GaP, qui permet de mieux détailler leurs similarités et différences.

2.4.1 Goulot d'information et mélange de multinomiales

Nous montrons à présent, en suivant [Slonim and Weiss, 2003], qu'il existe de fortes similarités entre l'algorithme itératif du goulot d'information (section 2.2.6) et l'algorithme EM appliqué au modèle de mélange de multinomiales (section 2.3.2).

Dans un premier temps, reprenons l'équation de mise à jour des probabilités d'appartenance aux thèmes pour le goulot d'information (2.1). Nous oublions la constante de normalisation et essayons de supprimer tous les facteurs multiplicatifs indépendants de t (puisqu'ils seront simplifiés par l'opération de normalisation) :

$$\mu_{dt} \quad \propto \quad \alpha_t \exp -\lambda \sum_{w=1}^{n_W} P(W=w|D=d) \log \frac{P(W=w|D=d)}{P(W=w|T=t)}$$

$$\begin{aligned}
&\propto \alpha_t \exp\lambda \sum_{w=1}^{n_W} P(W=w|D=d) \log P(W=w|T=t) \\
&\propto \alpha_t \exp\lambda \sum_{w=1}^{n_W} \frac{C_{wd}}{l_d} \log \beta_{tw} \\
&\propto \alpha_t \left(\prod_{w=1}^{n_W} \beta_{tw}^{C_{wd}} \right)^{\lambda/l_d}
\end{aligned}$$

Supposons dans un premier temps que les textes soient tous de même longueur, de sorte que l_d de dépende pas de d et fixons λ égal à cette longueur. La mise à jour de l'algorithme du goulot d'information devient alors identique à celle de l'EM :

$$\begin{aligned}
\mu_{dt} &\propto \alpha_t \prod_{w=1}^{n_W} \beta_{tw}^{C_{wd}} && \text{(goulot d'information)} \\
\mathrm{P}(T_d = t|C_d;\Theta') &\propto \alpha'_t \prod_{w=1}^{n_W} \beta_{tw}'^{C_{wd}} && \text{(EM)}
\end{aligned}$$

Dans ce cas de longueurs identiques, comme la méthode du goulot d'information consiste à estimer les probabilités relatives à W et D empiriquement sur la matrice de comptes, $\widehat{P}(D=d) = \frac{1}{n_D}$, ce qui établit l'équivalence entre :

$$\begin{aligned}
\alpha_t &= \sum_{d=1}^{n_D} \mu_{dt} P(D=d) && \text{(goulot d'information)} \\
\alpha_t &= \frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(T_d = t|C_d;\Theta') && \text{(EM)}
\end{aligned}$$

Enfin, puisque les équations de mise à jour de β se normalisent en w et $\widehat{P}(W=w, D=d) \propto C_{wd}$, la similarité des mises à jour de β est elle aussi évidente :

$$\begin{aligned}
\beta_{tw} &\propto \sum_{d=1}^{n_D} P(W=w, D=d)\mu_{dt} && \text{(goulot d'information)} \\
\beta_{tw} &\propto \sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T_d = t|C_d;\Theta') && \text{(EM)}
\end{aligned}$$

Lorsque les longueurs des documents ne sont pas homogènes, les deux algorithmes ne sont pas strictement équivalents mais l'article [Slonim and Weiss, 2003] montre expérimentalement, en fixant $\lambda = \frac{l}{n_D}$ qu'ils trouvent souvent des solutions proches et ce d'autant plus que les corpus sont grands. Il est intéressant de remarquer que lorsque les solutions diffèrent, aucune des deux n'est systématiquement meilleure que l'autre, au sens de l'optimisation des fonctions objectifs.

Enfin, nous constatons que le paramètre λ, en tant qu'exposant du produit des β a une influence sur la convergence de l'algorithme itératif, qui est également connu pour l'algorithme EM. De façon similaire, il existe une version modifiée de l'algorithme EM, dite *EM tempéré*, qui par l'introduction similaire d'un exposant, modifie la vitesse de convergence dans l'espoir d'éviter les optimums locaux (cette technique est par exemple utilisée dans [Hofmann, 2001]). Néanmoins, l'introduction et le réglage du paramètre de température semblant particulièrement arbitraire, nous avons cherché d'autres stratégies pour contourner les optimums locaux dans le cadre de cette thèse (chapitre 4).

2.4.2 NMF et PLSA

[Gaussier and Goutte, 2005] met en évidence une similarité importante entre l'application de l'algorithme EM à PLSA et l'algorithme NMF avec la divergence de Kullback-Leibler. Pour démontrer cette équivalence, écrivons les équations de mise à jour de l'étape M de PLSA (2.10) et (2.11) en injectant directement les résultats de l'étape E :

$$\begin{aligned}\beta_{tw} &= \frac{\sum_{d=1}^{n_D} C_{wd} \frac{\alpha'_t \beta'_{tw} \mu'_{td}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \beta'_{t'w} \mu'_{t'd}}}{l\alpha_t} \\ \mu_{td} &= \frac{\sum_{w=1}^{n_W} C_{wd} \frac{\alpha'_t \beta'_{tw} \mu'_{td}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \beta'_{t'w} \mu'_{t'd}}}{l\alpha_t}\end{aligned}$$

ce qui permet de déduire :

$$\mu_{td} = \mu'_{td} \frac{\sum_{w=1}^{n_W} l\alpha'_t \beta'_{tw} C_{wd} / \sum_{t'=1}^{n_T} l\alpha'_{t'} \beta'_{t'w} \mu'_{t'd}}{l\alpha_t} \tag{2.12}$$

$$l\alpha_t \beta_{tw} = l\alpha'_t \beta'_{tw} \sum_{d=1}^{n_D} \mu'_{td} C_{wd} / \sum_{t'=1}^{n_T} l\alpha'_{t'} \beta'_{t'w} \mu'_{t'd} \tag{2.13}$$

Réécrivons à présent avec des notations similaires les équations de mise à jour de NMF (les paramètres à l'étape précédente sont marqués d'un signe $'$ et les multiplications matricielles sont développées) :

$$H_{td} = H'_{td} \frac{\sum_{w=1}^{n_W} W'_{wt} C_{wd} / \sum_{t'=1}^{n_T} W'_{wt'} H'_{t'd}}{\sum_{w=1}^{n_W} W'_{wt}} \tag{2.14}$$

$$W_{wt} = W'_{wt} \frac{\sum_{d=1}^{n_D} H'_{td} C_{wd} / \sum_{t'=1}^{n_T} W'_{wt'} H'_{t'd}}{\sum_{d=1}^{n_D} H'_{td}} \tag{2.15}$$

Rappelons à présent de l'équation d'approximation de la matrice de comptes de PLSA, utilisée pour établir une analogie avec LSA : $l_d \mathrm{P}(W = w|D = d; \Theta) = \sum_{t=1}^{n_T} l\alpha_t \beta_{tw} \mu_{td}$. D'où l'approximation $C \approx WH$ avec les équivalences $W_{wt} \leftrightarrow l\alpha_t \beta_{tw}$ et $H_{td} \leftrightarrow \mu_{td}$.

Les équations de mise à jour peuvent se réécrire en suivant cette analogie. Supposons que nous réglions le problème de la normalisation de NMF en faisant sommer les lignes de H (ou p) à 1. Il est alors possible de montrer par récurrence que les équations de mise à jour propagent cette contrainte $\sum_{d=1}^{n_D} H_{td} = 1$ ($\sum_{d=1}^{n_D} \mu_{td} = 1$) ainsi qu'une autre sur les colonnes de W qui lui est liée : $\sum_{w=1}^{n_W} W_{wt} = \sum_{w=1}^{n_W} W'_{wt}$ ($\sum_{w=1}^{n_W} l\alpha_t \beta_{tw} = \sum_{w=1}^{n_W} l\alpha'_t \beta'_{tw} = l\alpha'_t$ si les β sont normalisés comme des probabilités).

Cette normalisation permet d'obtenir une équivalence parfaite entre (2.13) et (2.15). Pour (2.12) et (2.14), en revanche, nous observons que le dénominateur est différent : plus précisément, il s'agit dans (2.12) de la nouvelle valeur $l\alpha_t = \sum_{w=1}^{n_W} l\alpha_t \beta_{tw}$ et non de celle de l'étape précédente $l\alpha'_t$, comme dans (2.14). Par conséquent, il n'est pas possible de conclure à une équivalence parfaite et c'est probablement la raison pour laquelle [Ding et al., 2006] rapporte des résultats différents de PLSA et NMF à partir d'initialisations identiques. Néanmoins, dans la majorité des cas, nous nous attendons à ce que les α_t n'évoluent que peu d'une itération sur l'autre et, par conséquent, nous pouvons affirmer que les résultats de l'inférence de PLSA avec l'algorithme EM et l'algorithme NMF pour la divergence de Kullback-Leibler doivent donner des résultats similaires et présenter les mêmes caractéristiques dans la plupart des cas, ce que [Ding et al., 2006] confirme. Retenons, en particulier,

la remarque de [Gaussier and Goutte, 2005] selon laquelle NMF doit souffrir de la présence de minimums locaux, tout comme l'EM en grande dimension.

Sur NMF enfin, signalons une autre série d'analogie présentée dans [Ding et al., 2005]. Les auteurs traduisent l'algorithme des K-moyennes et les différents algorithmes de regroupement spectral en termes de décompositions matricielles. Ils montrent une certaine ressemblance dans les fonctions objectifs, qui peuvent s'exprimer comme distances euclidiennes de matrices, et insistent sur les différences entre les algorithmes, liées aux contraintes de normalisation, d'orthogonalité ou aux dimensions privilégiées.

2.4.3 Liens entre les modèles probabilistes

Bien que moins forts que les analogies que nous avons présentées dans ces deux premières sections, il existe également, au sein des modèles probabilistes présentés dans la section 2.3, des liens entre les algorithmes d'inférence et des généralisations englobant plusieurs modèles, l'essentiel des remarques se concentrant sur LDA.

Un premier constat opéré par [Girolami and Kabán, 2003] est que la log-vraisemblance de PLSA

$$\mathcal{L} = \sum_{w=1}^{n_W} \sum_{d=1}^{n_D} C_{wd} \log \left(\sum_{t=1}^{n_T} \alpha_t \beta_{tw} \mu_{td} \right)$$

peut être rapprochée de la probabilité a posteriori de μ dans LDA (en considérant les μ_{dt} comme des paramètres du modèle, de la même façon que dans PLSA), si l'on suppose un a priori uniforme (tous les α_t égaux à 1) :

$$\begin{aligned} p(\mu|C,\beta) &\propto \mathrm{P}(C|\mu,\beta) \\ &\propto \prod_{d=1}^{n_D} \prod_{w=1}^{n_W} (\sum_{t=1}^{n_T} \beta_{tw}\mu_{dt})^{C_{wd}} \end{aligned}$$

Leur conclusion est qu'il est possible de retrouver la procédure d'inférence de PLSA en partant du modèle LDA, en inférant des valeurs $\widehat{\mu}_{dt}$ des μ_{dt} document par document par maximum a posteriori puis en déterminant les β_{tw} par maximum de vraisemblance conditionnellement aux $\widehat{\mu}_{dt}$. Cette analogie reste néanmoins assez intuitive dans l'article et les correspondances dans les mises à jour de l'algorithme EM ne sont pas développées.

Wray Buntine a effectué un important travail de synthèse sur LDA, notamment sur les différentes méthodes d'inférence. Il plaide d'ailleurs pour le terme *analyse en composantes principales discrète* (*discrete PCA*) ou *multinomiale* (*multinomial PCA*), pour mieux refléter l'analogie avec l'analyse en composantes principales traditionnelle. Cette dernière en effet découle naturellement d'un double hypothèse gaussienne (répartition des composantes et génération d'une observation à partir de cette répartition) alors que LDA peut être vu comme un équivalent discret, en plaçant un a priori Dirichlet sur la variable de répartition des thèmes μ et une distribution multinomiale sur C. Ainsi [Buntine, 2002] constate que le modèle génératif LDA peut aussi s'exprimer pour chaque document d de façon plus compacte :

1. Tirer $\mu_d \sim \mathrm{Dir}(\alpha)$
2. Conditionnellement à μ_d, tirer $C_d \sim \mathrm{Mult}(l_d, \sum_{t=1}^{n_T} \beta_{tw}\mu_{dt})$

Buntine montre par ailleurs comment retrouver une méthode d'inférence variationnelle analogue à celle de [Blei et al., 2002], en partant d'un cadre théorique plus général sur les familles exponentielles. Il constate que, en adaptant les normalisations, cet algorithme

d'inférence variationnelle correspond aux mises à jour de NMF et de PLSI. [Buntine and Jakulin, 2004] dresse également un parallèle avec l'analyse en composantes indépendantes, montrant que l'analogie peut être renforcée en modélisant la longueur des documents dans LDA. Il y a dans cet article une solide section expérimentale, étudiant notamment l'apport de la classification non supervisée par LDA pour la classification supervisée, identifiant les (rares) cas dans lequelles la méthode peut améliorer les performances des séparateurs à vaste marge (SVM), et son utilisation en recherche d'information.

Il y a également eu pour LDA un certain nombre de comparaisons de méthodes d'inférence. Alternativement à l'inférence variationnelle [Blei et al., 2002], [Griffiths and Steyvers, 2002] propose un *échantillonneur de Gibbs* et [Minka and Lafferty, 2002] introduit la méthode d'*espérance-propagation* pour ce modèle, d'ailleurs appelé dans cet article *modèle génératif à aspects* (*generative aspect model*). [Buntine and Jakulin, 2004] plaide pour l'utilisation de l'échantillonnage de Gibbs à l'issue d'un bilan sur l'inférence pour LDA. Nous y reviendrons en section 5.2.2.

Par ailleurs, [Buntine and Jakulin, 2006] étudie simultanément GaP et LDA dans un cadre unificateur. Il met en évidence une équivalence entre les deux modèles, sous certaines conditions (deuxième paramètre de la loi Gamma constant pour GaP et modélisation du nombre total de comptes l par une loi Poisson-Gamma dans LDA). Dans le même article, Buntine et Jakulin présentent les algorithmes d'inférence variationnelle et d'échantillonnage de Gibbs pour les deux modèles.

[Banerjee et al., 2004] identifie plusieurs similarités intéressantes entre différentes méthodes de classification non supervisée, notamment les K-moyennes et l'algorithme EM dans le cas de mélanges de familles exponentielles. Les auteurs montrent que la donnée d'une fonction convexe est suffisante pour calculer une divergence sur un espace, qui peut correspondre ou non à une distance, mais qui recouvre notamment les cas des distances euclidiennes, de Mahalanobis ou de la divergence Kullback-Leibler. Ils introduisent ensuite le concept d'information de Bregman et montrent que l'on peut dans ce cadre général appliquer l'algorithme EM pour établir un partitionnement probabiliste. Cette généralisation est intéressante puisqu'elle recouvre des cas souvent considérés en fouille de textes et donne un algorithme simple de classification non supervisée, dès lors que l'on trouve une divergence de Bregman adaptée à un problème.

2.5 Intérêt du modèle de mélange de multinomiales

Nous concluons l'exposé de ces nombreuses méthodes par une section justifiant notre choix, qui prendra effet au chapitre 4, de nous concentrer sur le modèle de mélange de multinomiales.

La décision d'écarter les méthodes non probabilistes a déjà été esquissée en section 1.2.4 :

- Les méthodes non probabilistes offrent rarement des solutions théoriquement satisfaisantes pour calculer les probabilités d'appartenance aux thèmes d'un document non vu dans l'ensemble d'apprentissage.
- Un inconvénient commun aux méthodes vectorielles est la difficulté de donner une interprétation intuitive aux différents espaces sémantiques. Pour LSA, il est toujours possible d'exprimer les sous-espaces latents comme une combinaison des dimensions de base (les termes du vocabulaire) mais cette combinaison n'est jamais sémantique-

ment interprétable[25]. Pour les méthodes à noyaux telles que les noyaux sémantiques latents ou le regroupement spectral, la question de l'interprétabilité des résultats se pose plus encore que dans LSA puisque les espaces propres sont déformés et il est par conséquent plus difficile de leur donner une interprétation sémantique.
- Les méthodes vectorielles nécessitent de trouver une représentation adaptée pour donner de bons résultats. Cette représentation est souvent associée à des décisions arbitraires, empiriquement justifiées mais théoriquement peu satisfaisantes. La transformation idf, présentée sous de très nombreuses formes différentes dans la littérature, en est un des exemples les plus parlants.

À ces observations, s'ajoutent par ailleurs des questions de convergence, d'unicité de la solution, de robustesse pour certains algorithmes non probabilistes (NMF notamment).

La vision des modèles probabilistes, consistant à modéliser le comportement des données textuelles telles qu'elles sont observées, et non par le biais de multiples transformations, nous semble intuitivement plus satisfaisante. Outre les capacités de généralisation des modèles et l'interprétatibilité des thèmes, le cadre probabiliste s'adapte naturellement à un certain nombre de problèmes connexes, tels que l'apprentissage semi-supervisé et le choix du nombre de thèmes (par la théorie de sélection de modèles).

Parmi les modèles probabilistes, PLSA a l'inconvénient de ne pas être un modèle génératif, ce qui rend la généralisation théoriquement impossible. De nombreuses autres méthodes fondées sur des modèles réellement génératifs ont été développées (GaP, LDA et extensions hiérarchiques) et le sont encore (voir par exemple dernièrement le modèle auto-adaptatif Poisson [Gehler et al., 2006], avec inférence par échantillonnage de Gibbs). Cependant, il nous semble qu'elles peuvent toutes être vues comme des extensions du modèle de mélange de multinomiales de base, dans la mesure où elles consistent à ajouter des variables latentes de thèmes. Pourtant il ne paraît pas évident que ce modèle initial ait été suffisamment étudié et que ses limites soient parfaitement connues. La littérature existante s'interroge peu sur la qualité de l'inférence. Dans un contexte où le nombre de paramètres est très important et où la loi de Zipf permet de prédire qu'une grande partie d'entre eux seront très mal estimés, il nous paraît essentiel d'évaluer spécifiquement l'impact de l'étape d'inférence. Nous nous intéresserons en particulier à la convergence de l'EM et au problème de l'initialisation.

Ainsi, au lieu de recourir à l'ajout de niveaux de complexité sur le modèle, qui imposent, nous l'avons vu plus haut, des difficultés calculatoires considérables, il nous paraît souhaitable de s'interroger sur l'étape d'inférence dans le modèle de base, d'évaluer les performances de l'algorithme EM et de s'intéresser à des alternatives (chapitre 4). En d'autres termes, il nous semble que le rejet brutal de l'hypothèse de monothématicité motivant l'introduction de LDA est un peu excessif a priori, surtout compte tenu de l'excès inverse, peu intuitif, qu'il produit (une multi-thématicité extrême, avec une variable latente par occurrence).

[25] Difficile en effet de donner un sens à un thème qui serait par exemple : $4 \times$ cinéma $- 1.2 \times$ tour $+ 3.8 \times$ moutarde...

Chapitre 3

Évaluation

3.1 Introduction

Nous abordons à présent le problème de l'évaluation, dans une perspective générale, et avec l'objectif de définir un cadre d'étude pour le modèle de mélange de multinomiales (2.3.2) étudié dans le chapitre suivant. Parmi le vaste éventail de techniques utilisées dans la littérature, nous avons par conséquent privilégié celles qui ont le plus de sens dans le cadre d'un modèle probabiliste. Néanmoins, nous tâcherons de rester aussi général que possible dans la présentation et la définition des mesures.

Pour comparer l'importance des différents paramètres tels que l'initialisation, le choix du vocabulaire ou le nombre d'itérations de l'algorithme EM, il est indispensable de disposer d'une ou plusieurs méthodes d'évaluation, tout en étant conscient de leurs biais et limites. En effet, en classification non supervisée de documents, comme dans beaucoup d'autres tâches de traitement du langage [Galliers and Jones, 1995], il n'y a pas véritablement d'unanimité sur le protocole d'évaluation le mieux adapté. Le choix d'une mesure d'évaluation particulière n'est par conséquent jamais naïf et traduit un point de vue de l'expérimentateur sur la tâche. Ainsi, l'utilisation de critères généraux de classification non supervisée suppose une certaine confiance dans la représentation choisie puisque ces mesures font directement appel à une distance entre les documents. Le cadre probabiliste, quant à lui, permet de garder trace d'une autre mesure de qualité, la log-vraisemblance, ou d'une de ses variantes *la perplexité*, focalisant sur l'adéquation du modèle aux données, plus que sur la tâche finale. Enfin, la mesure d'une similarité avec une classification de référence néglige le problème de l'existence de multiples classifications sur un corpus, potentiellement très différentes les unes des autres mais toutes également pertinentes d'un certain point de vue. Une question que nous abordons peu dans ce chapitre est celle de l'évaluation « manuelle » subjective, consistant à présenter le résultat de la classification à un observateur humain et à le laisser juge de sa pertinence. Cette procédure, difficile à quantifier, est aussi, pragmatiquement, trop coûteuse pour pouvoir être appliquée de façon systématique. Nous reviendrons toutefois sur le problème connexe de l'interprétation des thèmes dans la section 5.3.

Ce chapitre est articulé comme suit. La première section présente brièvement les mesures communément utilisées dans le cadre de la classification non supervisée, pour des applications différentes de l'analyse textuelle. Nous verrons pourquoi leur utilisation est difficile dans notre cas. Puis nous présentons la perplexité. Nous montrons en quoi le lien de la perplexité avec la théorie de l'information en fait une mesure particulièrement naturelle dans notre contexte. D'autres travaux suggèrent de tirer partie de mesures d'erreur

éprouvées dans d'autres cadres, tel celui de la classification supervisée, en utilisant l'algorithme de partitionnement comme une étape d'un processus plus vaste. C'est l'objet de l'évaluation *extrinsèque* que nous étudierons dans la section 3.4. Enfin, la dernière idée exploitable pour l'évaluation de la classification non supervisée est de comparer avec un classement déjà existant (en général établi manuellement), en lequel nous avons une certaine confiance. Nous considérons deux mesures à cette fin : l'*information mutuelle* et le compte des cooccurrences avec la *méthode hongroise*. La comparaison de la partition établie manuellement par des humains et de celle que suggère le modèle permet, d'une certaine façon, de quantifier sa qualité. Enfin, nous concluons le chapitre par une discussion générale sur le cadre d'évaluation que nous utilisons dans les chapitres suivants.

3.2 Mesures générales en classification non supervisée

Les articles qui traitent de la classification non supervisée dans sa plus grande généralité sont nombreux, en particulier dans le domaine de l'apprentissage statistique. Le problème de l'évaluation y est souvent abordé sous l'appellation « validation » [Maulik and Bandyopadhyay, 2002, Halkidi et al., 2001]. Une tendance répandue est de considérer que les données d'un problème de classification non supervisée sont les exemples à ordonner d'une part et une distance sur l'espace dans lequel ils évoluent d'autre part. En faisant confiance à cette topologie, l'évaluation peut consister à mesurer conjointement la *compacité* de chaque groupe et la *distance inter-groupes*. Intuitivement, nous souhaitons que chaque objet soit le plus proche possible des individus qui sont dans son groupe et le plus loin possible de tous les autres.

La mise en œuvre la plus immédiate de ce principe consiste à observer les *inerties intra- et inter-classes* [Bisson, 2001]. Ainsi, dans le cas de la classification déterministe, supposons que les observations $x_1, \ldots, x_n$ fassent partie d'un espace vectoriel muni d'une distance $d(\cdot,\cdot)$. Nous pouvons alors définir le centroïde $g_t = \frac{1}{|G_t|}\sum_{x\in G_t} x$ de chaque classe (ou thème) G_t et le centroïde global $g = \frac{1}{n_T}\sum_{t=1}^{n_T} g_t$. Les définitions de l'inertie intra-classe I_a et l'inertie inter-classe I_e sont alors :

$$\begin{aligned} I_a &= \sum_{t=1}^{n_T}\sum_{x\in G_t} d(x,g_t)^2 \\ I_e &= \sum_{t=1}^{n_T} |G_t| d(g_t,g)^2 \end{aligned}$$

Or, dans une telle situation le théorème de König-Huygens stipule que la somme de ces deux inerties est constante et égale à l'inertie totale $\sum_{i=1}^{n} d(x_i,g)^2$. Cette quantité ne dépendant pas du partitionnement choisi, les objectifs de maximiser l'inertie inter-classes et de minimiser l'inertie intra-classes sont dans ce cas identiques [Bisson, 2001].

Néanmoins, en définissant de manière différente les inerties, il est possible de proposer d'autres mesures récompensant une variance intra-groupe (on dit encore le diamètre ou la dispersion intra-groupe) réduite et une variance inter-groupe (que l'on qualifie alternativement de distance entre ensembles ou de dispersion inter-groupe) élevée. Citons par exemple ceux qui sont connus dans la littérature sous les noms de Davies-Bouldin, Dunn, Calinski-Harabasz ou Xie-Beni [Maulik and Bandyopadhyay, 2002]. Dans tous les cas, ces indices sont hautement dépendants du choix de la distance. Or, pour notre application, nous avons déjà insisté dans le chapitre 2 sur le fait que ce problème de distance est particulièrement

aigü, sinon insurmontable. Nous avons donc eu tendance à éliminer toutes ces mesures, forcément biaisées en faveur de méthodes utilisant la même distance pour l'apprentissage de la classification et pour son évaluation.

Pour être complets, citons également dans cette section les indices qui ne mesurent qu'un aspect particulièrement ciblé du résultat de la classification. Ainsi, pour une classification probabiliste, [Halkidi et al., 2001] mentionne les coefficient de partition et coefficient de partition entropique qui mesurent tous deux l'aspect déterministe du regroupement. L'hypothèse sous-jacente semble être que la qualité de la classification est directement corrélée avec l'aptitude à placer un document dans un thème unique, c'est-à-dire la tendance de l'algorithme à retourner des classifications presque déterministes. La pertinence de cette affirmation ne paraît pas évidente a priori.

3.3 Perplexité

3.3.1 Mesure de prédiction des données

Une façon simple d'établir la valeur d'un modèle probabiliste est de chercher comment il *prédit* de nouveaux documents après apprentissage des paramètres. Ces capacités de prédicitions peuvent être évaluées avec la perplexité, mesure introduite en modélisation du langage [Jelinek, 1997] et reprise notamment par [Hofmann, 2001] et [Blei et al., 2002].

Considérons une suite de variables aléatoires $X_1, \ldots, X_n$ ($n \in \mathbb{N}$) indépendantes et identiquement distribuées (i.i.d.), suivant une densité de probabilité p supposée inconnue, et donc de même entropie $H_p(X)$, une application de la loi faible des grands nombres, appelée « propriété d'équipartition asymptotique » [Cover and Thomas, 1990], montre :

$$\begin{aligned}\lim_{n\to\infty} -\frac{1}{n}\log p(X_1,\ldots,X_n) &= \lim_{n\to\infty}\sum_{i=1}^{n} -\frac{1}{n}\log p(X_i)\\ &= E_p[-\log p(X)]\\ &= H_p(X)\end{aligned}$$

Ce qui nous intéresse en fait, c'est d'estimer la dissimilarité entre une distribution proposée q et la distribution réelle p. On peut utiliser à cet effet la divergence de Kullback-Leibler :

$$\begin{aligned}D(q||p) &= E_p[\log\frac{p(X)}{q(X)}]\\ &= -H_p(X) + \lim_{n\to\infty} -\frac{1}{n}\log q(X_1,\ldots,X_n)\end{aligned}$$

Le premier terme, l'entropie de X sous p, est indépendant du choix de q. De façon à minimiser la divergence, il s'agit donc pour nous de maximiser la moyenne des log-vraisemblances des observations sous la distribution estimée. On considère parfois l'exponentielle de ce terme pour obtenir un ordre de grandeur d'un « nombre de valeurs possibles », par analogie avec la théorie de l'information.

3.3.2 Adaptation au cas du modèle de mélange de multinomiales

Comment calculer la vraisemblance de l'ensemble des observations sous le modèle choisi ? L'application de cette méthode est assez claire dans le cas du modèle unigramme

où il n'y a pas d'ambiguïté sur les unités i.i.d. de base à considérer : ce sont les mots. La solution est moins évidente dans le cas du modèle de mélange de multinomiales car les tirages de mots à l'intérieur d'un même document ne sont indépendants que conditionnellement au thème. Il faut donc travailler au niveau des documents, qui sont bien indépendants[1]. Ils ne sont en revanche pas tout à fait identiquement distribués puisque le nombre de mots tirés diffère d'un texte sur l'autre. En supposant que l'on modélise également la probabilité de la longueur l_d du document d, représenté comme un vecteur de mots C_d, la quantité à calculer est donc, puisque la longueur est indépendante du thème :

$$\begin{aligned}
-\frac{1}{n_D}\log q(C_1,\ldots,C_{n_D}) &= -\frac{1}{n_D}\sum_{d=1}^{n_D}\log q(C_d) \\
&= -\frac{1}{n_D}\sum_{d=1}^{n_D}\left(\log q(l_d)+\log\left(\sum_{t=1}^{n_T} q(t)q(C_d|l_d,t)\right)\right) \\
&= -\frac{1}{n_D}\sum_{d=1}^{n_D}\log q(l_d) \\
&\quad -\frac{1}{n_D}\sum_{d=1}^{n_D}\log\left(\sum_{t=1}^{n_T}\alpha_t\frac{l_d!}{\prod_{w=1}^{n_W}C_{wd}!}\prod_{w=1}^{n_W}\beta_{tw}^{C_{wd}}\right) \\
&= -\frac{1}{n_D}\sum_{d=1}^{n_D}\log\left(q(l_d)\frac{l_d!}{\prod_{w=1}^{n_W}C_{wd}!}\right) \\
&\quad -\frac{1}{n_D}\sum_{d=1}^{n_D}\log\left(\sum_{t=1}^{n_T}\alpha_t\prod_{w=1}^{n_W}\beta_{tw}^{C_{wd}}\right)
\end{aligned}$$

Nous nous intéressons uniquement au second terme puisque le premier concerne la longueur et est indépendant des choix de α et β. Ces calculs étant effectués au niveau des documents, l'opération de diviser par la longueur moyenne ($\frac{l}{n_D}$) permet de se ramener au niveau des mots. Nous prenons ensuite l'exponentielle pour obtenir un nombre de choix potentiels, représentant l'entropie moyenne d'un mot. La formule résultante permettant de calculer la perplexité est valable aussi bien sur l'ensemble d'apprentissage que sur l'ensemble de test.

$$\mathcal{P} = \exp\left(-\frac{1}{l}\sum_{d=1}^{n_D}\log\left(\sum_{t=1}^{n_T}\alpha_t\prod_{w=1}^{n_W}\beta_{tw}^{C_{wd}}\right)\right) \tag{3.1}$$

Notons que le calcul de cette quantité est assez immédiat après l'application de l'algorithme EM, puisqu'elle ne met en jeu que les log-probabilités a posteriori, déjà évaluées par ailleurs. Sur l'ensemble d'apprentissage en particulier, l'algorithme EM conduit à calculer la log-vraisemblance (essentiellement à des fins de vérification) dont la perplexité se déduit presque immédiatement (en normalisant, en changeant de signe et en prenant l'exponentielle). D'une façon générale dans les modèles probabilistes, la log-vraisemblance et la perplexité désignent à peu de choses près la même quantité. Dans ce qui suit, nous considérerons donc plutôt la perplexité, pour pouvoir comparer avec des modèles plus simples, tel le modèle unigramme.

[1]Notons au passage qu'on procéderait de même pour évaluer la perplexité de LDA. Pour PLSA, en revanche, l'indice muet d présent dans les paramètres n'a pas de sens en dehors des textes déjà vus. Hofmann ne donne pas d'indication sur la façon dont il détermine la perplexité mais [Blei et al., 2002], qui reprend cette lacune comme une critique de PLSA, propose dans ce cas d'évaluer la probabilité d'un mot en sommant sur tous les thèmes et documents possibles.

Même en supposant que l'on sache parfaitement estimer les quantités qui interviennent dans les formules, les mesures de perplexité ne sont pas exemptes de reproches. Elles ne capturent qu'un aspect de l'adéquation aux données et adapter le modèle aux documents à l'excès peut conduire à du sur-apprentissage[2] Elles ne sont donc pas forcément toujours corrélées avec la notion intuitive de « meilleur » modèle. Par exemple, LDA a un niveau de complexité supplémentaire par rapport à PLSA et c'est donc sans surprise qu'il permet d'obtenir une meilleure perplexité mais cela n'est pas pour autant une preuve qu'il sera une meilleure aide à l'analyse exploratoire.

3.3.3 Approche « Leave-one-out »

Un autre point de vue, peut-être plus pertinent vis-à-vis du but final de l'algorithme, est de considérer la capacité de prédiction sur les mots. Mais comme ils ne sont pas indépendants, il faut formuler cette fois la mesure en termes d'entropie conditionnelle. On veut en effet savoir si la connaissance des autres mots du texte apporte plus de certitude sur un mot donné. Il s'agit d'une approche de type « leave-one-out ». Si l'on numérote les occurrences par leur ordre d'apparition dans le corpus, on note W_i et D_i les indices de mot et de document correspondant à l'occurrence numéro $i \in \{1, \ldots, l\}$ (le « sac de mot » associé est donc C_{D_i}) et $C_{D_i}^{-W_i}$ le document C_{D_i} sans le mot W_i[3].

$$
\begin{aligned}
H(W|C_D^{-W}) &= E_{\mathrm{P}}[-\log \mathrm{P}(W|C_D^{-W})] \\
&= \lim_{l\to\infty} -\frac{1}{l}\sum_{i=1}^{l} \log \mathrm{P}(W_i|C_{D_i}^{-W_i}) \\
&= \lim_{l\to\infty} -\frac{1}{l}\sum_{i=1}^{l} \log \left(\frac{\mathrm{P}(C_{D_i})}{\mathrm{P}(C_{D_i}^{-W_i})}\right) \\
&= \lim_{l\to\infty} -\frac{1}{l}\sum_{i=1}^{l} \log \left(\frac{\frac{l_{D_i}!}{\prod_{w=1}^{n_W} C_{wD_i}!}\sum_{t=1}^{n_T}\alpha_t \prod_{w=1}^{n_W}\beta_{tw}^{C_{wD_i}}}{\frac{(l_{D_i}-1)!}{\prod_{w=1}^{n_W} C_{wD_i}^{-W_i}!}\sum_{t=1}^{n_T}\alpha_t \prod_{w=1}^{n_W}\beta_{tw}^{C_{wD_i}}/\beta_{tW_i}}\right) \\
&= \lim_{l\to\infty} \frac{1}{l}\sum_{i=1}^{l} \log \left(\frac{C_{W_iD_i}\sum_{t=1}^{n_T}\alpha_t \prod_{w=1}^{n_W}\beta_{tw}^{C_{wD_i}}/\beta_{tW_i}}{l_{D_i}\sum_{t=1}^{n_T}\alpha_t \prod_{w=1}^{n_W}\beta_{tw}^{C_{wD_i}}}\right) \\
&= \lim_{l\to\infty} \frac{1}{l}\sum_{i=1}^{l} \log \left(\frac{C_{W_iD_i}}{l_{D_i}}\sum_{t=1}^{n_T}\frac{\alpha_t \prod_{w=1}^{n_W}\beta_{tw}^{C_{wD_i}}}{\sum_{t'=1}^{n_T}\alpha_{t'}\prod_{w=1}^{n_W}\beta_{t'w}^{C_{wD_i}}}\frac{1}{\beta_{tW_i}}\right) \\
&= \lim_{l\to\infty} \left(\frac{1}{l}\sum_{i=1}^{l} \log \sum_{t=1}^{n_T}\frac{\mathrm{P}(T_{D_i}=t)}{\beta_{tW_i}} + \frac{1}{l}\sum_{i=1}^{l}\log\frac{C_{W_iD_i}}{l_{D_i}}\right)
\end{aligned}
$$

Nous nous abstenons de calculer le deuxième terme, qui ne dépend pas des paramètres.

[2]Nous avons déjà évoqué ce problème de sur-apprentissage au chapitre 2 et suggéré un remède, le *lissage*. Ce principe de privilégier certains modèles a priori pour lutter contre le sur-apprentissage est utilisé dans bien des situations sous des noms et techniques différents : par exemple, principe d'*Occam Razor* en apprentissage statistique, usage de certaines lois a priori dans un cadre bayésien, régularisation en analyse, principe de *longueur minimale de description* (MDL) en théorie de l'information.

[3]On ne supprime ici qu'une seule occurrence du mot W_i. S'il apparaît plus d'une fois dans le texte, le compte est décrémenté d'une unité mais pas annulé.

En pratique, nous avons constaté que cette mesure a des variations très similaires à la perplexité $\mathcal{P}$ définie en (3.1). Ce phénomène s'explique facilement lorsque le partitionnement est presque déterministe, ce qui est notre cas (voir la section 4.2.4). Ainsi, si $\mathrm{P}(T_{D_i} = t) \approx \mathbb{1}_{\{T_{D_i}=t\}}$, c'est-à-dire si $\sum_{t=1}^{n_T} \alpha_t \prod_{w=1}^{n_W} \beta_{tw}^{C_{wD_i}} \approx \alpha_{T_{D_i}} \prod_{w=1}^{n_W} \beta_{T_{D_i}w}^{C_{wD_i}}$, $\mathcal{P}$ s'écrit :

$$\exp\left(-\frac{1}{l}\sum_{d=1}^{n_D}\left(\log\alpha_{T_d} + \sum_{w=1}^{n_W} C_{wd}\log\beta_{tw}\right)\right).$$

D'autre part, pour la perplexité « leave-one-out », nous obtenons l'approximation suivante :

$$\begin{aligned}\frac{1}{l}\sum_{i=1}^{l}\log\sum_{t=1}^{n_T}\frac{\mathrm{P}(T_{D_i}=t)}{\beta_{tW_i}} &\approx \frac{1}{l}\sum_{i=1}^{l}\log\frac{1}{\beta_{T_{D_i}W_i}} \\ &\approx -\frac{1}{l}\sum_{d=1}^{n_D}\sum_{w=1}^{n_W} C_{wd}\log\beta_{T_d w}\end{aligned}$$

Aux termes en $\log\alpha_t$, qui sont presque systématiquement négligeables dans la somme, et à la fonction exponentielle près, il s'agit d'une expression équivalente à $\mathcal{P}$. En conclusion, même si l'idée de la perplexité « leave-one-out » semble plus naturelle, le fait qu'elle donne quasiment les mêmes résultats que la perplexité (3.1) dans nos expériences nous a conduit à ne pas la conserver dans le cadre d'évaluation.

3.4 Évaluation extrinsèque

Devant les difficultés à trouver des critères objectifs d'évaluation de la classification non supervisée, il est fréquent de recourir à des mesures dites d'évaluation extrinsèque [Galliers and Jones, 1995], c'est-à-dire consistant à tester la performance du modèle par rapport à une tâche applicative spécifique donnée et sur laquelle on sait discriminer les mauvaises réponses des bonnes (par opposition à l'évaluation intrinsèque qui veut déterminer la qualité d'un modèle dans l'absolu).

C'est le choix par exemple de [Deerwester et al., 1990], qui utilise pour l'évaluation de LSA un corpus classique de recherche d'information, avec requêtes et documents pertinents associés. Dans ce cadre, la singularisation et l'approximation de la matrice des comptes sont des préalables à des tâches d'indexation en général et de réponse à une recherche en particulier. La requête est assimilée à un document très court et projetée dans l'espace latent pour retrouver les textes du corpus qui en sont le plus proche. De même, [Azzopardi et al., 2003] propose une étude intéressante sur PLSI en tant que modèle de langage, en préalable à une tâche de recherche d'information. Les auteurs constatent une corrélation empirique entre les performances en termes de perplexité pour évaluer le modèle de langage et la précision sur l'objectif final de recherche d'information. Une expérience analogue à propos des questions de liens entre différents « degrés » d'évaluation est décrite à la fin de l'article [Hofmann, 2001]. Les résultats montrent une corrélation importante entre perplexité et précision (à un niveau fixé de rappel). Enfin, une version encore plus évoluée de ce mode d'évaluation extrinsèque peut consister à développer une chaîne de traitement complexe s'appuyant sur un modèle de classification non supervisée comme élément de base. Par exemple, [Buntine et al., 2005] construit autour de LDA une version thématique des algorithmes de classement (*ranking*) utilisés classiquement par les moteurs de recherche.

Néanmoins, d'une façon générale, cette approche de parallèle avec une tâche supervisée ne nous satisfait pas totalement car il semble que l'analyse exploratoire et la recherche d'information, bien que voisines, ne soient pas totalement analogues. Une de nos contraintes est en effet que le plan de classement soit intelligible pour un observateur humain alors qu'un algorithme de recherche d'information n'est en aucun cas soumis à un tel impératif.

Citons dans la même catégorie la tâche de filtrage collaboratif [Blei et al., 2002] qui consiste à trouver les objets que préfère un utilisateur à partir de ses choix sur un ensemble d'entraînement. De même que pour la recherche d'information, il n'est pas nécessaire pour obtenir un bon filtre collaboratif d'établir un plan de classement complet.

Enfin, le fait que la classification supervisée de documents soit un domaine proche de l'analyse exploratoire, mais en même temps beaucoup mieux cerné, en fait une cible privilégiée pour l'évaluation extrinsèque. Il est possible, par exemple, de faire des sous-classes pour améliorer la performance de catégorisation [Vinot and Yvon, 2003] ou, pour les méthodes symétriques, de tester l'apport d'une étape de classification non supervisée sur les mots [Slonim and Tishby, 2001] ou de générer de nouveaux attributs thématiques [Buntine and Jakulin, 2006]. Un exemple pratique de ce type d'expériences est notre participation au défi fouilles de textes (DEFT) 2005, décrite en annexe B.

3.5 Mesures de comparaison avec un étiquetage manuel

3.5.1 Principe et notations

Pour autant, toutes les informations d'étiquetage ne sont pas inutiles. Ainsi, si l'on dispose déjà d'une catégorisation des documents, qui induit de fait un partitionnement, il existe plusieurs mesures pour comparer ces deux classifications. Parmi elles, évoquons brièvement la *pureté* [Bisson, 2001], qui définit la qualité d'un thème comme ayant une composition la plus homogène possible par rapport à l'autre classement. Cette mesure, bien que raisonnable, répond correctement à la question *le classement par thèmes respecte-t-il les coupures induites par la partition initiale ?* mais pas à la suivante, également importante *le classement par thèmes n'introduit-il pas de sous-thèmes superflus ?*. Ces questions montrent bien qu'il y a deux aspects à évaluer, de façon un peu analogue à la définition de la précision et du rappel en recherche d'information.

Un problème fondamental lié à une comparaison entre deux classifications est que nous n'avons pas de moyen immédiat de trouver quelles catégories initiales correspondent à chaque thème du nouveau classement et inversement. Les indices de Rand et de Jaccard [Halkidi et al., 2001, Rooney et al., 2006], issus des statistiques classiques, contournent le problème en considérant les « accords » entre paires de documents, c'est-à-dire qu'ils évaluent pour chaque couple de documents s'ils sont dans la même partition dans le classement de référence et dans le classement proposé. La synthèse convenablement normalisée des « accords » sur l'ensemble des paires donne un score de similarité entre les deux partitions. Bien que ces indices mesurent bien la similarité en contournant par l'examen exhaustif des paires le problème de l'appariement thème-catégorie, ils considèrent dans leur version de base uniquement le cas de la classification déterministe. Dans la prochaine sous-section, nous montrons que la mesure d'information mutuelle, qui dispose de fondements en théorie de l'information, permet de gérer également le cas de la classification probabiliste. Dans la dernière sous-section, nous présenterons un algorithme d'appariement qui permet de considérer le problème un peu différemment, sans effectuer un examen exhaustif de toutes les configurations possibles : la méthode hongroise.

Dans ce qui suit, nous appelons toujours thèmes la classification provenant de l'algorithme à tester et la notons $\tau = \{\tau_1, \ldots, \tau_{n_T}\}$. Chaque texte $d \in \{1, \ldots, n_D\}$ appartient alors au thème τ_t avec une certaine probabilité $\mathrm{P}(\tau_t|C_d)$. Par ailleurs, la partition qui existe déjà sur le corpus et qui a en général été établie manuellement, les catégories, est notée de façon similaire $\Gamma = \{\Gamma_1, \ldots, \Gamma_{n_C}\}$[4].

3.5.2 Information mutuelle

La mesure la mieux justifiée en termes de théorie de l'information semble être la distance entre la distribution proposée par le modèle et l'information de catégories, vue comme une distribution idéale. Il s'agit de l'*information mutuelle*[5], utilisée notamment dans [Vinokourov and Girolami, 2002] et définie comme l'entropie de la distribution « idéale » Γ moins son entropie conditionnelle connaissant la distribution approchée τ.

$$IM = \sum_{c=1}^{n_C} \sum_{t=1}^{n_T} \mathrm{P}(\Gamma_c, \tau_t) \log \frac{\mathrm{P}(\Gamma_c, \tau_t)}{\mathrm{P}(\Gamma_c)\,\mathrm{P}(\tau_t)}$$

On estime les probabilités de thème et catégorie par échantillonnage sur le corpus :

$$\begin{aligned} \forall c \in \{1, \ldots, n_C\}, \widehat{\mathrm{P}(\Gamma_c)} &= \sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d)\,\mathrm{P}(C_d) = \frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d) \\ \forall t \in \{1, \ldots, n_T\}, \widehat{\mathrm{P}(\tau_t)} &= \frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(\tau_t|C_d) \end{aligned}$$

De même, la probabilité jointe est estimée par :

$$\begin{aligned} \forall c \in \{1, \ldots, n_C\}, \forall t \in \{1, \ldots, n_T\}, \widehat{\mathrm{P}(\Gamma_c, \tau_t)} &= \sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c, \tau_t|C_d)\,\mathrm{P}(C_d) \\ &= \frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c, \tau_t|C_d) \\ &= \frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d)\,\mathrm{P}(\tau_t|C_d) \end{aligned}$$

en supposant que catégorie et thème sont indépendants conditionnellement à la donnée du document. L'information mutuelle peut donc être estimée par :

$$\begin{aligned} \widehat{IM} = \sum_{c=1}^{n_C} \sum_{t=1}^{n_T} &\left(\frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d)\,\mathrm{P}(\tau_t|C_d) \right) \\ &\times \log n_D \frac{\sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d)\,\mathrm{P}(\tau_t|C_d)}{\left(\sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d)\right)\left(\sum_{d=1}^{n_D} \mathrm{P}(\tau_t|C_d)\right)} \end{aligned}$$

[4]Par souci de généralité, nous adoptons plutôt ici une notation probabiliste pour les thèmes (respectivement catégories) adaptée à une classification probabiliste. Cependant, il est possible que chaque texte appartienne à un et un seul thème (respectivement catégorie), surtout si les catégories sont issues de corpus de classification supervisée. Ce cas de la classification déterministe peut être vu comme une limite du précédent, toutes les distributions étant concentrées en un seul point. Les formules peuvent alors être avantageusement simplifiées avec des notations ensemblistes.

[5]L'information mutuelle est aussi couramment utilisée en classification supervisée pour les problèmes de sélection d'attributs [McCallum and Nigam, 1998].

Il est possible d'utiliser ce score de similarité tel quel mais il devient plus parlant si nous parvenons à le normaliser pour obtenir une mesure comprise entre 0 et 1. On peut utiliser pour cela les entropies des distributions. En notant H l'entropie, l'information mutuelle des deux variables aléatoires τ et Γ est en effet égale à $H(\tau)-H(\tau|\Gamma)$ et $H(\Gamma)-H(\Gamma|\tau)$: il s'agit de l'entropie d'une des deux variables moins son entropie conditionnelle connaissant l'autre (forcément inférieure). [Popescu-Belis, 2000] montre dans un autre cadre, celui d'évaluation de la résolution de la co-référence, qu'en normalisant par les entropies des distributions, on obtient deux scores $\frac{IM}{H(\tau)}$ et $\frac{IM}{H(\Gamma)}$ compris entre 0 et 1 et que l'on peut comparer à des mesures de précision et rappel sur une quantité intuitivement comparable à l'information apportée par la catégorisation originale. En effet, si $H(\Gamma|\tau) = 0$, cela signifie que Γ est totalement connue par la donnée de τ et donc que les thèmes ont retrouvé l'ensemble de l'information contenue dans les catégories. En recherche d'information, nous aurions alors un rappel de 1, ici, $\frac{IM}{H(\Gamma)} = \frac{H(\Gamma)}{H(\Gamma)} = 1$. On appelle parfois cette quantité le *gain d'information relatif*. De même, on peut définir la précision qui, en normalisant par $H(\tau)$ permettra de pénaliser les classifications qui établissent plus de groupes que nécessaire. Il est d'usage de résumer ces deux informations par leur moyenne géométrique, dite mesure du *F-score d'information mutuelle* :

$$\begin{aligned} IMFS &= \frac{2}{1/\text{IM Précision} + 1/\text{IM Rappel}} \\ &= \frac{2IM}{H(\tau) + H(\Gamma)} \end{aligned}$$

$H(\tau) = -\sum_{t=1}^{n_T} \mathrm{P}(\tau_t) \log \mathrm{P}(\tau_t)$ et $H(\Gamma) = -\sum_{c=1}^{n_C} \mathrm{P}(\Gamma_c) \log \mathrm{P}(\Gamma_c)$ sont estimées par échantillonnage sur le corpus, de même que précédemment. D'autres stratégies de normalisation sont possibles, c'est ainsi que [Xu et al., 2003] préconise de normaliser par $\max(H(\Gamma), H(\tau))$. Toutefois dans un contexte où les groupes sont à peu près équilibrés, les différentes normalisations donnent des résultats analogues.

L'information mutuelle présente donc de nombreuses propriétés intéressantes et intuitives pour comparer deux distributions probabilistes thématiques. L'étude [Meila, 2003], qui préconise une mesure très proche du F-Score : la *variation d'information*, définie par $H(\tau) + H(\Gamma) - 2IM$, démontre un certain nombre de ces avantages dans le cas de classifications déterministes, par exemple les propriétés de symétrie et d'inégalité triangulaire. Néanmoins, ces propriétés ne se transposent pas toutes au cas de partitionnements probabilistes. C'est dans ce contexte que nos expériences ont même révélé un comportement gênant dans certaines situations limites, notamment lorsque le partitionnement de référence consiste à rassembler presque tous les individus dans le même groupe. Ainsi, si nous supposons, dans un cas extrême, que tous les documents sont classés de la même façon, c'est-à-dire que $P(\Gamma|C_d)$ ne dépend pas de d, nous obtenons pour le terme en log de l'information mutuelle :

$$\begin{aligned} \log n_D \frac{\sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d)\,\mathrm{P}(\tau_t|C_d)}{\left(\sum_{d=1}^{n_D} \mathrm{P}(\Gamma_c|C_d)\right)\left(\sum_{d=1}^{n_D} \mathrm{P}(\tau_t|C_d)\right)} &= \log n_D \frac{\mathrm{P}(\Gamma_c|C_d) \sum_{d=1}^{n_D} \mathrm{P}(\tau_t|C_d)}{n_D\,\mathrm{P}(\Gamma_c|C_d)\left(\sum_{d=1}^{n_D} \mathrm{P}(\tau_t|C_d)\right)} \\ &= 0 \end{aligned}$$

Ce résultat montre que si le partitionnement probabiliste de référence est le même pour tous les documents, l'information mutuelle est nulle quelle que soit la classification proposée par l'utilisateur, même si c'est la même que celle de référence ! Nous voyons donc ici les limites des mesures fondées sur l'information mutuelle telles que nous souhaitons les utiliser

(pour comparer deux partitions). Il est vrai que ce cas un peu « pathologique » ne devrait pas se produire dans la réalité. Nous souhaitons en effet que la catégorisation de référence nous apporte réellement une information sur le corpus en séparant les documents, ce qui signifie qu'elle doit être en général plus déterministe que probabiliste. Néanmoins, nous avions choisi l'information mutuelle précisément pour sa capacité à prendre en compte les affectations probabilistes. S'il s'avère qu'elle ne présente pas un comportement convenable dans ce domaine, il paraît souhaitable de chercher une autre solution. Dans la section suivante, nous résolvons le problème de l'appariement thèmes-catégories explicitement, par une méthode issue de la théories des graphes.

3.5.3 Méthode hongroise

Dans ce qui précède, nous avons soulevé le problème de l'ambigüité de l'association thèmes-catégories et avons proposé des mesures qui permettaient de le contourner. Nous montrons dans cette section qu'il existe un moyen d'établir un appariement entre deux partitionnements. Si les deux regroupements ont le même nombre de thèmes, ceci peut être fait au moyen d'un algorithme baptisé *méthode hongroise* [Kuhn, 1955, Papadimitriou and Steiglitz, 1998, Frank, 2004], qui calcule le meilleur appariement pondéré dans un graphe bi-parti.

Nous supposons que $n_T = n_C$ et reformulons le problème de la façon suivante. Soit un graphe bi-parti contenant d'un côté un sommet pour chaque thème t et de l'autre un sommet par catégorie c. Chaque thème est connecté à chaque catégorie par une arête à laquelle on associe le poids : $d_{tc} = \sum_{d=1}^{n_D} \mathrm{P}(\tau_t|C_d)\,\mathrm{P}(\Gamma_c|C_d)$[6]. Nous cherchons l'appariement de poids maximal, c'est-à-dire l'ensemble $\{(t_1, c_1), \ldots, (t_{n_T}, c_{n_T})\}$ de n_T arêtes touchant une et une seule fois chaque thème et catégorie tel que $\sum_{i=1}^{n_T} d_{t_i c_i}$ soit maximal. On appelle sous-appariement un ensemble d'arêtes qui touche au maximum une fois chaque thème et chaque catégorie mais qui ne couvre pas forcément tous les sommets.

Le principe général est d'ajuster itérativement une *couverture* du graphe, c'est-à-dire un ensemble de poids associés à chaque thème $\sigma_\tau(t)$ et à chaque catégorie $\sigma_\Gamma(c)$ et tels que

$$\forall t, c \in \{1, \ldots, n_T\}, \sigma_\tau(t) + \sigma_\Gamma(c) \geq d_{tc}\,. \tag{3.2}$$

On dit qu'une arête est *saturée* lorsqu'il y a égalité dans l'équation (3.2). L'algorithme adapte progressivement les poids pour saturer le plus d'arêtes possibles, de telle façon que lorsqu'il existe un appariement dans l'ensemble des arêtes saturées, nous sommes sûrs qu'il s'agit d'un appariement de poids maximum.

L'initialisation affecte un poids nul à chaque catégorie c : $\sigma_\Gamma(c) = 0$ et le poids le plus élevé possible à chaque thème t : $\sigma_\tau(t) = \max_{c \in \{1,\ldots,n_T\}} d_{tc}$. Considérons alors l'ensemble des arêtes saturées (il y en a au moins une par thème) et déterminons, en sélectionnant parmi ces arêtes, le sous-appariement couvrant le plus grand nombre de thèmes :

- Nous partons d'un sous-appariement quelconque A (par exemple, en considérant les thèmes dans l'ordre, en retenant pour chacun, si elle existe, la première arête saturée conduisant à une catégorie non incluse dans A jusqu'alors).
- Nous orientons les arêtes saturées :
  - de Γ vers τ pour celles qui sont incluses dans A ;
  - de τ vers Γ pour les autres.

[6] En pratique, il peut être utile d'arrondir la quantité d_{tc} pour n'avoir à traiter que des poids entiers, car des propriétés de convergence sont démontrables dans ce cas [Frank, 2004]. Cette approximation est d'autant plus valide que le nombre de documents est grand par rapport au nombre de thèmes.

- Nous nous intéressons alors à l'ensemble des sommets (dans τ et dans Γ) que nous pouvons atteindre par le biais de chaînes orientées de longueur quelconque en partant des *thèmes* qui ne sont pas dans A. Notons Z cet ensemble[7]. Si Z contient une catégorie qui n'est pas dans A, il est possible d'obtenir un sous-appariement plus grand, en échangeant les orientations des arêtes de la chaîne alternante [Frank, 2004].
- Nous répétons l'opération tant qu'il existe de telles chaînes alternantes et le sous-appariement maximal final A est caractérisé par les arêtes orientées de Γ vers τ.

Si l'appariement maximal A obtenu est complet, l'algorithme s'arrête. Sinon notons :

$$\begin{aligned} Z_\tau &= Z \cap \tau \\ Z_\Gamma &= Z \cap \Gamma \\ \epsilon &= \min\{\sigma_\tau(t) + \sigma_\Gamma(c) - d_{tc}, t \in Z_\tau, c \in \Gamma \backslash Z_\Gamma\} \end{aligned}$$

Parmi les thèmes non encore couverts par A, nous cherchons donc ceux dont le poids peut être diminué. On ajuste enfin les poids de la façon suivante :

$$\begin{aligned} \forall t \in Z_\tau, \sigma_\tau(t) &\leftarrow \sigma_\tau(t) - \epsilon \\ \forall c \in Z_\Gamma, \sigma_\Gamma(c) &\leftarrow \sigma_\Gamma(c) + \epsilon \end{aligned}$$

Puis on reprend itérativement depuis la détermination de l'ensemble A, jusqu'à ce que ce dernier soit un appariement complet.

La complexité de cet algorithme est cubique par rapport au nombre de thèmes n_T. Nous ne détaillons pas plus ici cet algorithme, en particulier sur les preuves de convergence et de complexité. Nous renvoyons le lecteur intéressé à [Frank, 2004] pour la version que nous avons présentée (s'appuyant sur des graphes) et à [Papadimitriou and Steiglitz, 1998] pour un traitement algorithmique plus complet.

Une fois l'étape délicate d'appariement thème-catégorie établie, nous pouvons calculer le pourcentage de documents sur lesquels les deux partitionnements sont « d'accord », c'est-à-dire ceux dont le thème et la catégorie sont associés par la méthode hongroise. Ce score se calcule aisément après application de l'algorithme : il suffit de diviser le score d'appariement maximal par le nombre de documents. [Lange et al., 2004] décrit plus précisément comment cette méthode peut être exploitée dans le cas de la classification non supervisée.

Une limite d'une stratégie d'évaluation reposant sur la méthode hongroise est qu'elle n'est pas adaptée à la comparaison de deux classifications probabilisées avec différents nombres de classes. Un cas que la mesure sanctionne de façon particulièrement sévère est celui où une classe de la partition τ est divisée en deux classes dans la partition Γ (ou inversement), un inconvénient dont ne souffre pas l'information mutuelle. Il est toutefois possible d'adapter la méthode hongroise au cas déséquilibré $n_T \neq n_C$[8]. Une autre solution, proposée par [Zhou et al., 2005], consiste à adapter la *distance de Mallows* définie entre deux distributions de probabilités pour déterminer un appariement probabiliste optimal.

[7] Par construction, Z contient au moins les thèmes qui ne sont pas dans A et les catégories auxquelles ils sont directement connectés. De proche en proche, il est aussi susceptible de contenir d'autres thèmes et catégories qui ne sont pas directement liés aux thèmes de départ.

[8] Mais la définition de « meilleur appariement » devient alors non triviale en fonction des différents jeux de contraintes. Si nous supposons sans perte de généralité que $n_T > n_C$ et que nous choisissons de respecter la contrainte *chaque thème doit être associé à une et une seule catégorie* (l'inverse étant manifestement incompatible avec $n_T > n_C$), il est possible de montrer que la meilleure stratégie consiste à trouver les n_C meilleurs mariages par la méthode hongroise et à couvrir les $n_T - n_C$ thèmes restants en associant chacun à sa catégorie la plus proche.

Le problème est ensuite numériquement résolu par programmation linéaire. Bien que cette méthode nécessite de choisir un critère d'optimalité parmi les différentes distances de Mallows et d'appliquer un algorithme de complexité plus grande que la méthode hongroise, elle peut constituer une alternative à explorer dans les (rares) applications où l'appariement de la méthode hongroise ne semble pas satisfaisant.

Sur la base de l'appariement, nous pouvons également imaginer calculer d'autres mesures qui n'étaient pas robustes à une permutation des indices des thèmes, telles que la divergence de Kullback-Leibler entre τ et Γ. Néanmoins, comme nous le verrons en section 4.2.4, les classifications que nous étudierons en pratique sont suffisamment déterministes pour pouvoir se contenter du score de cooccurrences.

3.6 Discussion et méthodologie

Quelles que soient les justifications théoriques des mesures de comparaison avec un classement de référence, elles reposent sur une hypothèse implicite que les catégories constituent le résultat idéal vers lequel on souhaite tendre. C'est en général abusif car les catégories ne représentent qu'un classement possible parmi bien d'autres, plus fins ou plus grossiers, dont certains peuvent être tout aussi pertinents. Puisque nous sommes dans un cadre non supervisé, il est justement contestable de réduire la qualité d'un modèle à une telle mesure. Il paraîtrait donc plus logique d'évaluer la qualité de l'analyse exploratoire « à l'oeil nu »[9], en fonction des résultats qu'elle donne sur un corpus. Cela semble faisable ponctuellement pour un ensemble de textes donné avec une méthode donnée, mais répéter l'expérience sur des domaines suffisamment variés avec chacun des modèles étudiés est irréaliste. Les scores de cooccurrences retournés par la méthode hongroise constituent un compromis intéressant entre tester l'adéquation du modèle aux données (désirable théoriquement) et son efficacité pratique vis-à-vis de la tâche à remplir.

Dans tous les problèmes d'apprentissage statistique, il est fondamental de distinguer la performance sur corpus d'apprentissage de celle sur ensemble de test. La première montre comment l'algorithme s'*adapte* aux données, la seconde comment il *généralise* à des situations non vues. Le but final est d'optimiser ce dernier score, puisque dans les applications réelles de l'apprentissage, on s'intéresse plutôt à des données non vues auparavant. Une question importante est alors de savoir si l'optimisation d'un critère sur l'ensemble d'apprentissage est bénéfique pour la performance sur le corpus de test. Le phénomène de *sur-apprentissage* montre que ce n'est pas toujours le cas et nous nous attacherons donc à considérer les scores sur les deux ensembles dans les expériences à venir. Nous avons expliqué pour quelles raisons nous privilégierons dans la suite les scores de cooccurrence. Il sera en outre intéressant de mesurer conjointement la perplexité et d'étudier ses variations dans la mesure où elle évalue plus directement la qualité de l'inférence.

Les valeurs finales obtenues pour les mesures sont naturellement dépendantes des données. Une question naturelle est donc de se demander à quel point les résultats obtenus ont valeur de généralité. Une réponse possible consiste à calculer des marges d'erreur statistique sur les résultats. Nous avons délibérément occulté ce problème dans ce chapitre. Ce choix se justifie par l'usage que nous ferons du cadre d'évaluation : il s'agit plus ici de relever des tendances que de prendre des décisions définitives en faveur d'une méthode ou d'une autre. Dès lors, le calcul de la *signification statistique* (*statistical significance*) des résultats n'a pas grand intérêt pratique. Néanmoins, si nous souhaitions le mettre en

[9]Le problème sous-jacent qui se pose alors est celui d'interprétation des thèmes. Voir la section 5.3.

place à d'autres fins, il serait certainement possible d'adapter les tests proposés pour la recherche d'information dans l'article [Hull, 1993] ou dans l'étude plus complète [Goutte and Gaussier, 2005].

D'autres domaines du TAL montrent que la recherche d'une métrique idéale est probablement un objectif inacessible. Ainsi, des efforts considérables ont été investis pour trouver une procédure d'évaluation automatique de référence, en résolution de la coréférence [Popescu-Belis, 2000], tout comme en traduction automatique [Goutte, 2006]. La conclusion est identique dans les deux cas : aucune des différentes mesures proposées ne se détache naturellement car elles ont toutes leurs avantages et leurs inconvénients propres, chacune rendant compte de différents aspects des performances et apportant par là-même un élément intéressant de réflexion. Comme nous l'avons vu dans ce chapitre, la classification non supervisée ne fait pas exception à la règle dans la multiplication des mesures proposées. Toutefois, le domaine semble un peu moins avancé dans la sélection des mesures de référence et l'on a le sentiment que certaines mesures, telles que l'information mutuelle et le score de cooccurrence par la méthode hongroise, sont redondantes. C'est pourquoi nous avons décidé de n'en conserver que deux, perplexité et score de cooccurrence, un parti-pris que nous avons justifié tout au long de ce chapitre.

Chapitre 4

Modèle de mélange de multinomiales

Nous présentons dans ce chapitre une étude menée sur le modèle de mélange de multinomiales vu en section 2.3.2.

Nous commençons par présenter le modèle plus en détail que dans le chapitre 2, en montrant notamment comment il est possible de dériver dans un cadre bayésien des formules intégrées pour les probabilités thématiques conditionnelles. Nous exposons à titre d'exemple une application à l'apprentissage supervisé dont l'utilité pratique sera étudiée dans les sections 4.4 (pour le cadre non supervisé) et 4.6.2 (pour le cadre supervisé).

La section 4.1, qui détaille les premières expériences sur le modèle de base, a la double vocation d'illustrer en pratique la façon dont nous utilisons le cadre d'évaluation et de mentionner des résultats préliminaires utiles dans la suite. Ainsi, nous proposons dans les sections suivantes des stratégies répondant aux lacunes de l'algorithme EM mises en évidence en section 4.2 et probablement dues à la présence de très nombreux optimums sur l'espace de grande dimension qu'est le vocabulaire. Une première idée, née de nos expériences sur le lissage, est la réduction de la dimensionnalité. Nous l'étudions en détail en section 4.3 et montrons qu'elle peut donner lieu à une méthode d'inférence heuristique compétitive.

Nous comparons ensuite ces résultats à une stratégie fondée sur les chaînes de Markov Monte Carlo (MCMC) et plus précisément l'échantillonnage de Gibbs. Enfin, nous concluons par deux notes sur les cadres semi-supervisé et supervisé, la première pour évaluer le nombre d'étiquettes nécessaires pour améliorer l'inférence avec l'algorithme EM, initialisation Dirichlet, et la seconde pour revenir sur les questions posées en fin de section 4.1.4 relatives au classifieur bayésien naïf.

4.1 Retour sur le modèle

Dans cette section, nous reprenons la présentation du modèle de mélange de multinomiales. À la différence de la section 2.3.2, nous allons à présent le développer dans un cadre totalement bayésien, avec des a prioris Dirichlet sur les paramètres.

La suite de cette section est constituée de développements théoriques sur le modèle. Nous montrons en section 4.1.3 en quoi le modèle est lié au classifieur bayésien naïf, puis expliquons en section 4.1.4 que certaines densités conditionnelles d'intérêt suivent une autre loi, appelée *Dirichlet-Multinomial* et en quoi cela est utile aussi bien en classification supervisée que non supervisée.

4.1.1 Approche bayésienne et lois conjuguées

Dans une approche bayésienne, les paramètres du modèle sont eux-mêmes considérés comme des variables aléatoires. En l'absence de toute observation, ils sont supposés suivre une loi dite *a priori*. Les observations modifient notre connaissance des paramètres et nous pouvons calculer leur nouvelle loi dite *a posteriori* [Robert, 2006].

En l'absence d'informations précises sur la distribution a priori, il est d'usage de choisir une loi *conjuguée* avec la loi des observations. On dit qu'une loi a priori est conjuguée avec la loi des observations si la loi a posteriori des paramètres conditionnellement aux observations a la même forme que la loi a priori. En d'autres termes, les observations ne modifient pas radicalement nos hypothèses sur la forme de la distribution du paramètre, elles ne font que modifier ses hyperparamètres. Le principal attrait des lois conjuguées est calculatoire : elles permettent d'obtenir des formules explicites pour la densité de la distribution a posteriori et, par conséquent, facilitent le processus d'inférence. Toutefois, les simplifications permises par ces distributions conjuguées ne doivent pas servir de prétexte au choix de distributions manifestement inadaptées à nos connaissances a priori. Dans notre cas, la loi de Dirichlet, conjuguée de la loi multinomiale, est en pratique suffisamment flexible pour représenter correctement un certain nombre de configurations concernant des probabilités discrètes.

La loi de Dirichlet définit des distributions de probabilité sur le simplexe. Chaque observation multi-dimensionnelle $\alpha = (\alpha_1, \ldots, \alpha_{n_T})$ vérifie donc : $\sum_{t=1}^{n_T} \alpha_t = 1$. En dimension n_T, cette loi est paramétrisée par un vecteur de n_T paramètres $p = (\lambda_1, \ldots, \lambda_{n_T})$. La probabilité d'une observation α est :

$$p(\alpha|\lambda) = \frac{\Gamma(\sum_{t=1}^{n_T} \lambda_t)}{\prod_{t=1}^{n_T} \Gamma(\lambda_t)} \prod_{t=1}^{n_T} \alpha_t^{\lambda_t - 1},$$

où Γ dénote la fonction Gamma d'Euler. Lorsque tous les λ_t sont égaux à λ_α, cette expression se simplifie en :

$$p(\alpha|\lambda_\alpha) = \frac{\Gamma(n_T \lambda_\alpha)}{\Gamma(\lambda_\alpha)^{n_T}} \prod_{t=1}^{n_T} \alpha_t^{\lambda_\alpha - 1}$$

Les distributions Dirichlet et multinomiale sont conjuguées. Plus précisément, supposons que le vecteur α suive une loi a priori Dirichlet, de paramètre $(\lambda_1, \ldots, \lambda_{n_T})$, et que l'on effectue l tirages selon une multinomiale de paramètre $(\alpha_1, \ldots, \alpha_{n_T})$, K_t étant au final le nombre d'obtention du résultat t, avec $t \in \{1, \ldots, n_T\}$ (naturellement, $\sum_{t=1}^{n_T} K_t = l$). La distribution a posteriori de α conditionnellement aux observations K est Dirichlet de paramètre $(K_1 + \lambda_1, \ldots, K_{n_T} + \lambda_{n_T})$. Nous percevons alors mieux la signification des hyperparamètres (qui sont parfois interprétés, dans un cadre bayésien, comme des observations *virtuelles* [Robert, 2006]). Chaque paramètre est à comparer avec le nombre d'observations recueilli dans sa composante et l'intensité $\sum_{t=1}^{n_T} \lambda_t$ est en « concurrence » avec le nombre total d'observations l. Plus cette intensité est importante, plus l'influence de l'a priori mettra de temps à s'effacer devant le nombre d'observations.

4.1.2 Présentation bayésienne du modèle

Ici, nous supposons que les lois a priori des paramètres α et $\beta_{t\bullet}$[1] sont des lois de Dirichlet équilibrées (c'est-à-dire que tous leurs paramètres sont égaux) d'hyperparamètres respectifs $\lambda_\alpha > 0$ et $\lambda_\beta > 0$.

Le modèle génératif pour le corpus entier $C = (C_1, \ldots, C_{n_D})$ devient en conséquence :

[1] Nous désignons ainsi la colonne de la matrice β relative au thème t.

1. $\alpha \sim \mathrm{Dir}(\lambda_\alpha, \ldots, \lambda_\alpha)$
2. Pour chaque thème $t = 1, \ldots, n_T$:
   $\beta_{t\bullet} \sim \mathrm{Dir}(\lambda_\beta, \ldots, \lambda_\beta)$
3. Conditionnellement à α, β, pour chaque $d = 1, \ldots, n_D$:
   (a) tirer un thème $T_d \sim \mathrm{Mult}(1, (\alpha_1, \alpha_2, \ldots, \alpha_{n_T}))$
   (b) conditionnellement à T_d, tirer l_d mots $C_d \sim \mathrm{Mult}(l_d, (\beta_{T_d 1}, \beta_{T_d 2}, \ldots, \beta_{T_d n_W}))$

La loi a posteriori s'obtient de façon classique (en omettant les termes indépendants de α et β) :

$$\begin{aligned} p(\alpha, \beta | C) &\propto \mathrm{P}(C|\alpha, \beta) p(\alpha) p(\beta) \\ &\propto \left(\prod_{d=1}^{n_D} \sum_{t=1}^{n_T} \alpha_t \prod_{w=1}^{n_W} \beta_{tw}^{C_{wd}} \right) \prod_{t=1}^{n_T} \alpha_t^{\lambda_\alpha - 1} \prod_{t=1}^{n_T} \prod_{w=1}^{n_W} \beta_{tw}^{\lambda_\beta - 1} \end{aligned} \tag{4.1}$$

Cette expression n'est pas maximisable explicitement. Il est néanmoins possible d'obtenir un *maximum a posteriori* local, via l'algorithme EM, de la même façon qu'en section 2.3.2 :

$$\mathrm{P}(T_d = t | C; \alpha', \beta') = \frac{\alpha'_t \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{tw}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \prod_{w=1}^{n_W} \beta'^{C_{wd}}_{t'w}} \tag{4.2}$$

$$\alpha_t \propto \lambda_\alpha - 1 + \sum_{d=1}^{n_D} \mathrm{P}(T_d = t | C; \alpha', \beta') \tag{4.3}$$

$$\beta_{tw} \propto \lambda_\beta - 1 + \sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T_d = t | C; \alpha', \beta') \tag{4.4}$$

et les facteurs de normalisation sont déterminés par les contraintes :

$$\begin{cases} \sum_{t=1}^{n_T} \alpha_t = 1 \\ \sum_{w=1}^{n_W} \beta_{tw} = 1 \quad \text{pour } t \in \{1, \ldots, n_T\}. \end{cases}$$

En outre, nous allons voir dans la section suivante que si les variables latentes de thèmes sont connues de façon déterministe (cadre supervisé), une maximisation explicite de (4.1) est possible.

4.1.3 Le classifieur bayésien naïf

Lorsque le vecteur de thèmes $T = (T_1, \ldots, T_{n_D})$ est connu pour l'ensemble d'apprentissage, nous pouvons définir les quantités suivantes : pour tous $t \in \{1, \ldots, n_T\}, w \in \{1, \ldots, n_W\}$, notons S_t le nombre de documents de l'ensemble d'apprentissage qui appartiennent au thème t et K_{wt} le nombre d'occurrences du mot w dans le thème t.

La recherche d'estimateurs du maximum a posteriori donne :

$$\begin{aligned} \log p(\alpha, \beta | C, T) &= \sum_{d=1}^{n_D} \log \mathrm{P}(C_d | \alpha, \beta, T_d) + \log p(\alpha) + \log p(\beta) + k \\ &= \sum_{d=1}^{n_D} \left(\log \alpha_{T_d} + \sum_{w=1}^{n_W} C_{wd} \log \beta_{T_d w} \right) \\ &\qquad + \sum_{t=1}^{n_T} \left((\lambda_\alpha - 1) \log \alpha_t + \sum_{w=1}^{n_W} (\lambda_\beta - 1) \log \beta_{tw} \right) + k' \\ &= \sum_{t=1}^{n_T} \left((S_t + \lambda_\alpha - 1) \log \alpha_t + \sum_{w=1}^{n_W} (K_{wt} + \lambda_\beta - 1) \log \beta_{tw} \right) + k' \end{aligned}$$

où k et k' sont des constantes indépendantes des paramètres. Ces termes n'ayant aucune influence sur la recherche des α et β maximisant $p(\alpha, \beta|C, T)$, nous pouvons les ignorer dans ce qui suit.

Prenons en compte les contraintes $\sum_{t=1}^{n_T} \alpha_t = 1$ et $\forall t \in \{1, \ldots, n_T\}, \sum_{w=1}^{n_W} \beta_{tw} = 1$ par l'introduction de multiplicateurs de Lagrange. Une résolution classique donne les solutions :

$$\alpha_t = \frac{S_t + \lambda_\alpha - 1}{n_D + n_T(\lambda_\alpha - 1)} \qquad \beta_{tw} = \frac{K_{wt} + \lambda_\beta - 1}{K_t + n_W(\lambda_\beta - 1)}$$

avec $K_t = \sum_{w=1}^{n_W} K_{wt}$ nombre d'occurrences dans le thème t.

Cela nous permet de déterminer des valeurs $\hat{\alpha}$ et $\hat{\beta}$ pour les paramètres et il est alors facile de calculer le thème le plus probable pour un nouveau document (non vu dans l'ensemble d'apprentissage) $d^\star$:

$$\begin{aligned}
t_{d^\star} &= \underset{t \in \{1,\ldots,n_T\}}{\arg\max}\ \mathrm{P}(T_{d^\star} = t | C_{d^\star}, \hat{\alpha}, \hat{\beta}) \\
&= \underset{t \in \{1,\ldots,n_T\}}{\arg\max}\ \mathrm{P}(T_{d^\star} = t | \hat{\alpha})\, \mathrm{P}(C_{d^\star} | T_{d^\star} = t, \hat{\beta}) \\
&= \underset{t \in \{1,\ldots,n_T\}}{\arg\max}\ \hat{\alpha}_t \prod_{w=1}^{n_W} \hat{\beta}_{tw}^{C_{wd^\star}} \\
&= \underset{t \in \{1,\ldots,n_T\}}{\arg\max}\ (S_t + \lambda_\alpha - 1) \frac{\prod_{w=1}^{n_W} (K_{wt} + \lambda_\beta - 1)^{C_{wd^\star}}}{(K_t + n_W(\lambda_\beta - 1))^{l_{d^\star}}}
\end{aligned} \tag{4.5}$$

À nouveau, nous avons ignoré les constantes indépendantes de t qui ne sont pas liées à la maximisation. Il suffit finalement de calculer (4.5) (ou son logarithme) pour obtenir le thème prédit par le classifieur. Nous retrouvons de cette façon la célèbre formule du classifieur bayésien naïf [Lewis, 1998, McCallum and Nigam, 1998].

4.1.4 La distribution Dirichlet-Multinomiale

Dans le cas général non supervisé, les thèmes des documents d'apprentissage sont inconnus. Mais nous allons maintenant voir que, même dans ce cas, il est possible d'obtenir une expression exploitable de la loi jointe de C et T, les paramètres α et β étant marginalisés. L'observation importante ici est que la distribution prédictive bayésienne d'un vecteur de comptes étant donnés les autres et leur variable indicatrice de thème peut être déterminée explicitement et est connue sous le nom de distribution Dirichlet-Multinomial [Mosimann, 1962].

Commençons par considérer la densité jointe de C et T :

$$\begin{aligned}
\mathrm{P}(C, T, \alpha, \beta) &\propto \underbrace{\prod_{d=1}^{n_D} \prod_{w=1}^{n_W} \beta_{T_d w}^{C_{wd}}}_{\mathrm{P}(C|T,\beta)} \underbrace{\prod_{d=1}^{n_D} \alpha_{T_d}}_{\mathrm{P}(T|\alpha)} \underbrace{\prod_{t=1}^{n_T} \alpha_t^{\lambda_\alpha - 1}}_{p(\alpha)} \underbrace{\prod_{t=1}^{n_T} \prod_{w=1}^{n_W} \beta_{tw}^{\lambda_\beta - 1}}_{p(\beta)} \\
&\propto \prod_{t=1}^{n_T} \left(\alpha_t^{S_t + \lambda_\alpha - 1} \prod_{w=1}^{n_W} \beta_{tw}^{K_{wt} + \lambda_\beta - 1} \right)
\end{aligned}$$

Il est possible d'intégrer cette expression par rapport à α et β sur les simplexes corres-

pondants. Ce calcul fait intervenir la constante de normalisation de la loi Dirichlet :

$$\begin{aligned} \mathrm{P}(T|C) &\propto \frac{\prod_{t=1}^{n_T}\Gamma(S_t+\lambda_\alpha)}{\Gamma\left[\sum_{t=1}^{n_T}(S_t+\lambda_\alpha)\right]}\prod_{t=1}^{n_T}\frac{\prod_{w=1}^{n_W}\Gamma(K_{wt}+\lambda_\beta)}{\Gamma\left[\sum_{w=1}^{n_W}(K_{wt}+\lambda_\beta)\right]} \\ &\propto \prod_{t=1}^{n_T}\left(\Gamma(S_t+\lambda_\alpha)\frac{\prod_{w=1}^{n_W}\Gamma(K_{wt}+\lambda_\beta)}{\Gamma\left[\sum_{w=1}^{n_W}(K_{wt}+\lambda_\beta)\right]}\right) \end{aligned}$$

À partir de ce résultat, il est possible de scinder le vecteur d'indicatrices de thème suivant ses différentes composantes T_d, et d'estimer la distribution conditionnelle a posteriori d'un document d d'appartenir à un thème, connaissant les thèmes des autres documents. Nous noterons $\mathrm{P}(T_d|T_{-d},C)$ cette densité conditionnelle, où T_{-d} désigne le vecteur d'indicatrices de thème pour tous les documents sauf le d-ième. De façon similaire, notons C_{-d} le corpus sans le document d et K_{wt}^{-d} tel que $K_{wt}^{-d}=\sum_{\{d'\neq d:T_d=t\}}C_{wd}$. Alors :

$$\begin{aligned} \mathrm{P}(T_d=t|T_{-d},C) &= \frac{\mathrm{P}(T_d=t,T_{-d}|C)}{P(T_{-d}|C_d,C_{-d})} \\ &= \frac{\mathrm{P}(T_d=t,T_{-d}|C)}{P(T_{-d}|C_{-d})}\frac{1}{\frac{P(C_d|T_{-d},C_{-d})}{P(C_d|C_{-d})}} \end{aligned}$$

Le second facteur ne dépend pas du thème t du document d. Donc $\mathrm{P}(T_d=t|T_{-d},C)$ est proportionnel à $\frac{\mathrm{P}(T_d=t,T_{-d}|C)}{P(T_{-d}|C_{-d})}$, c'est-à-dire à :

$$\frac{\prod_{t'=1}^{n_T}\left(\Gamma(S_{t'}+\lambda_\alpha)\frac{\prod_{w=1}^{n_W}\Gamma(K_{wt'}+\lambda_\beta)}{\Gamma\left[\sum_{w=1}^{n_W}(K_{wt'}+\lambda_\beta)\right]}\right)}{\left(\prod_{t'\neq t}\left(\Gamma(S_{t'}+\lambda_\alpha)\frac{\prod_{w=1}^{n_W}\Gamma(K_{wt'}+\lambda_\beta)}{\Gamma\left[\sum_{w=1}^{n_W}(K_{wt'}+\lambda_\beta)\right]}\right)\right)\times\Gamma(S_t-1+\lambda_\alpha)\frac{\prod_{w=1}^{n_W}\Gamma(K_{wt}^{-d}+\lambda_\beta)}{\Gamma\left[\sum_{w=1}^{n_W}(K_{wt}^{-d}+\lambda_\beta)\right]}},$$

dans la mesure où nous examinons la même quantité au numérateur et au dénominateur, au changement thématique du document d près. Dans ces conditions, la plupart des facteurs se simplifient et il reste l'expression suivante :

$$\mathrm{P}(T_d=t|T_{-d},C)\propto(S_t-1+\lambda_\alpha)\frac{\prod_{w=1}^{n_W}\Gamma(K_{wt}+\lambda_\beta)}{\prod_{w=1}^{n_W}\Gamma(K_{wt}^{-d}+\lambda_\beta)}\frac{\Gamma\left[\sum_{w=1}^{n_W}(K_{wt}^{-d}+\lambda_\beta)\right]}{\Gamma\left[\sum_{w=1}^{n_W}(K_{wt}+\lambda_\beta)\right]} \qquad (4.6)$$

Nous verrons dans la suite que cette formule peut être utile à au moins deux titres :

- Dans un cadre non supervisé, la connaissance des distributions conditionnelles de tous les T_d suggère une façon d'appliquer l'échantillonnage de Gibbs en les simulant les uns après les autres. Cette idée sera développée dans la section 4.4.3.
- Dans un cadre supervisé, la formule (4.6) correspond à la *distribution bayésienne predictive* du thème T_d du document C_d en supposant les autres documents C_{-d} et leurs étiquettes T_{-d} connus (ensemble d'apprentissage). Cette approche sera détaillée et comparée au classifieur bayésien naïf en section 4.6.2.

Enfin, nous remarquons de façon purement pragmatique pour l'implémentation que le calcul de l'équation (4.6) n'a pas d'incidence sur le temps d'exécution de l'algorithme dans la mesure où la fonction Γ (ou plutôt son logarithme) n'est jamais calculée qu'en des points de la forme $n+\lambda_\beta$ ou $n+n_W\lambda_\beta$, avec n entier et peut donc faire l'objet d'une tabulation préalable adéquate.

4.2 Performances de l'algorithme EM

Dans cette section, nous présentons les résultats d'une première série d'expériences avec l'algorithme EM. Après avoir présenté le corpus utilisé, nous évoquons les difficultés liées à la stratégie d'initialisation avant d'étudier l'influence empirique des paramètres de lissage.

4.2.1 Cadre expérimental

Pour les expériences de cette thèse, nous avons utilisé principalement un corpus, que nous avons choisi relativement simple (catégories faiblement recouvrantes) mais en même temps d'une taille suffisante (5000 textes) pour commencer à observer des effets liés à la dimensionnalité. Il s'agit d'un sous-ensemble du corpus Reuters 2000[2] [Reuters, 2000], lequel a été par exemple exploité dans [Rooney et al., 2006, Manning et al., 2007]. Nous avons sélectionné 5000 textes en veillant à ce que chacun d'eux ait au moins une catégorie parmi les cinq suivantes : emploi, sports, santé, culture ou accidents. Ces classes ont été choisies parce qu'elles étaient suffisamment fréquentes dans le corpus et qu'elles nous semblaient peu ambigües. Mais la pratique montre que la classification non supervisée de ce corpus, bien que probablement facile pour un observateur humain, se révèle être une tâche non triviale pour les algorithmes testés.

Une des questions qui se posent est celle du traitement des documents ayant plusieurs étiquettes catégorielles. Lorsqu'un document possède plusieurs étiquettes parmi les cinq retenues, une solution est d'adapter sa probabilité d'appartenance aux différentes classes (1/2, 1/2 par exemple). Mais ce traitement n'est pas très satisfaisant car il est possible que, d'une part, un document soit beaucoup plus lié à une classe qu'à l'autre et que, d'autre part, le document appartienne encore à d'autres catégories que les cinq privilégiées (cette information n'a de toute façon pas été conservée). Finalement, nous avons décidé de ne considérer que la première étiquette pour chaque document, les textes appartenant à plusieurs catégories n'étant pas très nombreux (voir le tableau 4.1).

Nous avons choisi l'option de limiter le plus possible les prétraitements : les chiffres et plus généralement tous les symboles non alphabétiques sont éliminés. Nous faisons l'hypothèse qu'ils sont trop bruités par rapport à l'information qu'ils apportent. Les délimiteurs de mots sont tous les symboles non alphabétiques. Ainsi, par exemple, une forme contenant une chaîne de chiffres est séparée en deux formes ne contenant que des caractères alphabétiques. Par ailleurs, nous avons décidé de conserver les anti-mots par défaut[3], une position validée a posteriori par certains résultats, comme nous le verrons en section 5.3. Dans l'immédiat, nous conservons l'intégralité du vocabulaire, dans la mesure où le corpus est suffisamment petit pour être utilisable sans réduction du vocabulaire. Enfin, en l'absence de preuve claire d'un avantage à le faire, nous n'avons eu recours à aucune technique de racinisation ou de lemmatisation.

Sur le plan pratique, la tâche de *tokenisation* est laissée aux soins de la bibliothèque *bag-of-words* [McCallum, 1996], que nous avons adaptée en C++ pour notre programme *textclust* (cf. annexe C).

[2]Ce corpus comprend 806 791 dépêches Reuters sur de nombreux sujets, diffusées entre août 1996 et août 1997. Les textes sont annotés en XML avec un grand nombre d'informations supplémentaires, dont une qui nous intéresse plus particulièrement : la (ou les) catégorie(s) à laquelle (auxquelles) appartient le document.

[3]Cependant, dans la section 4.3.1, nous étudions l'effet d'une diminution du vocabulaire sur la base des fréquences de mots et une de ces expériences est, de fait, équivalente au filtrage des anti-mots.

Thèmes	Nombre de textes
culture	981
emploi	1000
accidents	992
santé	932
sports	1000
accidents/santé	34
emploi/santé	33
culture/emploi	10
culture/sports	5
accidents/emploi	5
culture/santé	3
accidents/sports	3
culture/accidents	1
santé/sports	1

FIG. 4.1 – Les différentes catégories du corpus sélectionné.

Nous effectuons les expériences sur le corpus présenté en section 4.2.1 à l'aide d'une validation croisée en 10 étapes[4] avec un vocabulaire reconstitué à chaque étape avec l'ensemble des mots du corpus d'apprentissage (40000 environ). Les résultats sont donc moyennés sur l'ensemble des jeux et, lorsque l'étape d'initialisation est aléatoire, nous conduisons également plusieurs répétitions indépendantes[5]). Suivant la lisibilité des résultats, nous représentons soit toutes les trajectoires, soit un résumé sous la forme de boîtes-à-moustaches : les boîtes s'étendent entre les quartiles inférieur et supérieur alors que les moustaches s'allongent des deux côtés jusqu'à ± 1.5 fois l'intervalle inter-quartile (les observations aberrantes, au nombre de quelques unités sur 300, ont été ignorées).

Pour les raisons évoquées lors de l'étude présentée au chapitre 3, notre choix de mesures s'est porté sur la perplexité et les scores de cooccurrences après application de la méthode hongroise.

4.2.2 Initialisation

Il est important de remarquer que l'algorithme EM peut être initialisé indifféremment de valeurs données des paramètres α et β ou de probabilités a posteriori $\mathrm{P}(T_d = t|C;\alpha,\beta)$ particulières. Les formules (4.2), (4.3) et (4.4) nous permettent en effet d'obtenir les paramètres en fonction des probabilités a posteriori et inversement. Par conséquent, la question se pose pour l'initialisation de commencer par l'étape E, avec des valeurs initiales pour les paramètres, ou de commencer par l'étape M avec des valeurs initiales pour les probabilités a posteriori (c'est-à-dire un partitionnement probabiliste initial des documents). Nous préférons la seconde solution, pour plusieurs raisons :

– L'initialisation de la matrice β nécessite de savoir déterminer des valeurs « raisonnables » pour un très grand nombre de paramètres. Une initialisation aléatoire uni-

[4]L'ensemble de test est constitué de 500 documents, qui changent à chaque étape. Le reste du corpus est utilisé pour l'apprentissage. Nous obtenons ainsi 10 *jeux* de données distincts.

[5]Le nombre de 30 répétitions, utilisé dans ce qui suit, nous a semblé empiriquement bien couvrir la diversité des performances obtenues.

forme semble contre-indiquée, dans la mesure où elle produirait des valeurs complètement irréalistes dans l'espace des paramètres dans une grande majorité des cas. Une autre possibilité serait de perturber[6] légèrement les fréquences de mots du modèle unigramme mais il est difficile de quantifier l'adverbe « légèrement » dans ce cas !

– L'initialisation des probabilités a posteriori ne nécessite aucune connaissance sur le modèle. Par exemple, il n'est pas nécessaire de connaître la taille du vocabulaire. Nous verrons en section 4.3 en quoi cette propriété est intéressante.

Ainsi, dans ce qui suit, nous considérerons uniquement des schémas d'initialisation associés aux probabilités a posteriori des documents d'appartenir à un thème donné. Comme il n'y a pas de raison de privilégier un thème a priori, nous allons tirer pour chaque document ses probabilités d'appartenance selon une loi Dirichlet équilibrée. Nous appellerons dans ce qui suit cette stratégie d'initialisation *Dirichlet*. Il reste à déterminer l'intensité de la loi Dirichlet utilisée. Comme l'EM tend à amplifier les différences les plus petites entre composants, nous avons remarqué que la variabilité finale des estimations n'était pas réduite de façon significative en choisissant une intensité importante (variance de la loi Dirichlet plus faible). En pratique, nous avons donc choisi une distribution uniforme sur le simplexe de dimension n_T (Dirichlet de paramètre 1).

Pour avoir une idée de la meilleure performance possible, nous avons également utilisé les catégories Reuters pour une autre stratégie d'initialisation. Nous établissons une bijection entre les composantes du mélange et les catégories Reuters en fixant pour chaque texte la probabilité a posteriori (4.2) à 1 pour le thème correspondant[7].

La figure 4.2 présente la perplexité sur les ensembles d'apprentissage et de test en fonction du nombre d'itérations de l'algorithme EM.

Les variations sont relativement similaires sur les ensembles d'apprentissage et de test. La différence essentielle est que les scores de perplexité sont meilleurs sur l'ensemble d'apprentissage que de test. Ce phénomène classique est une manifestation de sur-apprentissage. Compte tenu de la façon dont le vocabulaire est établi (en écartant les mots qui n'apparaissent pas dans l'ensemble d'apprentissage), cet effet n'est pas observé pour le modèle unigramme[8].

[6] L'introduction d'une perturbation entre deux thèmes est rendue nécessaire par le fait que lorsque deux colonnes de β sont exactement identiques, les équations de réestimation les transforment de la même façon et elles continuent donc à représenter exactement le même thème quel que soit le nombre d'itérations de l'algorithme EM.

[7] Il s'agit naturellement d'une solution totalement inapplicable sur un problème réel, puisque reposant sur la solution que l'on cherche.

[8] La plupart des mots rares de l'ensemble de test sont ignorés ce qui fait mécaniquement augmenter les fréquences des autres mots. Notons $P_w = \frac{\sum_{d=1}^{n_D} C_{wd} + \lambda_{\text{lissage}}}{l + n_W \lambda_{\text{lissage}}}$ la probabilité unigramme du mot w et $f_w, f_w^\star$ les fréquences de mots dans les ensembles d'apprentissage et de test et définissons arbitrairement une fréquence seuil θ qui sépare les mots rares des mots « non rares ». Les perplexités unigrammes sont égales à :

$$\widehat{\mathcal{P}} = \exp\left(-\sum_{w=1}^{n_W} f_w \log P_w\right) = \exp\left(-\sum_{\text{mots rares}} f_w \log P_w - \sum_{\text{mots non rares}} f_w \log P_w\right)$$

$$\widehat{\mathcal{P}}^\star = \exp\left(-\sum_{w=1}^{n_W} f_w^\star \log P_w\right) = \exp\left(-\sum_{\text{mots non rares}} f_w^\star \log P_w\right),$$

$$\widehat{\mathcal{P}} = \exp(-\sum_{w} f_w \log P_w) = \exp(-\sum_{f_w \leq \theta} f_w \log P_w - \sum_{f_w > \theta} f_w \log P_w)$$

$$\widehat{\mathcal{P}}^\star = \exp(-\sum_{w} f_w^\star \log P_w) = \exp(-\sum_{f_w > \theta} f_w^\star \log P_w),$$

L'observation la plus évidente est que l'écart entre les deux initialisations est très grand. La distribution prédictive sur les mots avec l'initialisation Dirichlet est meilleure qu'avec le modèle unigramme mais bien pire qu'avec l'initialisation « idéale ». Cet écart est également flagrant pour les scores de cooccurrences : sur l'ensemble de test, la performance finale de l'initalisation *catégories Reuters* est de 0.95 contre une moyenne de 0.6 pour l'initialisation Dirichlet.

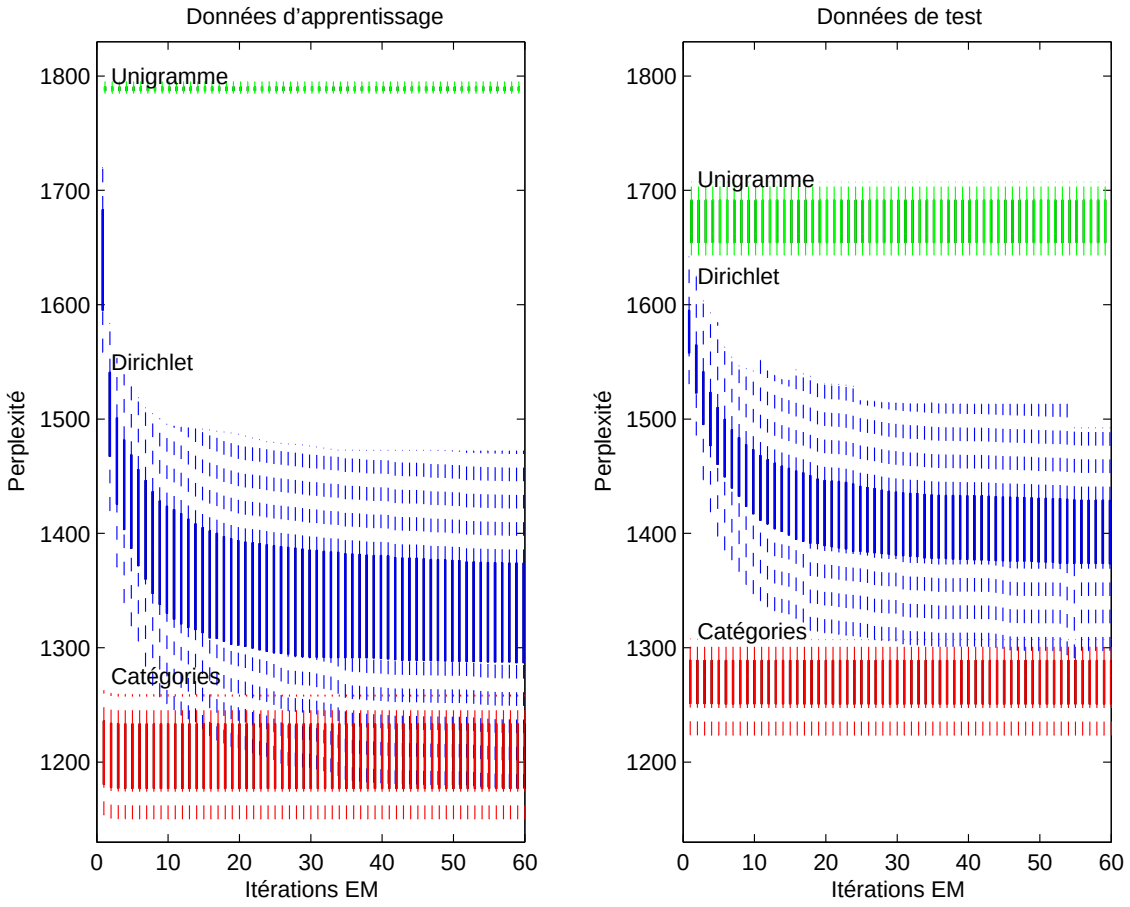

FIG. 4.2 – Variations de la perplexité sur les ensembles d'apprentissage et de test en fonction du nombre d'itérations d'EM.

Étant donné que l'initialisation Dirichlet a un caractère aléatoire, il peut être instructif de mesurer les évolutions de performances d'un essai à l'autre. Nous représentons les valeurs de perplexité sur l'ensemble d'apprentissage et les scores de cooccurrence sur l'ensemble de test pour une série d'exécutions sur le premier jeu sur la figure 4.3[9]. Comme le montre la figure, la variabilité d'une initialisation à une autre est très grande pour les deux mesures : les scores de cooccurrence sont par exemple étalés entre 0.4 et un peu plus de 0.7. Cette variabilité est révélatrice de la tendance de l'algorithme EM à rester bloqué dans l'un des

en supposant grossièrement qu'il n'y ait pas de mots rares dans l'ensemble de test. Or, si $f^\star_w > f_w$ pour les mots fréquents, de même que les $\log P_w$, et puisque $\sum_w f^\star_w = \sum_w f_w = 1$, nous obtenons : $\sum_{f_w \leq \theta} f_w \log P_w + \sum_{f_w > \theta} f_w \log P_w < \sum_{f_w > \theta} f^\star_w \log P_w$. $\widehat{\mathcal{P}} > \widehat{\mathcal{P}}^\star$ se déduit alors immédiatement.

[9] Dans ce qui suit, les scores de perplexité sont données uniquement sur l'ensemble d'apprentissage (adaptation du modèle aux données) et la performance pour la tâche de classification non supervisée est essentiellement mesurée par les scores de cooccurrences sur l'ensemble de test.

nombreux maximums locaux d'un espace des paramètres de très grande dimension.

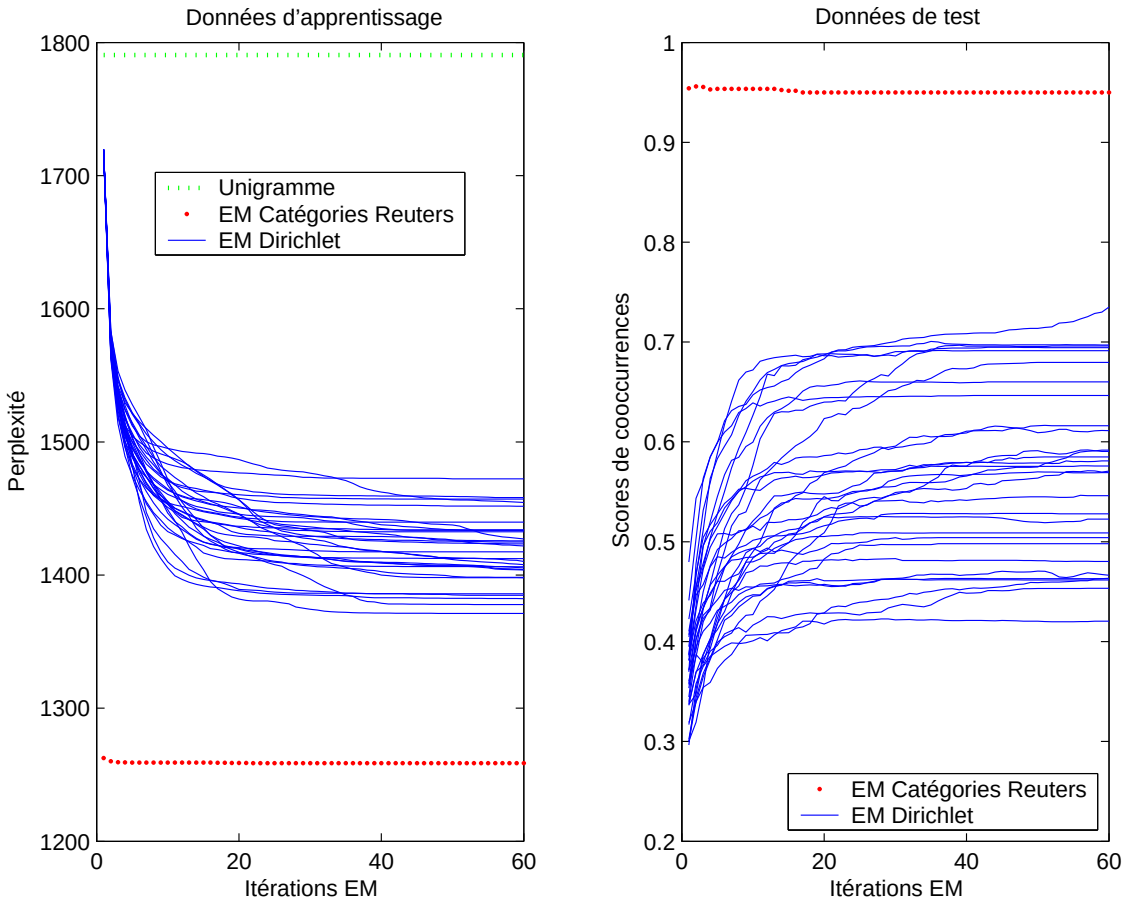

Fig. 4.3 – Perplexité et score de cooccurrence en fonction des itérations de l'EM pour différentes initialisations Dirichlet sur le premier jeu.

4.2.3 Influence du paramètre de lissage

La figure 4.4 illustre l'influence du paramètre de lissage $\lambda_\beta - 1$ en termes de perplexité et de scores de cooccurrences. Nous ne nous intéressons pas ici à l'influence de $\lambda_\alpha - 1$, dans la mesure où il est, dans ce contexte, toujours négligeable par rapport à la somme sur tous les documents des probabilités a posteriori. Pour l'initialisation sur les catégories Reuters, il n'y a quasiment aucune différence au niveau des scores de perplexité pour des valeurs de $\lambda_\beta - 1$ faibles (c'est-à-dire $\lambda_\beta - 1 \leq 0.2$). Les performances se dégradent régulièrement ensuite, ce qui montre qu'une partie de l'information utile est perdue, probablement correspondant à l'ensemble des mots rares (puisque le lissage concerne en priorité les paramètres estimés sur très peu d'occurrences). Pour l'initialisation Dirichlet, les variations de perplexité ne sont pas très importantes non plus sur l'intervalle [0.01,1], mais il est cette fois possible d'isoler un optimum, autour de 0.2. L'incorporation d'une conviction a priori selon laquelle les probabilités d'apparition des mots ne devraient jamais être très petites, même sur des associations terme-thème très rares, est donc profitable pour l'ajustement du modèle à de nouvelles données.

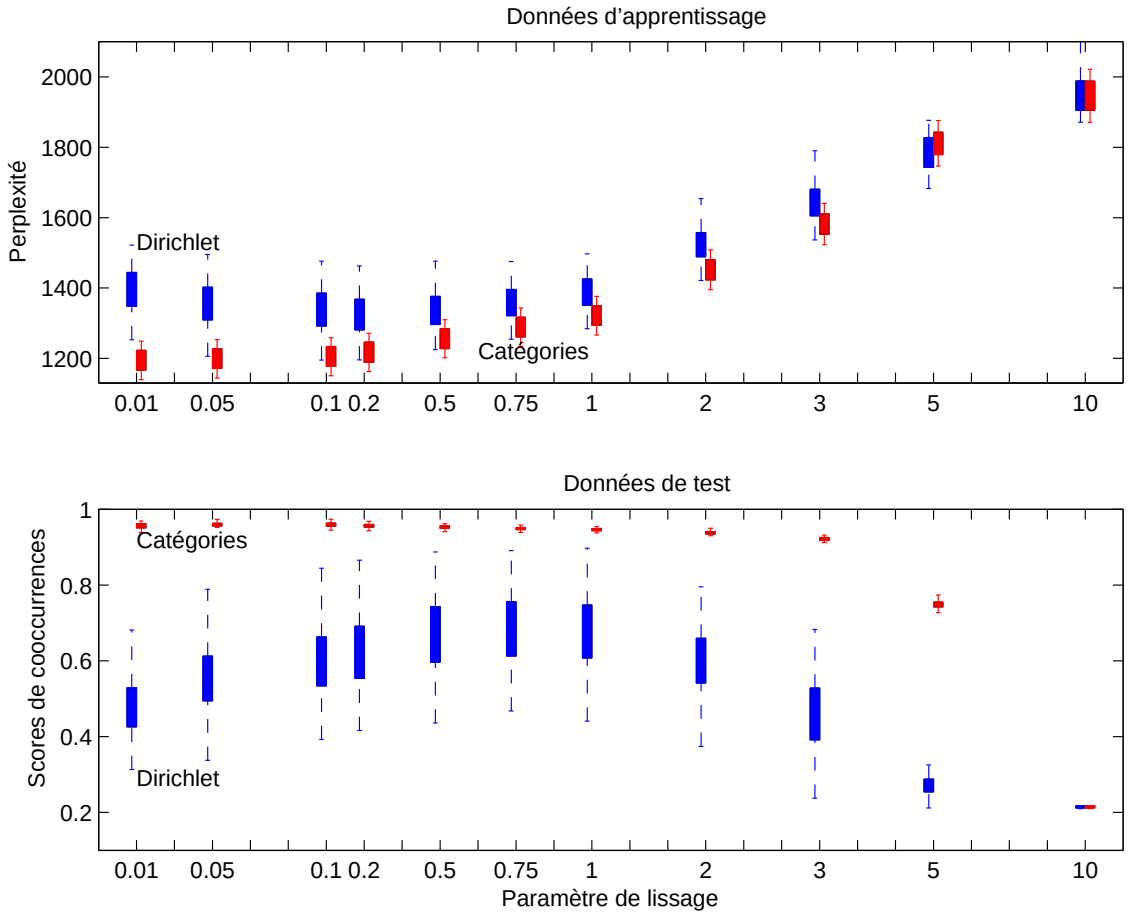

FIG. 4.4 – Perplexité sur l'ensemble d'apprentissage et cooccurrences sur l'ensemble de test, en fonction du paramètre de lissage $\lambda_\beta - 1$.

L'observation des scores de cooccurrences sur l'ensemble de test va dans le même sens. Remarquons tout d'abord que les performances de classification avec l'initialisation sur les catégories Reuters ne sont quasiment pas influencées par le paramètre de lissage (à part pour les valeurs supérieures à 5, qui marquent une chute particulièrement importante). Mais nous sommes en pratique plus intéressés par le comportement obtenu avec l'initialisation Dirichlet : les résultats varient également assez peu, avec cette fois un maximum pour une valeur de lissage égale à 0.75. Indéniablement, le lissage a un effet positif sur les capacités de catégorisation (même lorsqu'il dégrade légèrement l'ajustement). Une explication possible est que la plupart des paramètres sont très mal estimés à cause de la distribution « zipfienne » des données et que, donc, seuls les mots relativement fréquents sont une aide à la classification. Les autres la perturbent plutôt, sauf s'ils sont correctement initialisés. Cette hypothèse incite à modifier le vocabulaire, une expérience que nous présentons en section 4.3.1. Dans l'immédiat, nous mettons en évidence un autre phénomène lié à la dimensionnalité qui a trait à la distribution a posteriori des indicatrices de thème.

En conclusion, les variations des valeurs de $\lambda_\alpha - 1$ et $\lambda_\beta - 1$ ne semblent pas avoir d'importance cruciale sur les résultats, tant qu'elles restent dans un intervalle raisonnable (que nous avons déterminé expérimentalement entre 0 et 1 pour $\lambda_\beta - 1$). Par conséquent, dans ce qui suit, nous les fixons respectivement à 0 et 0.1.

4.2.4 Comportement de l'algorithme EM en grande dimension

Un constat surprenant en travaillant avec ce modèle est qu'une immense majorité des probabilités a posteriori qu'un document appartienne à un thème est très proche de 0 ou 1. Ainsi, pour l'initialisation sur les catégories Reuters, la proportion de textes classés dans un thème unique avec probabilité 1 (à la précision numérique de l'ordinateur près) est quasiment de 100%. L'effet est un peu moins marqué avec l'initialisation Dirichlet, ce qui semble logique dans la mesure où nous partons potentiellement de points opposés (d'équiprobabilité des thèmes) dans le simplexe. Malgré tout, après la cinquième itération, plus de 90% des documents sont classés avec une certitude absolue.

Nous nous sommes par conséquent interrogé sur les différences de comportement avec un algorithme qui serait similaire à EM mais qui s'appuyerait sur des classifications déterministes. Il s'agit en réalité d'une version de l'algorithme des K-moyennes avec une distance particulière entre un texte $d \in \{1, \ldots, n_D\}$ et un thème $t \in \{1, \ldots, n_T\}$ (c'est-à-dire entre un texte et le centroïde du groupe considéré) :

$$dist(d,t) = -\log\left(\alpha_t \prod_{w=1}^{n_W} \beta_{tw}^{C_{wd}}\right)$$

Notons que toute fonction décroissante de la similarité $\alpha_t \prod_{w=1}^{n_W} \beta_{tw}^{C_{wd}}$ produirait les mêmes résultats. Le choix de $-\log$ permet toutefois de faire une analogie avec la distribution gaussienne pour laquelle la densité est définie, aux facteurs multiplicateurs près, comme une inverse de l'exponentielle de la distance. Autre analogie intéressante : si l'on omet le terme $-\log \alpha_t$, en général négligeable, la distance devient $-\sum_{w=1}^{n_W} C_{wd} \log \beta_{tw}$, ce qui, à des facteurs multiplicatifs et additifs (constants par rapport à t) près, est égal à :

$$\sum_{w=1}^{n_W} \frac{C_{wd}}{l} \log \frac{C_{wd}/l}{\beta_{tw}},$$

à savoir la divergence de Kullback-Leibler entre la distribution unigramme prédite à partir de ce document et la distribution thématique définie par les β_{tw}. En d'autres termes, on considère qu'un document est d'autant plus proche d'un thème que la divergence de Kullback-Leibler entre son profil probabiliste et les β_{tw} associés au thème est faible.

Cette distance est calculée pour tous les documents et tous les thèmes et chaque texte est affecté au thème le plus proche. Les paramètres β_{tw} sont ré-estimés conformément à (4.4), les probabilités a posteriori étant donc dans ce cas exclusivement égales à 0 ou 1. α_t est alors simplement la proportion des documents qui appartiennent au thème t et β_{tw} la contribution en occurrences du mot w au thème t divisée par le nombre total d'occurrences dans ce thème.

$$\alpha_t = \frac{1}{n_D} \sum_{d=1}^{n_D} \mathbb{1}_{\{d \in t\}} \text{ et } \beta_{tw} = \frac{\sum_{d \in t} C_{wd}}{\sum_{w=1}^{n_W} \sum_{d \in t} C_{wd}}$$

Nous avons appliqué cet algorithme sur le même corpus, avec les mêmes stratégies d'initialisation que ci-dessus. À la fin de chaque itération, nous avons calculé le score de cooccurrences entre la classification probabiliste d'EM et la classification déterministe de cette version des K-moyennes.

- À partir des catégories Reuters, le score de cooccurrences entre les classifications est supérieur à 0.99 après 10 itérations (écart-type sur les 10 jeux : 0.002).

- Avec l'initialisation Dirichlet, les scores de similarité convergent également très vite pour atteindre un score moyen de 0.92 après 10 itérations (écart-type sur les 10 jeux et 3 trajectoires par jeu : 0.03).

La proximité des résultats finals des méthodes probabiliste et déterministe incite donc à penser que l'apport du cadre probabiliste est limité dans cette situation.

Nous pensons que cette tendance des probabilités a posteriori d'appartenance d'être presque égales à 0 ou 1 peut s'expliquer en partie par l'étude de deux paramètres :

- la très grande dimension (de l'ordre de 40000) de l'espace du vocabulaire ;
- les longueurs des textes l_d.

Pour illustrer ce phénomène, nous faisons appel à des données artificielles générées selon la démarche suivante :

- nous considérons la loi de Zipf vue en section 2.3.7.1, la probabilité d'apparition du mot numéro w est Z/w (avec $Z \approx 0.06$ sur notre corpus) ;
- simuler n_T distributions thématiques $\beta_t \sim \mathrm{Dir}(k\frac{Z}{1}, k\frac{Z}{2}, \ldots, k\frac{Z}{n_W})$.
  k est liée à l'intensité de la loi de Dirichlet : plus k est grand, plus les différentes distributions thématiques se ressemblent. Expérimentalement, il semble que la valeur $k = 1000$ donne des résultats relativement proches des données réelles ;
- pour tout texte $d = 1, \ldots, n_D$ ($n_D = 5000$),
  - tirer un thème $T_d \sim \mathrm{Mult}(1, (\frac{1}{n_T}, \ldots, \frac{1}{n_T}))$;
  - conditionnellement à T_d, tirer les l_d occurrences[10] du texte, $C_d \sim \mathrm{Mult}(l_d, \beta_{T_d})$.

Nous répétons le procédé pour différentes valeurs de l_d et n_W et appliquons à chaque fois l'algorithme EM sur ces données simulées (le nombre d'itérations de l'algorithme EM est de 20, n_T est fixé à 5 pour la génération des données comme pour l'inférence). Nous représentons un histogramme des n_D probabilités a posteriori en figure 4.5.

Le fait que l_d et n_W ont une influence directe sur le degré de « certitude » des prédictions de l'algorithme EM est ici manifeste. l_d est le nombre de facteurs dans le produit

$$\prod_{w=1}^{n_W} \beta_{tw}^{C_{wd}} \tag{4.7}$$

qui intervient dans le calcul des probabilités a posteriori et n_W détermine l'ordre de grandeur des β_{tw} eux-mêmes. Lorsque ces deux paramètres augmentent, il est donc clair que le produit (4.7) est de plus en plus loin de 1. Néanmoins, le phénomène que nous constatons reste difficile à expliquer : plus n_W et l_d augmentent, plus, pour chaque document, l'algorithme EM a tendance à trouver une composante pour laquelle le produit (4.7) est beaucoup plus grand que pour toutes les autres. Ajoutons que pour ces expériences sur données artificielles, nous avons conduit une série de calculs pour une valeur plausible de l_d (300) mais pas pour des valeurs réalistes de la taille du vocabulaire, pour laquelle nous n'avons pas dépassé 100. Naturellement, pour des dimensionnalités supérieures, et notamment pour la dimensionnalité réelle de 40000, les histogrammes présentent, a fortiori, la même concentration autour de 0 et 1.

Par ailleurs, cette confiance dans les prédictions n'a aucun lien avec les performances réelles de l'algorithme, qui sont loin d'être parfaites en grande dimension, comme le soulignent les sections 4.2 et 4.3. Le modèle a donc ceci de commun avec un mauvais homme politique qu'il est toujours persuadé d'avoir raison, alors même qu'il a souvent tort. Une

[10] Les longueurs l_d des textes sont supposées constantes. Elles pourraient être modélisées par une loi de Poisson mais nous n'avons pas retenu cette option, qui nécessite plus de temps de calcul pour un résultat guère plus proche des données réelles.

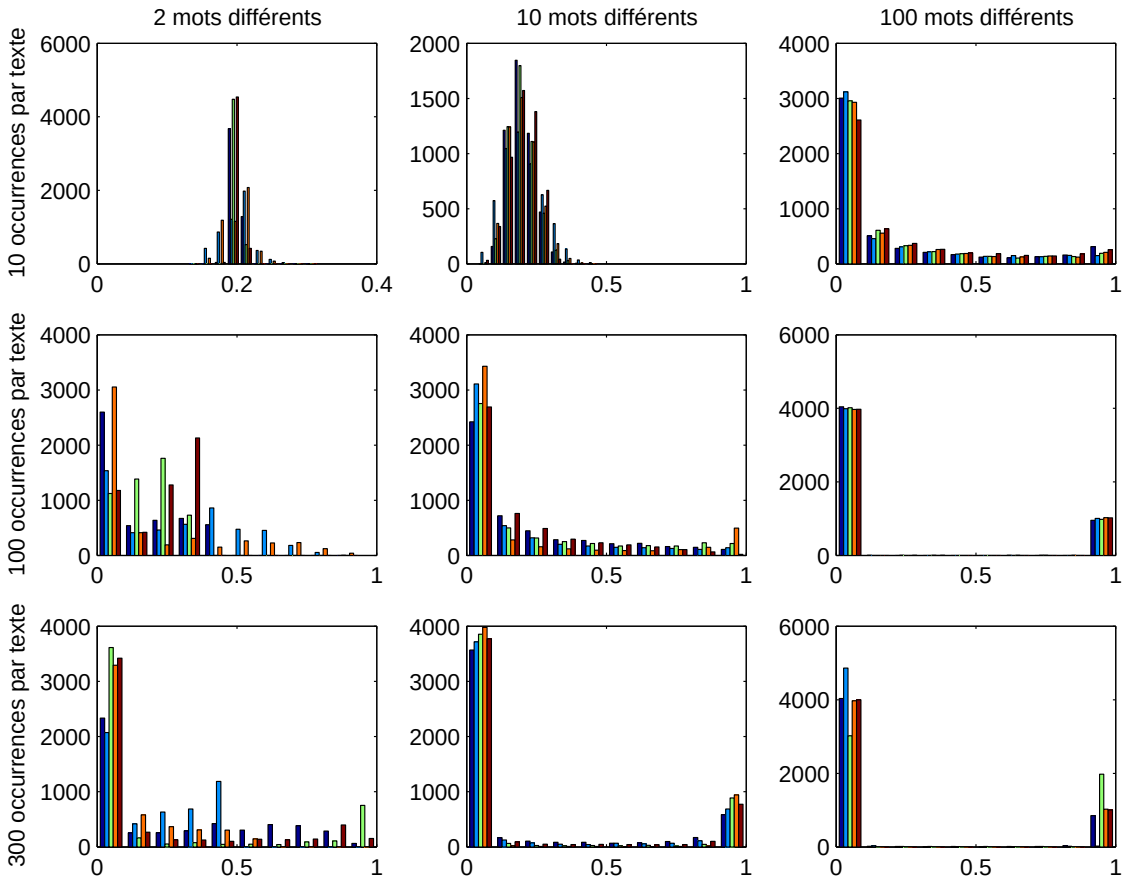

Fig. 4.5 – Histogramme des probabilités a posteriori sur des données simulées, pour différents l_d et n_W.

stratégie pour améliorer l'inférence consiste précisément à jouer sur l'aspect des probabilités a posteriori par le biais d'un paramètre de température. C'est le principe de l'EM tempéré dont nous avons déjà relevé le côté arbitraire au chapitre 2. Nous avons par conséquent cherché d'autres moyens d'améliorer l'inférence.

4.3 Amélioration des performances de l'EM par réduction de la dimensionnalité

Dans cette section, nous développons notre intuition que le retrait des mots rares ne devrait pas nuire aux performances de l'algorithme EM et pourrait éventuellement fournir une solution aux difficultés d'initialisation évoquées précédemment. Après avoir étudié l'effet de la réduction de la dimensionnalité, nous présentons une stratégie d'inférence itérative qui en exploite les aspects positifs.

4.3.1 Ajustement de la taille du vocabulaire

Une question naturelle, après avoir décidé de réduire la taille du vocabulaire, est de déterminer s'il est plus pertinent de retirer les mots rares ou les mots fréquents (nous avons évoqué en section 2.1.1 la possibilité que la liste des mots les plus fréquents, appelés

anti-mots ou stop words, contienne peu de mots porteurs de sens). Nous comparons dans cette section les deux stratégies en retirant successivement plusieurs dizaines, centaines puis milliers de mots du vocabulaire. Cette opération s'effectue en supprimant les lignes correspondantes de la matrice de comptes.

La figure 4.6 montre qu'il est possible d'améliorer de façon significative les performances du modèle avec l'initialisation Dirichlet en ne conservant qu'un faible nombre de mots fréquents (900 sur 40000).

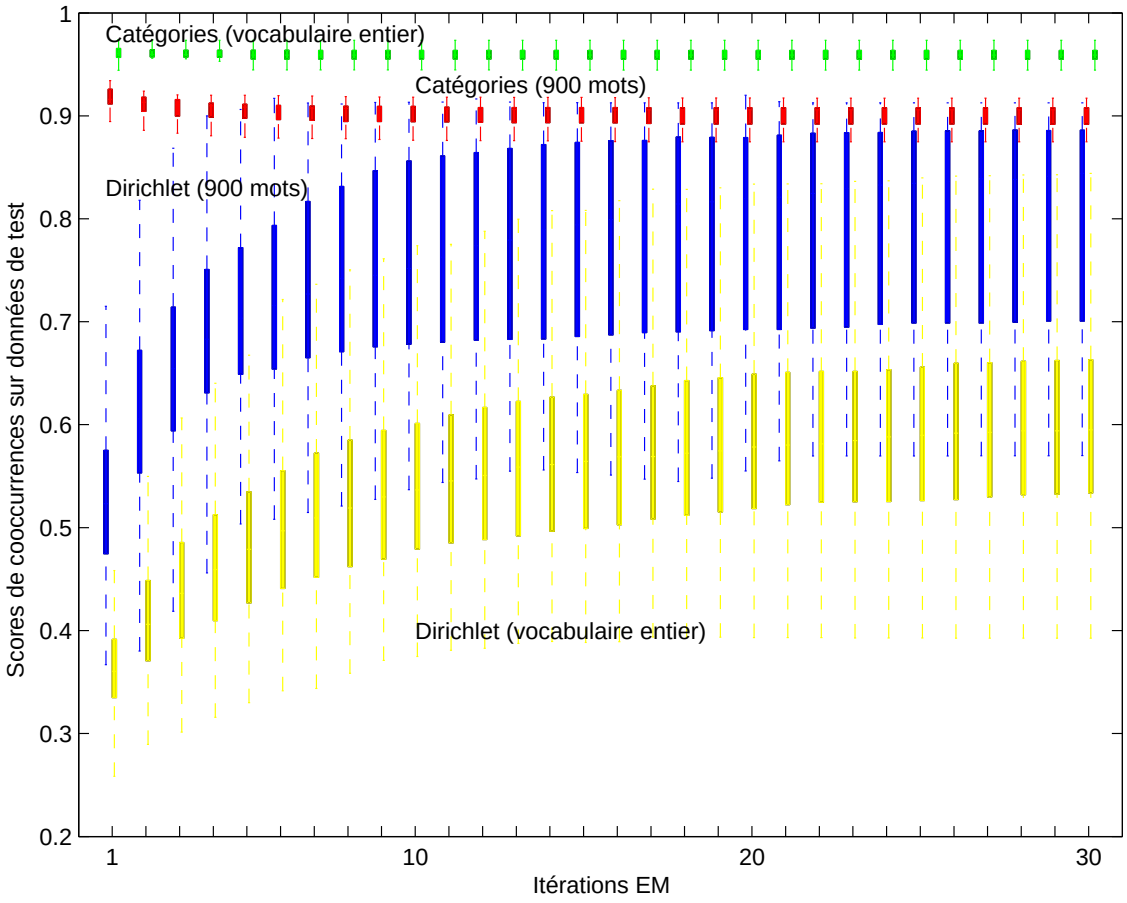

FIG. 4.6 – Scores de cooccurrences en fonction des itérations de l'EM pour un vocabulaire de taille 900.

Lorsque les tailles de vocabulaire sont différentes, la comparaison des scores de perplexité n'a pas de sens puisque la dimensionnalité a un effet direct sur l'échelle de cette mesure qui masque automatiquement tout effet éventuel de meilleur ajustement aux données. Les scores de cooccurrences restent en revanche significatifs même si les vocabulaires ne sont pas identiques.

Nous exposons en figure 4.7 une étude plus systématique en rapportant les scores de cooccurrences sur l'ensemble de test après la trentième itération en fonction de la taille du vocabulaire. Dans un souci de lisibilité, l'axe des abscisses n'est pas gradué de façon régulière mais plutôt de sorte à insister sur les parties les plus remarquables, c'est-à-dire entre 100 et 3000 pour les mots fréquents et au-dessus de 40000 (sur un vocabulaire de 43320) pour les mots rares. Ceci est une conséquence du fait que la plupart des occurrences

sont localisées dans les mots fréquents : ainsi, en ne conservant que les 3320 mots les plus courants, nous gardons en fait 75% du nombre total d'occurrences.

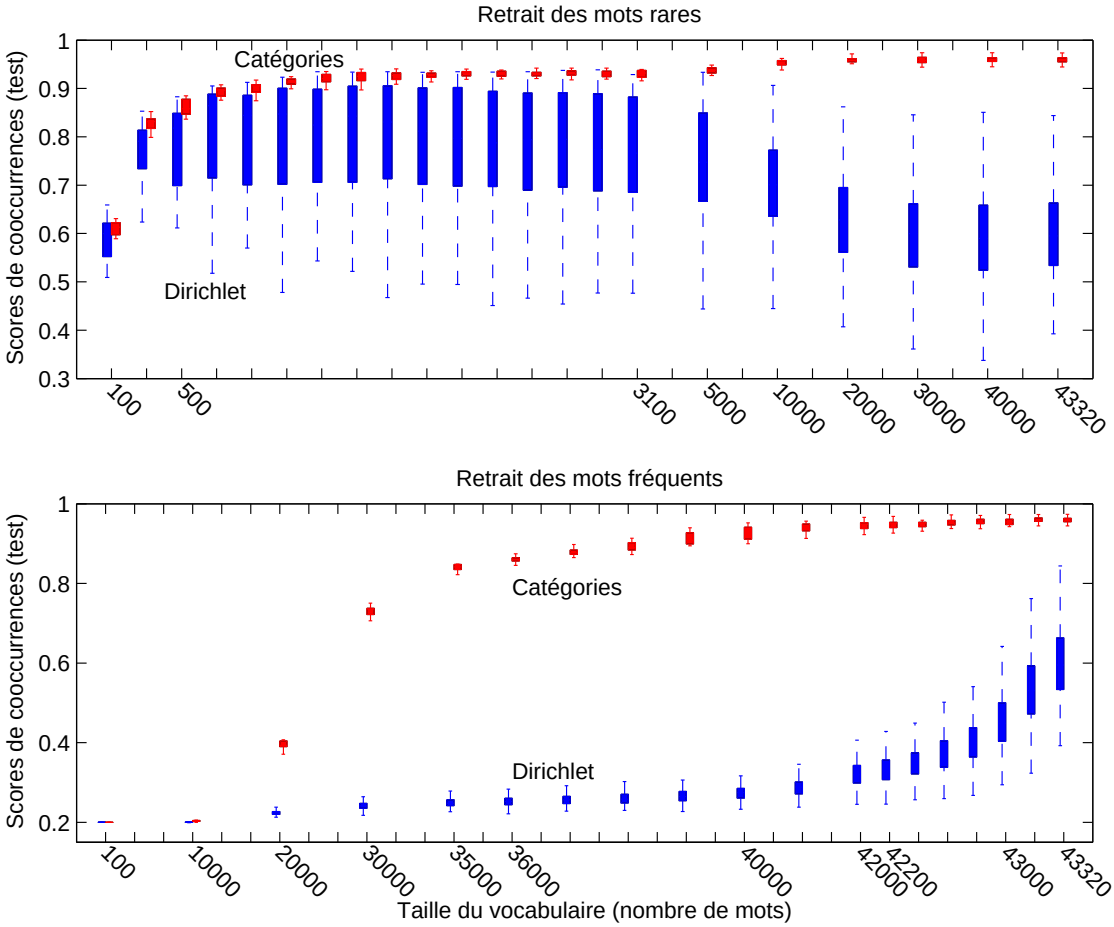

Fig. 4.7 – Scores de cooccurrences sur l'ensemble de test en fonction du vocabulaire pour deux stratégies différentes : sur le graphe du haut, il s'agit de supprimer les mots les plus rares ; sur celui du bas, les mots les plus fréquents.

Le premier graphe de la figure 4.7 montre que le retrait des mots rares est toujours néfaste avec l'initialisation des catégories Reuters alors que, avec l'initialisation Dirichlet, les performance sont clairement meilleures avec un vocabulaire entre 300 et 3000 mots. La taille « optimale » de vocabulaire, tout au moins en regard de cette mesure spécifique, semble être aux alentours de 1000. De manière non moins importante, la stabilité des performances semble également accrue pour de très petites tailles de vocabulaire et il s'agit d'un effet intéressant que nous n'avions pas observé avec les stratégies de lissage. Nous y reviendrons dans la sous-section suivante. Dans l'immédiat, notons que, à nouveau, le meilleur score obtenu avec l'initialisation Dirichlet, reste très inférieur aux performances atteintes avec l'initialisation catégories Reuters. Cela est cohérent avec l'hypothèse que lorsque leurs paramètres sont correctement initialisés, même les mots les plus rares ont une influence positive.

De façon moins surprenante, le second graphe montre que le retrait des mots fréquents a presque toujours un effet négatif. Une seule exception se distingue : avec l'initialisation

catégories Reuters, le fait d'ignorer les 100 mots les plus fréquents (qui sont bien les anti-mots) conduit à une très légère amélioration, mais qui est à peine visible sur la figure. Ensuite le score diminue régulièrement au fur et à mesure que l'on retire les autres mots fréquents pour atteindre la limite inférieure de 0.2 (accord aléatoire) avec 20000 mots rares, ce qui n'a rien d'étonnant puisque, dans ce cas, le vocabulaire n'est plus constitué que de mots n'apparaissant qu'une fois dans le corpus (hapax) et chaque texte est alors réduit à une douzaine de termes tout au plus.

4.3.2 Inférence itérative

Les deux conclusions de ces expériences les plus importantes pour la suite sont que :
- Réduire la dimensionnalité (taille du vocabulaire) à nombre d'observations (taille du corpus) constant améliore globalement la qualité de l'inférence ;
- Lorsque le nombre de mots est très faible, nous parvenons également à limiter de façon significative la variabilité des estimateurs (même si c'est au prix d'une performance inférieure à la meilleure possible).

Dans cette sous-section, nous montrons en quoi la seconde propriété peut se révèler plus importante encore que la première en inspirant une stratégie d'inférence plus fiable (au sens de *moins variable*). Le principe est d'obtenir des probabilités a posteriori « raisonnables » sur un vocabulaire très réduit et de les utiliser ensuite en tant qu'initialisations pour une nouvelle série d'itérations de l'EM, avec un vocabulaire plus grand. Nous contournons ainsi le problème de l'initialisation des paramètres β relatifs aux mots rares en partant de l'autre étape de l'algorithme (étape M). Lorsque de nouveaux mots sont ajoutés, les probabilités qui leur sont associées sont donc automatiquement initialisées sur les comptes moyens de ces mots, pondérés par les probabilités a posteriori thématiques. Notre objectif en scindant ainsi le processus d'inférence en plusieurs étapes suffisament stables est d'obtenir une stratégie globale fiable, atteignant à chaque tentative un bon score.

Nous présentons en figure 4.8 les résultats de trois expériences consistant à lancer 15 itérations de l'algorithme EM sur des vocabulaires de tailles limitées (respectivement 40 mots, 200 mots et 800 mots), à conserver les valeurs des probabilités a posteriori ainsi obtenues puis à reconduire 15 itérations en partant de ce point, mais avec le vocabulaire entier. Nous ne présentons que les scores de cooccurrences dans la mesure où, comme dit précédemment, les comparaisons en termes de perplexité ne sont valables que pour des vocabulaires de même taille. Nous indiquons également les résultats précédents obtenus sur le vocabulaire entier en figure 4.3 à titre de comparaison. Nous constatons l'importance de la taille initiale de vocabulaire : plus celle-ci est grande, plus le score maximal est bon. Cependant cette amélioration des performances maximales se fait également au détriment de la stabilité des résultats : alors que les différentes trajectoires lancées sur un vocabulaire de 40 mots restent comprises à l'issue de la 15ème itération entre des scores de 0.35 et 0.45, l'écart entre pire et meilleure trajectoires est deux fois plus grand avec 200 mots (0.61 et 0.79) et plus de trois fois plus important (0.57 et 0.90) et en partant de 800 mots .

La solution semble donc être de procéder graduellement, en ajoutant les paramètres à apprendre à un rythme progressif permettant de préserver la stabilité. C'est cette stratégie que nous présentons en figure 4.9 sur une base de quatre tailles de vocabulaire successives. De cette manière, nous démontrons qu'il est possible, sur l'ensemble des trajectoires, d'améliorer le score maximal tout en maintenant le score minimal dans un intervalle acceptable. Il est évident, suite à ces expériences, que le choix des tailles successives de vocabulaire est particulièrement crucial, puisque c'est un compromis délicat entre qualité et stabilité.

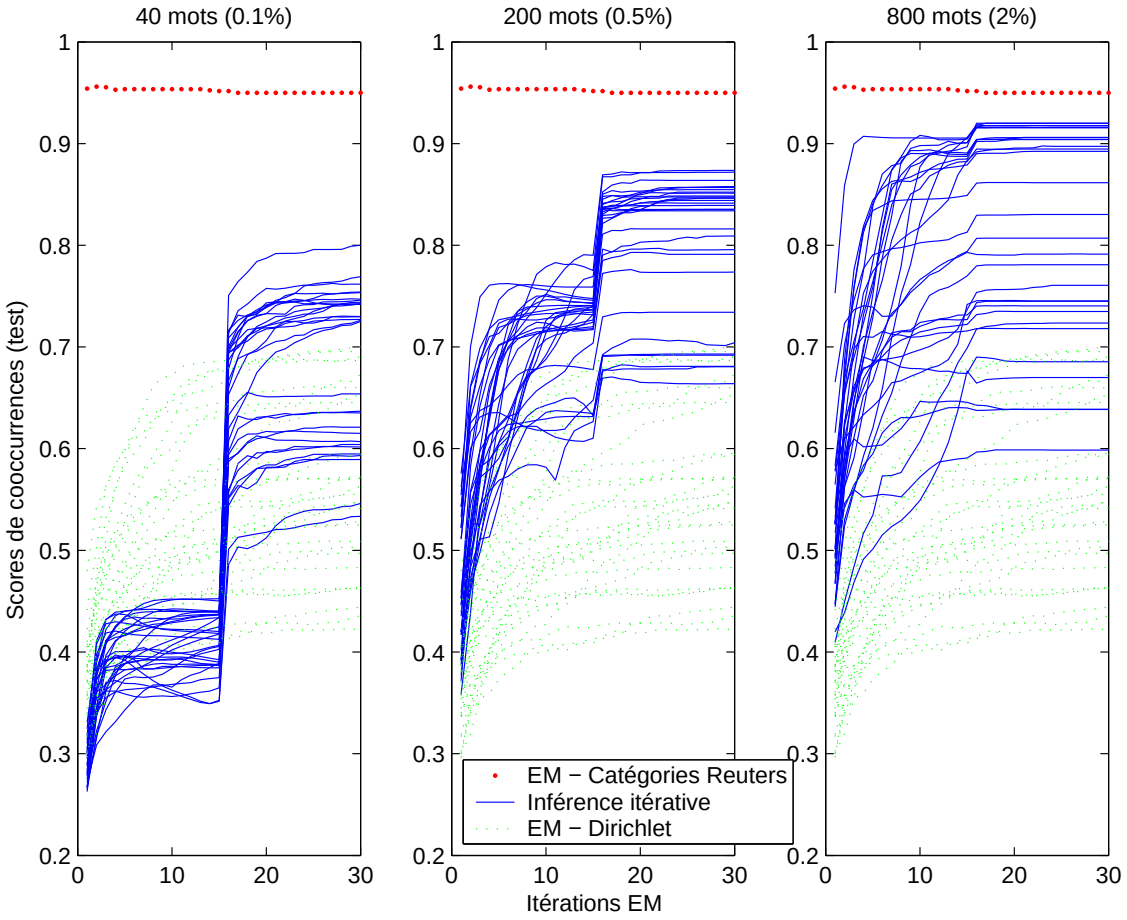

Fig. 4.8 – Scores de cooccurrences sur l'ensemble de test pour différentes tailles de vocabulaire. Les 15 premières itérations sont conduites sur des vocabulaires réduits dont les tailles sont indiquées en titres des figures, alors que les 15 dernières sont lancées sur l'ensemble du vocabulaire.

Nous avons utilisé lors de ces expériences une stratégie d'« essais-erreurs » mais déterminer les intervalles de taille de façon mieux justifiée théoriquement permettrait d'améliorer l'applicabilité de cet algorithme.

Nous avons jusqu'à présent cherché des moyens de réduire la variance de la qualité de la classification afin qu'un utilisateur final du système pour une application réelle obtienne avec certitude un résultat suffisamment fiable sans avoir à se soucier d'initialisations multiples ou de mesures d'évaluation. Toutefois, une autre approche courante devant un problème présentant un grand nombre d'optimums locaux de qualité variable est une stratégie de *démarrages multiples* (multiple restarts) consistant à lancer plusieurs trajectoires et à conserver la meilleure selon un certain critère. On parle également de *recherche par faisceau* (beam search) pour désigner cette heuristique. Ici, une stratégie raisonnable serait donc de sélectionner la taille de vocabulaire conduisant à la meilleure performance, par exemple 800 mots, si nous nous référons à la figure 4.8, exécuter plusieurs essais et choisir celui produisant le meilleur score de cooccurrences sur l'ensemble de test.

Néanmoins, dans un cas d'application réel de classification non supervisée, d'une part, nous ne souhaitons pas isoler d'ensemble de test mais plutôt conserver l'ensemble des

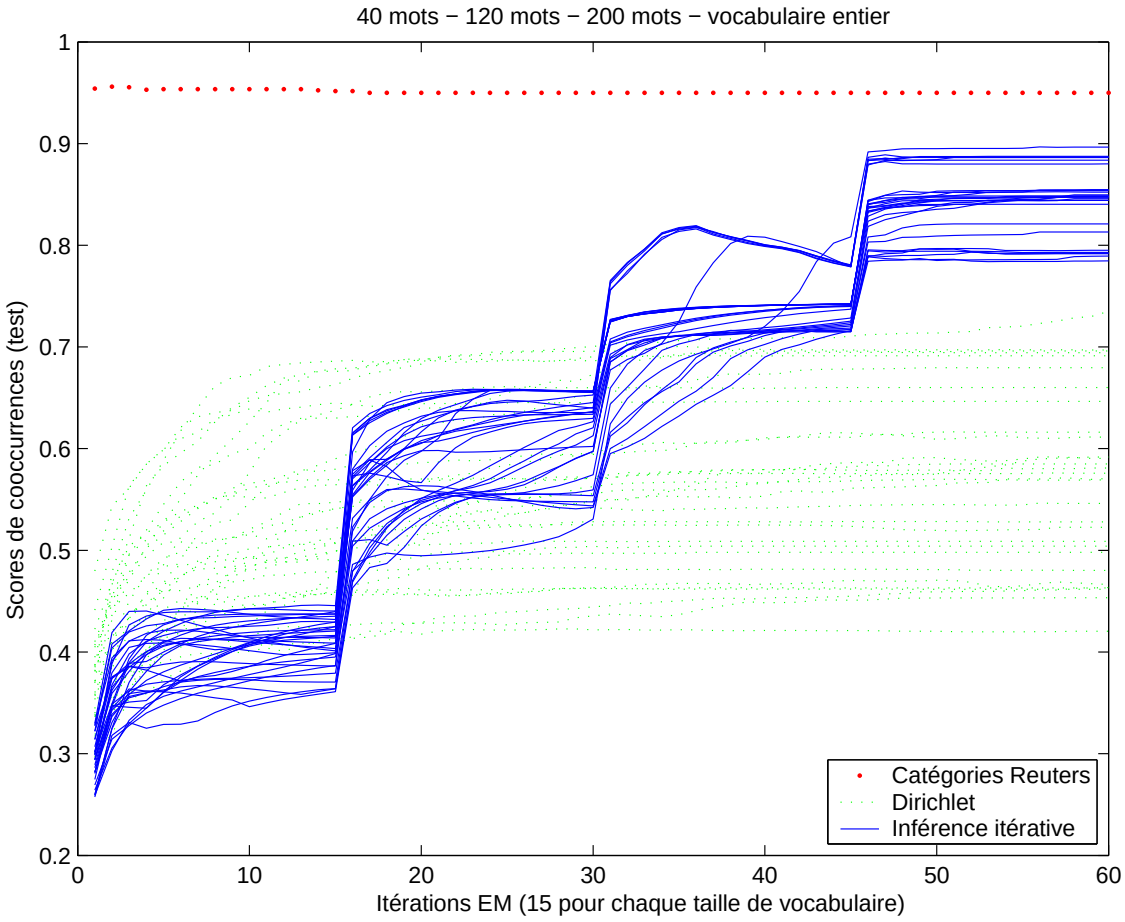

FIG. 4.9 – Scores de cooccurrences sur l'ensemble de test pour différentes étapes d'un algorithme itératif (30 trajectoires sur le même jeu de données sont ici représentées).

documents pour l'apprentissage et, d'autre part, nous ne disposons pas d'information sur ce que peut être le meilleur classement et donc ne pouvons calculer un score de cooccurrences. En revanche, une mesure disponible dans tous les cas dès la phase d'apprentissage est la perplexité. Par conséquent, la question à poser ici est de savoir si la perplexité sur l'ensemble d'apprentissage est un indicateur fiable pour choisir les meilleures trajectoires (au sens des scores de cooccurrences)[11]. La réponse est oui, comme le montre la figure 4.10.

Nous avons repris l'expérience consistant à lancer 15 trajectoires de l'EM sur un vocabulaire de 800 mots avant d'ajouter tous les autres mots et de conduire 15 itérations supplémentaires. Nous mesurons perplexité sur l'ensemble d'apprentissage et scores de cooccurrences sur l'ensemble de test, à la fin de la quinzième itération sur le vocabulaire réduit d'une part, puis à la fin de la trentième itération sur l'ensemble du vocabulaire, d'autre part. Nous observons une corrélation inverse manifeste, particulièrement dans la zone des meilleures trajectoires (faible perplexité–scores de cooccurrences importants) qui nous intéresse au premier chef. Conserver la trajectoire donnant la perplexité la plus faible

[11] Il est intéressant de noter que cette problématique de corrélation entre l'adéquation d'un modèle aux données et ses performances finales pour une application particulière a déjà été étudiée dans d'autres contextes, comme par exemple l'apport du lissage pour la reconnaissance de la parole [Chen and Goodman, 1996].

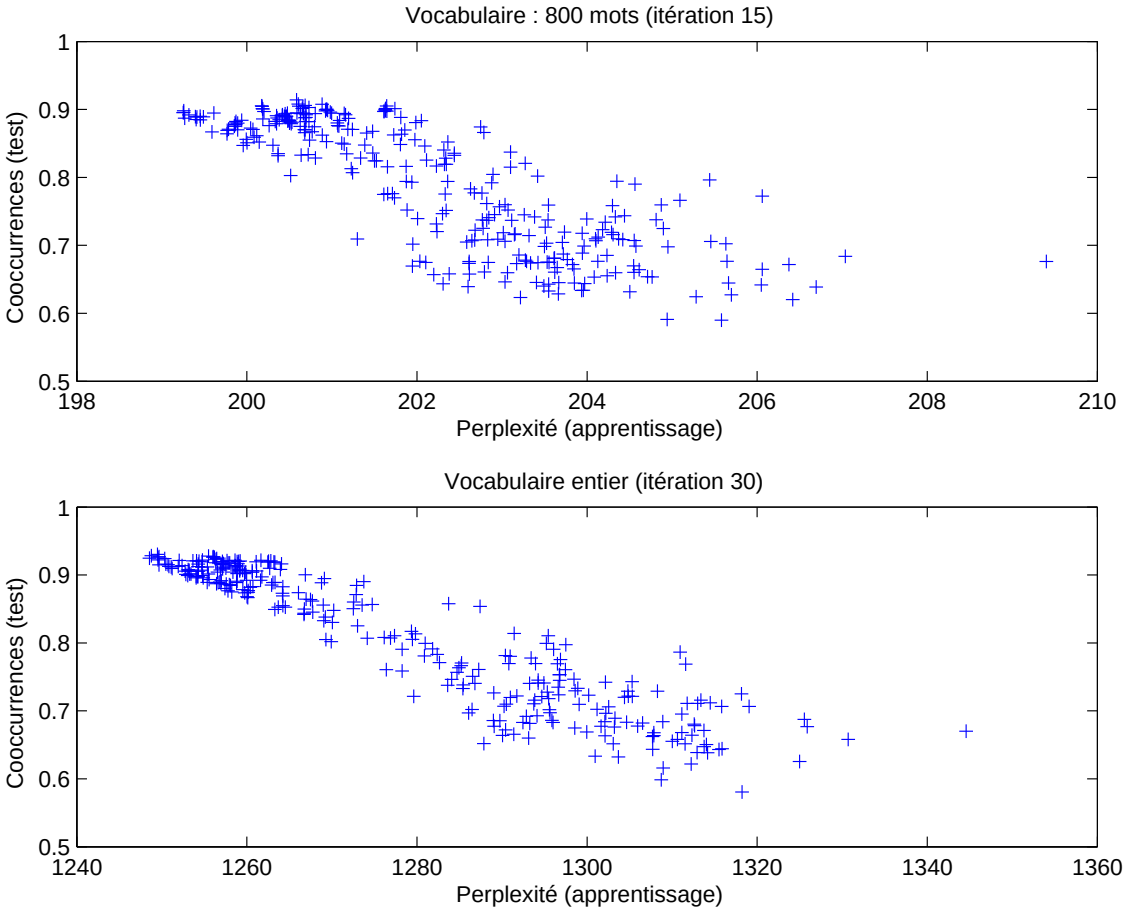

Fig. 4.10 – Correlation entre perplexité sur l'ensemble d'apprentissage et scores de cooccurrences sur l'ensemble de test.

est donc une bonne stratégie dans les deux cas et la corrélation est encore plus forte sur le vocabulaire entier (les 15 itérations avec tous les mots ont donc une valeur ajoutée intéressante dans ce cas).

En conclusion, nous avons présenté dans cette section deux stratégies d'inférence performantes :

- scinder le vocabulaire en plusieurs parties (au moins 4 expérimentalement), lancer l'EM sur le vocabulaire le plus petit et ajouter itérativement des mots plus rares avant de renouveler des itérations d'EM.
- écarter les mots rares, conduire plusieurs itérations d'EM et conserver la trajectoire donnant la perplexité la plus faible sur l'ensemble d'apprentissage.

Dans l'article [Rigouste et al., 2005b], nous montrons que ces deux stratégies peuvent être combinées pour améliorer encore la qualité et la stabilité de l'inférence.

4.4 Échantillonnage de Gibbs

Dans cette section, nous présentons nos expériences avec une méthode d'inférence alternative à l'EM : l'*échantillonnage de Gibbs*[Robert and Casella, 1999]. La première sous-section est dédiée à une présentation générale de la technique alors que les deux suivantes

développent son application dans notre cas. Nous exposons les résultats obtenus avec l'application la plus élémentaire de l'échantillonnage de Gibbs puis nous introduisons une version un peu plus élaborée, dite *rao-blackwellisée* et reposant sur la formule intégrée (4.6).

4.4.1 Présentation

Les méthodes de simulation par chaînes de Markov Monte Carlo (MCMC) sont adaptées aux cas où la loi selon laquelle les échantillons doivent être générés est trop compliquée pour pouvoir appliquer des techniques de simulation plus directes. Leur nom vient du fait que les échantillons successifs sont liés entre eux par une chaîne de Markov et ce sont les propriétés de cette dernière qui garantissent la validité de la simulation [Robert and Casella, 1999]. Un cas particulier de méthode MCMC est l'échantillonnage de Gibbs. Il est applicable lorsque les densités conditionnelles des variables d'intérêt sont suffisamment simples et connues pour être simulées alors que ce n'est pas le cas pour la loi jointe.

Supposons que nous souhaitions simuler un vecteur décomposable en $n \in \mathbb{N}$ sous-vecteurs (éventuellement des scalaires mais pas nécessairement) $X = (X_1, \ldots, X_n)$ selon une densité jointe $\pi(x_1, \ldots, x_n)$. Supposons en plus que la simulation directe ne soit pas possible mais que nous sachions en revanche générer des échantillons selon les lois conditionnelles $\pi_k(x_k|x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)$ pour $k \in \{1, \ldots, n\}$. L'algorithme d'échantillonnage de Gibbs s'écrit comme suit : en partant d'un état initial arbitraire $x^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})$, le passage de l'état $x^{(i)}$ à $x^{(i+1)}$ s'effectue par la boucle :

- pour $k = 1 \ldots n$,
  - simuler $x_k^{(i+1)} \sim \pi_k(\bullet|x_1^{(i+1)}, \ldots, x_{k-1}^{(i+1)}, x_{k+1}^{(i)}, \ldots, x_n^{(i)})$

Nous allons à présent montrer que π est une loi stationnaire de la chaîne $x^{(1)}, \ldots, x^{(i)}$ par la propriété de *réversibilité des pas de Gibbs élémentaires*. Nous nous limitons à une présentation du cas discret, qui est celui qui nous concerne directement. Pour une présentation plus générale du cas continu, voir par exemple [Cappé et al., 2005].

Considérons un état x et notons y la variable simulée depuis x après un pas : y a les mêmes sous-vecteurs que x sauf un, par exemple le k-ième, qui est obtenu ainsi :

$$y_k \sim \pi_k(\bullet|x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)$$

On note Q le noyau de transition de la chaîne de Markov[12] :

$$Q(x, y) = \mathrm{P}(X^{(i+1)} = y|X^{(i)} = x)$$

La réversibilité consiste à montrer que :

$$\pi(x)Q(x, y) = \pi(y)Q(y, x) \tag{4.8}$$

Or la densité conditionnelle de la k-ième composante est, par définition :

$$\pi_k(y_k|x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n) = \frac{\pi(x_1, \ldots, x_{k-1}, y_k, x_{k+1}, \ldots, x_n)}{\sum_{x_k} \pi(x_1, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_n)}$$

[12] Rappelons que nous considérons à présent les pas élémentaires ; par conséquent les $X^{(i)}$ et $X^{(i+1)}$ dans la définition de Q ne sont pas les mêmes que ceux de la définition de l'algorithme, où nous considérions qu'une étape était une série de simulations successives, une pour chaque sous-vecteur, c'est-à-dire n pas élémentaires.

D'où la dérivation suivante :

$$\begin{aligned}
\pi(x)Q(x,y) &= \pi(x)\left(\left(\prod_{j\neq k}\mathbb{1}_{x_j}(y_j)\right)\pi_k(y_k|x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n)\right)\\
&= \pi(x)\left(\prod_{j\neq k}\mathbb{1}_{x_j}(y_j)\right)\frac{\pi(x_1,\ldots,x_{k-1},y_k,x_{k+1},\ldots,x_n)}{\sum_z \pi(x_1,\ldots,x_{k-1},z,x_{k+1},\ldots,x_n)}\\
&= \left(\prod_{j\neq k}\mathbb{1}_{y_j}(x_j)\right)\frac{\pi(x)\pi(y_1,\ldots,y_{k-1},y_k,y_{k+1},\ldots,y_n)}{\sum_z \pi(y_1,\ldots,y_{k-1},z,y_{k+1},\ldots,y_n)}\\
&= \left(\prod_{j\neq k}\mathbb{1}_{y_j}(x_j)\right)\frac{\pi(x_1,\ldots,x_{k-1},x_k,x_{k+1},\ldots,x_n)\pi(y)}{\sum_z \pi(y_1,\ldots,y_{k-1},z,y_{k+1},\ldots,y_n)}\\
&= \pi(y)\left(\prod_{j\neq k}\mathbb{1}_{y_j}(x_j)\right)\frac{\pi(y_1,\ldots,y_{k-1},x_k,y_{k+1},\ldots,y_n)}{\sum_z \pi(y_1,\ldots,y_{k-1},z,y_{k+1},\ldots,y_n)}\\
&= \pi(y)\left(\left(\prod_{j\neq k}\mathbb{1}_{y_j}(x_j)\right)\pi_k(x_k|y_1,\ldots,y_{k-1},y_{k+1},\ldots,y_n)\right)\\
&= \pi(y)Q(y,x)\,,
\end{aligned}$$

en utilisant le fait que lorsque les fonctions indicatrices sont toutes non nulles, on peut remplacer tous les x_j pour $j \neq k$ par le y_j correspondant. Ceci prouve la réversibilité de la chaîne.

En sommant (4.8) sur tous les x possibles, nous obtenons :

$$\begin{aligned}
\mathrm{P}(X^{(i+1)} = y) &= \sum_x \pi(x)Q(x,y)\\
&= \sum_x \pi(y)Q(y,x)\\
&= \pi(y)\,,
\end{aligned}$$

ce qui établit la stationnarité.

Une propriété intéressante de l'échantillonnage de Gibbs est que si nous savons marginaliser une partie des sous-vecteurs dont les valeurs ne nous intéressent pas directement, l'algorithme peut s'appliquer de la même façon. Ainsi, dans l'hypothèse où seulement une partie des sous-vecteurs, par exemple $X_1,\ldots,X_m$ avec $m < n$, nous intéresse et que nous savons marginaliser les autres $X_{m+1},\ldots,X_n$ pour simuler les lois conditionnelles de chacun des X_j connaissant $X_1,\ldots,X_{j-1},X_{j+1},\ldots,X_m$ (pour $j \in \{1,\ldots,m\}$) , il n'est pas nécessaire de simuler les $n-m$ dernières composantes : l'algorithme converge vers la loi jointe de $X_1,\ldots,X_m$. Cette astuce, qui permet de gagner en temps et parfois également en qualité d'inférence, comme nous le verrons en section 4.4.3, est connue sous le nom d'*échantillonnage de Gibbs rao-blackwellisé* [Robert and Casella, 1999].

Ces résultats ne disent malheureusement rien sur la rapidité de convergence. En pratique, il est nécessaire d'effectuer plusieurs milliers d'itérations et d'échantillonner à intervalles réguliers mais suffisamment grands pour que les échantillons ne soient pas trop corrélés.

4.4.2 Échantillonnage à partir des formules de l'EM

Nous nous plaçons toujours dans un cadre bayésien et par conséquent les paramètres sont considérés comme des variables aléatoires. L'application de l'échantillonnage de Gibbs nécessite donc de trouver un ensemble de variables pour lequel on sait simuler chacune connaissant les autres.

Ici, la manière la plus évidente de le faire est de faire appel aux équations de mise à jour issues de l'algorithme EM (4.2), (4.3), (4.4). Ainsi, itérativement, nous :

- tirons une indicatrice de thème dans l'ensemble $\{1, \ldots, n_T\}$ pour chaque document, conditionnellement aux paramètres α et β, selon une distribution multinomiale de paramètres les probabilités a posteriori que le document en question appartienne aux différents thèmes ;
- tirons de nouvelles valeurs pour α, β qui, conditionnellement aux indicatrices de thème, suivent des lois de Dirichlet ;
- calculons les nouvelles probabilités a posteriori selon (4.2).

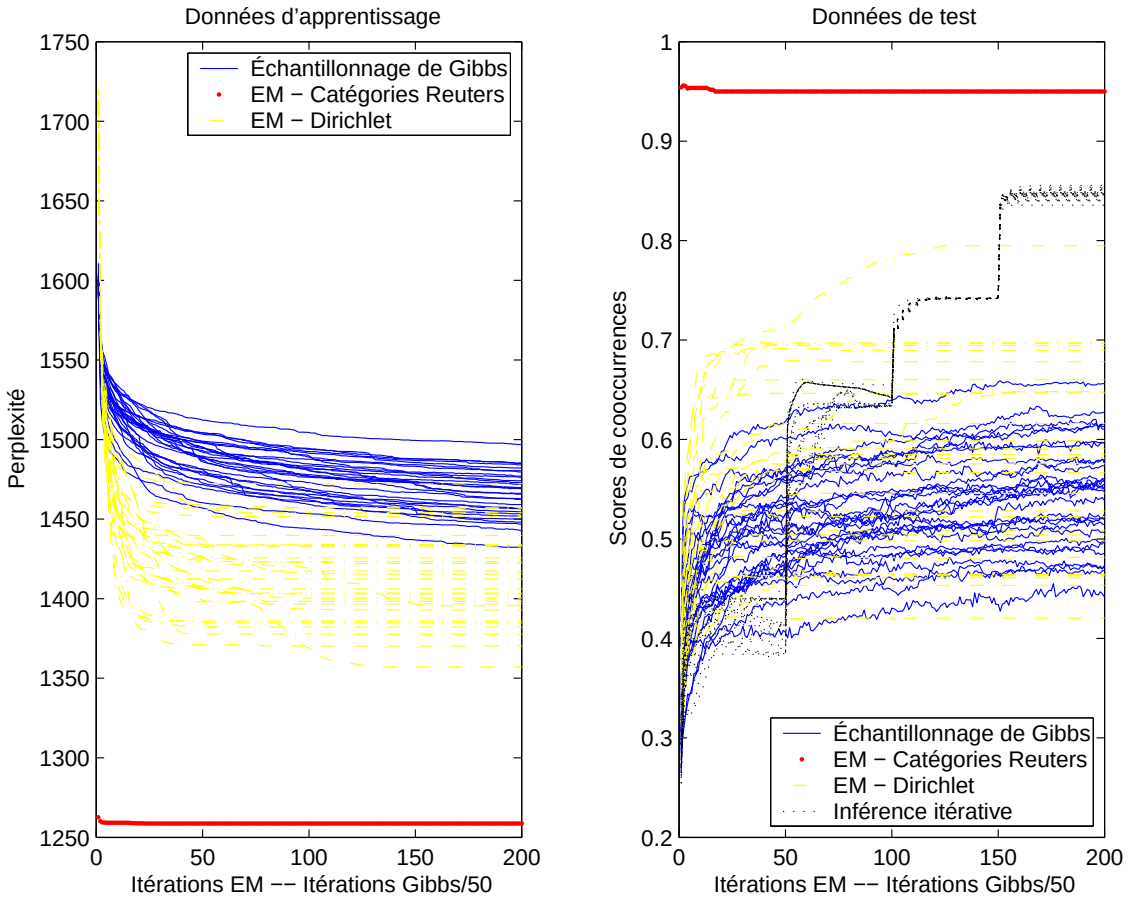

FIG. 4.11 – Perplexité sur l'apprentissage et scores de cooccurrences sur le test en fonction des itérations d'EM ou d'échantillonnage de Gibbs.

Nous représentons sur la figure 4.11 10000 itérations de l'échantillonneur de Gibbs avec 30 initialisations différentes sur un jeu fixé de données, avec toujours les courbes correspondant à l'EM de base et à la méthode d'inférence itérative de la section 4.3.2, à titre de comparaison. De nouveau, nous retrouvons une grande variabilité des performances d'un

essai sur l'autre et même parfois, à l'intérieur d'une même trajectoire. Ce comportement est typique des cas où l'échantillonneur de Gibbs ne remplit pas pleinement son objetif et reste bloqué, comme l'EM, dans des optimums locaux. En réalité, nous ne simulons pas tout-à-fait selon la distribution a posteriori réelle mais plutôt sur une restriction de celle-ci à un sous-ensemble réduit de l'espace des variables latentes et des paramètres. Les résultats en termes de perplexité et de scores de cooccurrences sont dans les mêmes ordres de grandeurs que ceux de l'algorithme EM, nettement inférieurs à ceux de la méthode d'inférence itérative.

4.4.3 Échantillonnage de Gibbs rao-blackwellisé

Nous remarquons à présent qu'il n'est en fait pas nécessaire de simuler les paramètres α et β puisqu'ils peuvent être intégrés dans l'équation conditionnelle (4.6). Cela nous permet d'appliquer un algorithme de Gibbs rao-blackwellisé.

Nous obtenons ainsi des estimations de la distribution des thèmes T des différents documents, en appliquant l'algorithme de Gibbs pour simuler tour à tour chaque variable latente T_d, conditionnellement aux affectations des autres documents. Remarquons que si le document d est de longeur 1, nous retrouvons l'échantillonnage de Gibbs décrit dans [Griffiths and Steyvers, 2002] pour le modèle LDA (en utilisant l'identité $\Gamma(a+1) = a\Gamma(a)$).

La figure 4.12 trace les scores de perplexité d'apprentissage et de scores de cooccurrences pour 30 initialisations aléatoires indépendantes, comparés aux mêmes références que dans la sous-section précédente. De nouveau, pour chaque initialisation, nous présentons 10000 itérations de l'échantillonneur de Gibbs. L'échantillonnage de Gibbs dépasse l'algorithme EM sur quasiment toutes les trajectoires. Ses performances sont dans le même intervalle que celles de la méthode itérative, bien que beaucoup plus variables (scores de cooccurrences entre 70% et 95%). Néanmoins l'allure des trajectoires suggère que l'échantillonneur de Gibbs n'est pas irréductible dans ce contexte et n'explore toujours que des optimums locaux. Indiquons enfin que la stratégie consistant à sélectionner les trajectoires sur la base de leur scores de perplexité sur l'ensemble d'apprentissage fonctionnerait très bien ici également, la corrélation étant similaire à celle de la figure 4.10.

Cette stratégie d'échantillonnage rao-blackwellisée est manifestement bien meilleure que la première, plus naïve, pas seulement du point de vue des scores, mais également du point de vue de la rapidité. Si nous supposons grossièrement que le calcul des paramètres de simulation prend approximativement le même temps dans les deux cas (en ayant pris soin de tabuler correctement la fonction Gamma), l'échantillonneur de Gibbs issu des formules de l'EM nécessite de générer $n_T + 1$ échantillons Dirichlet (de dimension respectives n_W et n_T), ce qui correspond à un total d'environ $n_W n_T$ lois Gammas à tirer dans l'étape M, puis n_D échantillonnages d'une distribution multinomiale de dimension n_T lors de l'étape E. En comparaison, l'échantillonnage de Gibbs rao-blackwellisé est réduit à la seconde partie, c'est-à-dire n_D tirages multinomiaux de dimension n_T. La différence n'est pas que théorique : sur un Pentium IV CPU 2.40GHz avec 1.00Gb de mémoire, le second algorithme tourne 20 fois plus vite.

Au vu des performances de l'échantillonneur de Gibbs rao-blackwellisé, l'idée de l'employer au sein d'un algorithme itératif avec augmentation progressive du vocabulaire paraît assez naturelle. Nous conservons donc les mêmes conditions d'expérience que dans la figure 4.9 (tailles de vocabulaire successives de 40, 120, 200 mots avant d'utiliser l'ensemble du vocabulaire), en remplaçant à chaque étape les 50 itérations d'EM par 2500 itérations d'échantillonnage de Gibbs, et présentons les scores de cooccurrences sur l'ensemble de

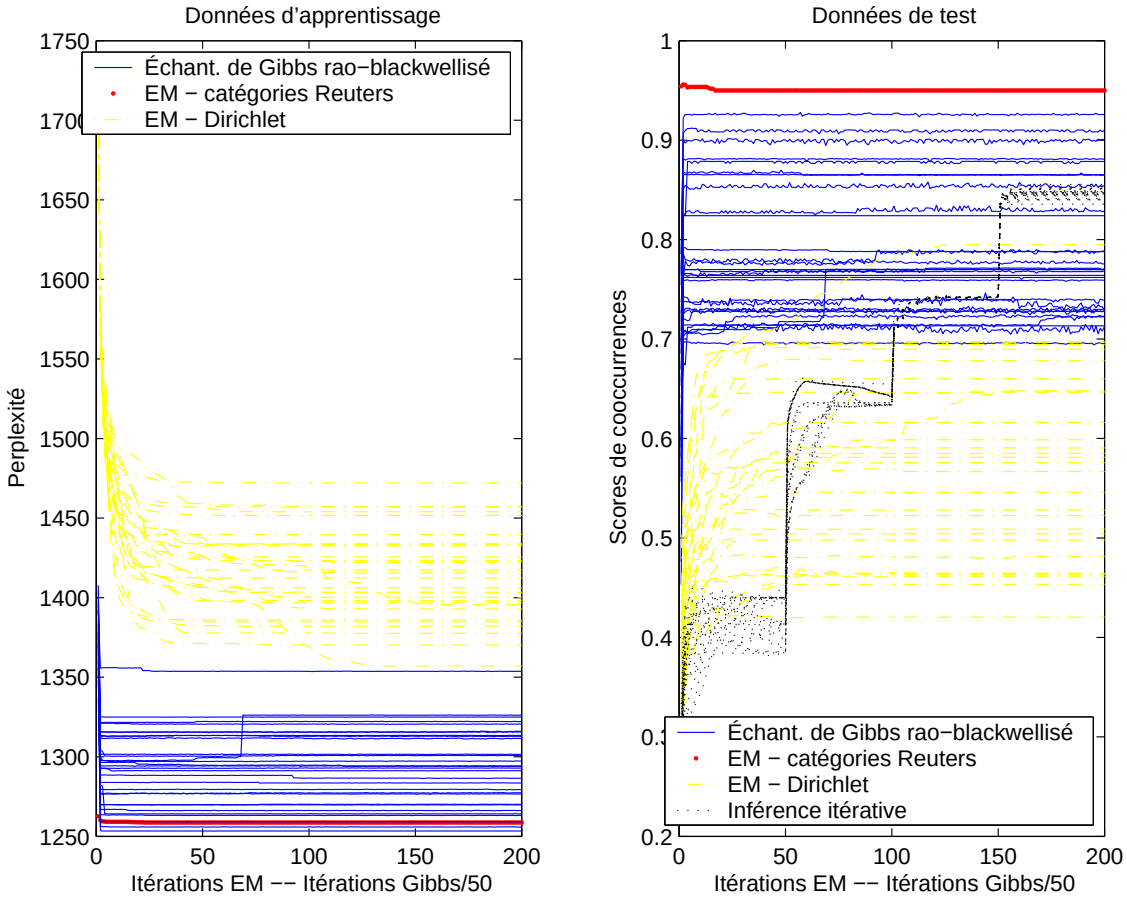

FIG. 4.12 – Perplexité sur l'apprentissage et scores de cooccurrences sur le test en fonction des itérations d'EM ou d'échantillonnage de Gibbs pour l'échantillonnage de Gibbs rao-blackwellisé.

test. Les résultats, présentés en figure 4.13, montrent que la technique d'inférence itérative est également efficace dans ce contexte, tant sur le plan de la stabilité que sur celui de la qualité des performances finales. Nous obtenons en effet un score très proche de 0.9 pour l'ensemble des 30 trajectoires. Il est intéressant de noter que, contrairement à ce que l'on observait avec l'algorithme EM, l'échantillonneur de Gibbs propose une certaine stabilité dans les performances dès les premières itérations sur vocabulaire très réduit. Il ne semble en effet y avoir que deux niveaux de scores de cooccurrences à l'itération 2500 et les différentes trajectoires, qu'elles viennent d'un mode ou de l'autre, atteignent rapidement ensuite des configurations très voisines en termes de scores de cooccurrences.

4.5 Généralisation des résultats

Dans cette section, nous testons les algorithmes sur d'autres corpus connus de classification de documents : Spamassassin [Mason, 2002], 20 newsgroups [Lang, 1995] et Reuters 21578[13] [Lewis, 1997] (voir [Vinot, 2004] par exemple pour une étude plus complète sur les corpus utilisés en classification supervisée).

[13]Nous avons conservé seulement les documents contenant une balise <BODY> et une classe unique.

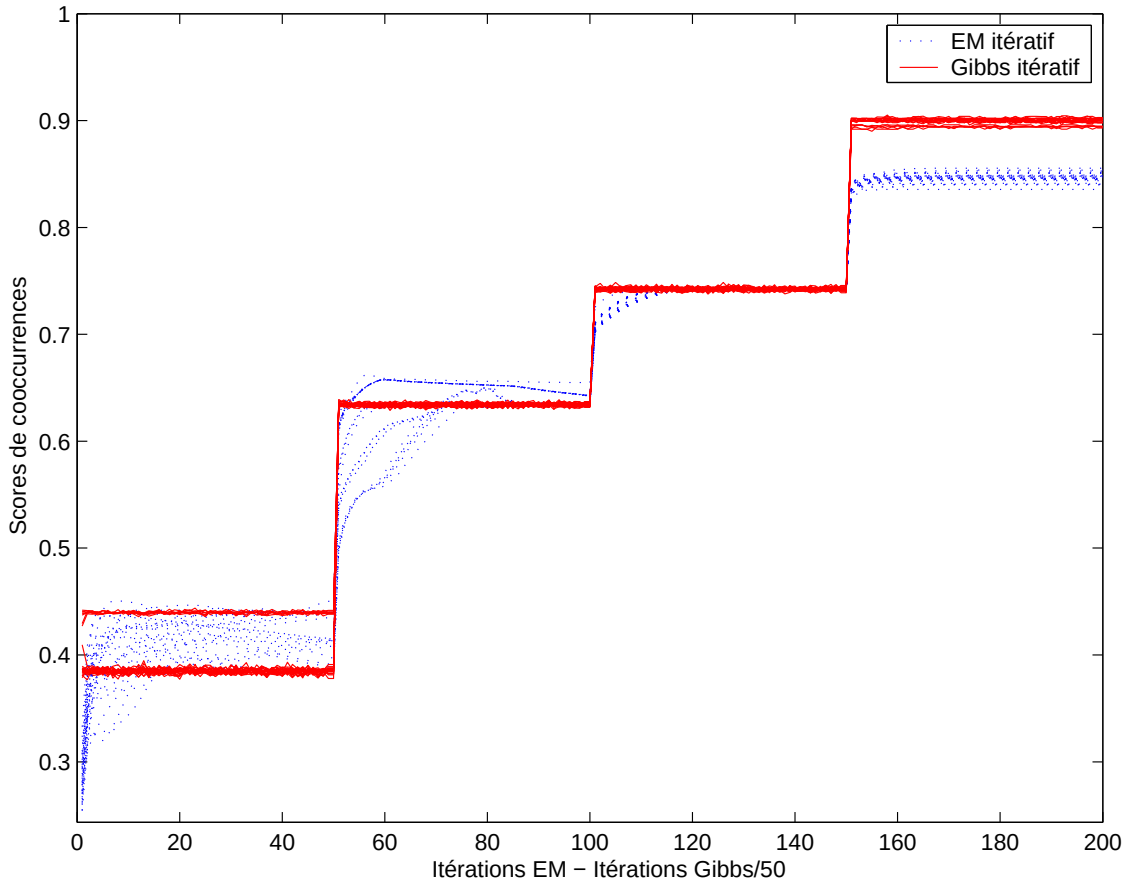

Fig. 4.13 – Comparaison de méthodes itératives fondées sur l'algorithme EM et sur l'échantillonneur de Gibbs par leurs scores de cooccurrences sur l'ensemble de test.

Ces nouveaux corpus restent assez similaires à notre corpus dans la mesure où la monothématicité est privilégiée, à défaut d'un moyen convaincant de gérer la multi-thématicité lors du processus de construction de la matrice de référence documents-catégories mais le fait que le nombre de classes soit différent et parfois plus grand rend les résultats, et en particulier les scores de cooccurrence, difficiles à comparer aux expériences précédentes.

Les résultats sont présentés en table 4.5. Nous avons mesuré les performances après 20 itérations d'EM, 40 itérations de l'algorithme itératif mis en œuvre en figure 4.9 (10 par taille de vocabulaire ; les coupes sont faites aux mêmes points en pourcentage du vocabulaire total) et 10000 itérations de l'échantillonneur de Gibbs. Pour Spamassassin, l'évaluation est conduite avec l'aide de la 10-validation croisée alors que pour les deux derniers corpus, nous n'avons lancé les algorithmes que sur un jeu de données[14].

Une des seules remarques restant valables dans toutes les expériences est que l'inférence itérative améliore significativement l'adéquation aux données, tout en réduisant la variance, par rapport à l'algorithme EM de base. Mais ces performances ne se traduisent pas nécessairement en meilleurs scores de cooccurrences. L'échantillonneur de Gibbs a également des résultats très variables pour les deux mesures. Nous faisons l'hypothèse que ses

[14]La raison en est que le temps de calcul s'allonge considérablement avec l'augmentation du nombre de thèmes.

Corpus	Méthode	Perplexité apprentissage	Cooccurrences test
Corpus Spamassassin			
6046 textes	EM (catégories)	1507.17±36.84	95.41%±0.93%
2 classes	EM (Dirichlet)	1404.82±52.25	80.34%±6.44%
4102548 occurrences	Inférence itérative	1372.59±39.25	81.06%±2.01%
178526 mots	Gibbs	1353.95±33.71	82.60%±0.94%
Corpus 20 newsgroups			
18828 textes	EM (catégories)	1251.79	70.73%
20 classes	EM (Dirichlet)	1366.84±22.24	35.68%±3.46%
4987889 occurrences	Inférence itérative	1272.51±7.11	38.12%±2.01%
113725 mots	Gibbs	1678.74±8.66	13.96±0.60%
Corpus Reuters 21578			
8654 textes	EM (catégories)	494.32	76.09%
65 classes	EM (Dirichlet)	466.45±3.35	38.66%±6.09%
868647 occurrences	Inférence itérative	427.20±2.67	23.80%±2.24%
22687 mots	Gibbs	716.64±34.47	54.26%±2.41%

TAB. 4.1 – Performances des méthodes d'inférence pour le modèle de mélange de multinomiales sur trois autres corpus.

mauvais scores de perplexité sur les deux derniers corpus sont dûs au fait que les classifications déterministes sont plus fortement pénalisées (du point de vue de la perplexité) lorsque le nombre de thèmes est grand.

Retenons également que, sur les trois corpus, l'initialisation sur les catégories obtient de bien meilleurs scores de coocurrences que toutes les autres. Nous pensons que ces résultats montrent que des phénomènes différents de ceux étudiés se produisent pour des grands nombres de thèmes ou des grandes tailles de vocabulaire. Cette hypothèse sera à confirmer par des travaux futurs dans ces contextes.

4.6 Cadres semi-supervisé et supervisé

Dans cette dernière sous-section, nous nous posons deux questions distinctes :

- Nous avons vu en section 4.2 combien l'algorithme EM est loin de la solution de classification obtenue en considérant l'ensemble des étiquettes pour l'initialisation, lorsque nous procédons à une initialisation *non informative*. Il est cependant possible que seule une petite partie des étiquettes soit nécessaires lors de l'apprentissage pour mettre l'algorithme « sur la voie ». C'est ce que nous vérifions dans la première sous-section avec un cadre semi-supervisé.
- Dans la seconde sous-section, nous évaluons l'apport de la formule (4.6) dans un cadre supervisé, pour améliorer les performances du classifieur bayésien naïf.

4.6.1 Ajout d'information de supervision

Nous nous proposons d'évaluer le nombre d'étiquettes nécessaire pour améliorer de façon significative la classification produite par l'algorithme EM, une expérience proche de

celle décrite dans l'article [Nigam et al., 2000]. Nous espérons ainsi combler la différence importante entre l'initialisation Dirichlet et l'initialisation catégories Reuters. Dans ce but, nous considérons un nombre limité de textes pour chaque classe (entre 1 et 50) pour l'initialisation de la matrice β : les fréquences de chaque thème sont obtenues en effectuant la moyenne des fréquences de chaque mot dans les textes tirés. Les appartenances de ces textes a un thème donné étant connues de façon certaine, l'algorithme EM est appliqué en mode *semi-supervisé*, c'est-à-dire que seules les probabilités a posteriori des textes dont les thèmes sont vraiment inconnus sont mises à jour (les autres sont naturellement fixées à 1 pour le thème « correct » et à 0 pour les autres thèmes). Pour chaque nombre de textes et chaque jeu, nous répétons l'expérience 10 fois pour compenser les risques de tirer des documents « non représentatifs » de leur thème.

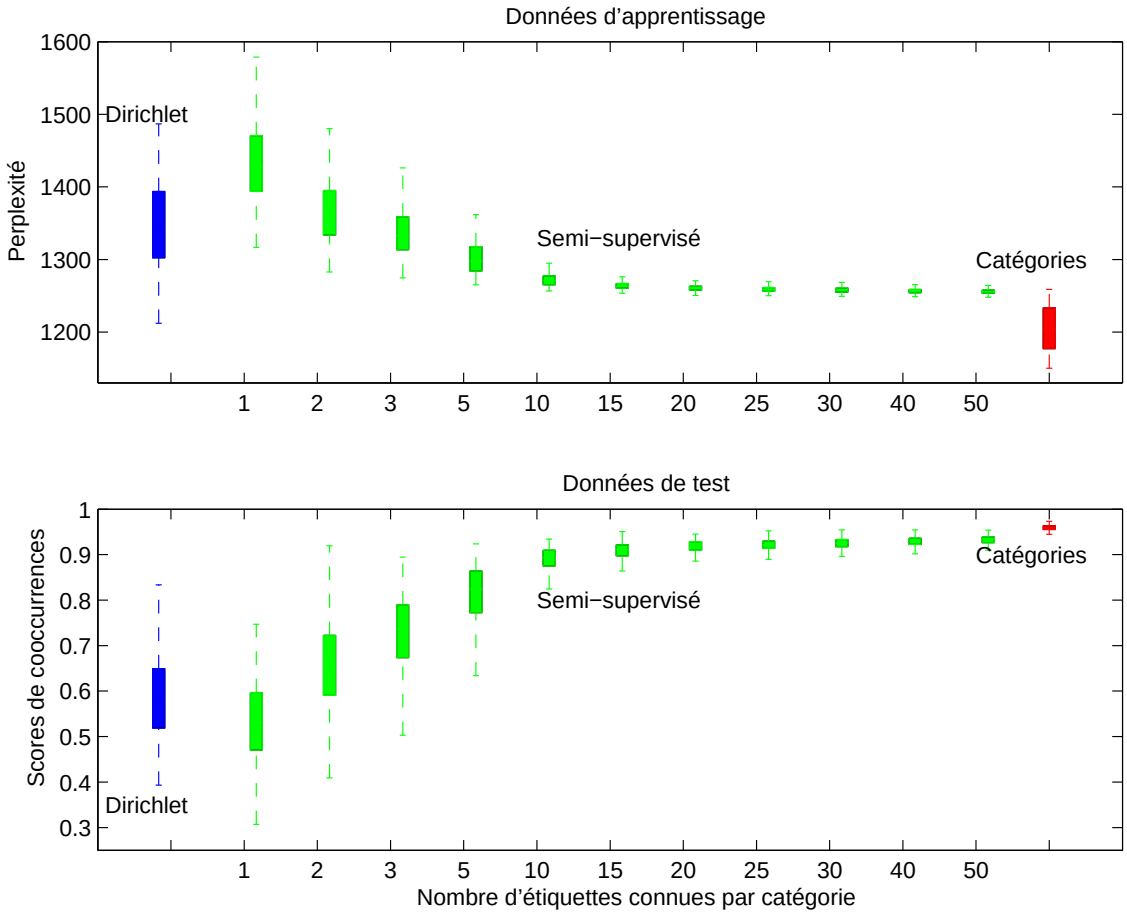

Fig. 4.14 – Perplexité et scores de cooccurrence en fonction du nombre d'étiquettes connues par thème en apprentissage semi-supervisé.

Le graphe inférieur de la figure 4.14 montre que, conformément à l'intuition, les résultats s'améliorent très vite avec la connaissance du nombre d'étiquettes par thème. De bons scores sont atteints relativement rapidement : avec 10 étiquettes par thème (soit 1.1% de l'ensemble d'apprentissage étiqueté), le score est déjà aux alentours de 0.7 (à comparer avec 0.8 lorsque 5.5% des catégories sont connues et à 0.9 lorsque toutes les étiquettes sont données).

Le graphe supérieur de la figure 4.14 confirme la première impression et suggère que

connaître 20 ou 50 étiquettes revient quasiment au même en termes de perplexité. Par conséquent, quelques pourcents de documents étiquetés suffisent à obtenir une bonne modélisation de la distribution des mots (perplexité) et à peine plus pour obtenir également de bonnes performances de classification.

4.6.2 Inférence supervisée

D'un point de vue bayésien, la règle discriminante (4.6) est mieux justifiée que la stratégie du classifieur bayésien naïf (4.5) couramment utilisée en classification supervisée de textes. Nous établissons dans cette section une comparaison empirique.

Pour utiliser (4.6) dans un cadre supervisé, nous supposons que T_{-d} sont les étiquettes au sein de l'ensemble d'apprentissage et T_d la classe à prédire pour un nouveau document non-étiqueté. En reprenant la formule (4.5) avec les mêmes notations (S_t devient $S_t - 1$ alors K_t et K_{wt} sont remplacés par respectivement K_t^{-d} et K_{wt}^{-d})), nous avons :

$$\mathrm{P}(T_d = t | T_{-d}, C) \propto (S_t - 1 + \lambda_\alpha - 1) \frac{\prod_{w=1}^{n_W} (K_{wt}^{-d} + \lambda_\beta - 1)^{C_{wd}}}{\Big(K_t^{-d} + n_W(\lambda_\beta - 1)\Big)^{l_d}}$$

En modifiant légèrement (4.6), grâce aux propriétés de la fonction Γ, il devient plus facile de comparer les deux formules :

$$\begin{cases} (S_t - 1 + \lambda_\alpha - 1) \dfrac{\prod_{w=1}^{n_W} (K_{wt}^{-d} + \lambda_\beta - 1)^{C_{wd}}}{\Big(K_t^{-d} + n_W(\lambda_\beta - 1)\Big)^{l_d}} & \text{(classifieur bayésien naïf);} \\ (S_t - 1 + \lambda_\alpha) \dfrac{\prod_{w=1}^{n_W} \prod_{i=0}^{C_{wd}-1} (K_{wt}^{-d} + \lambda_\beta + i)}{\prod_{i=0}^{l_d-1} (K_t^{-d} + n_W \lambda_\beta + i)} & \text{(approche entièrement bayésienne).} \end{cases}$$

En négligeant le décalage d'une unité sur les hyperparamètres (dû à la non coïncidence du mode et de l'espérance de la distribution Dirichlet), nous observons que les deux formules sont approximativement équivalentes si :

1. Tous les comptes sont égaux à 0 ou 1 ($\prod_{i=0}^{C_{wd}-1}(K_{wt}^{-d} + \lambda_\beta + i)$ se simplifierait alors en $(K_{wt}^{-d} + \lambda_\beta)^{C_{wd}}$)
2. La longueur d'un document est négligeable devant $K_t^{-d} + n_W \lambda_\beta$ (alors $\prod_{i=0}^{l_d-1}(K_t^{-d} + n_W \lambda_\beta + i) \approx (K_t^{-d} + n_W \lambda_\beta)^{l_d}$)

Pour vérifier les conséquences pratiques sur les performances, nous avons à nouveau utilisé notre corpus issu de Reuters (cf. section 4.2.1) et pratiqué l'évaluation à l'aide d'une 10-validation croisée. Pour observer l'évolution du taux d'erreur dans les deux cas en modifiant la valeur des hyperparamètres, nous posons :

$$\begin{cases} \lambda_\alpha - 1 = 1 \text{ et } \lambda_\beta - 1 = \lambda & \text{pour le classifieur bayésien naïf} \\ \lambda_\alpha = 1 \text{ et } \lambda_\beta = \lambda & \text{dans l'autre cas} \end{cases},$$

et jouons sur la valeur de λ. Sur la figure 4.15, λ est porté sur l'axe des abscisses.

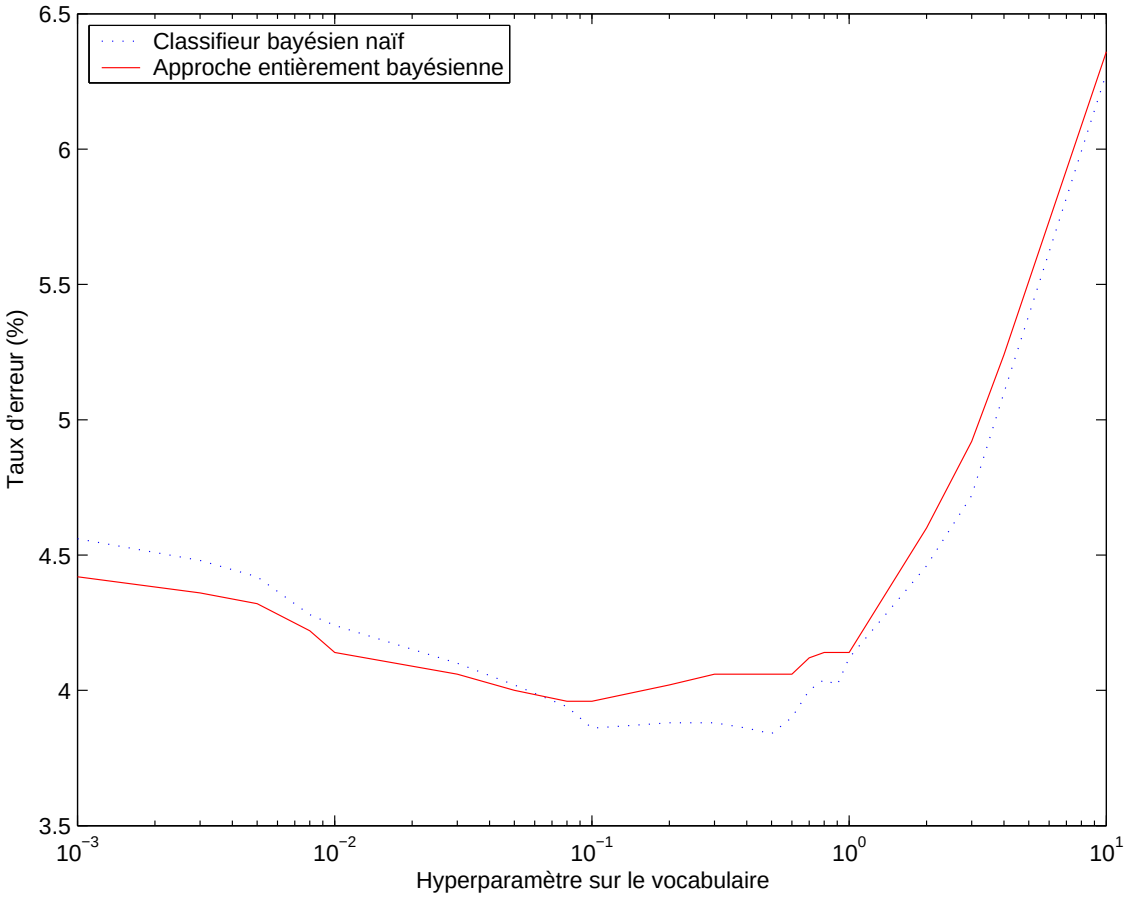

Fig. 4.15 – Évolution du taux d'erreur en fonction de l'hyperparamètre λ_β.

Les résultats sont remarquablement proches entre les deux approches. Le classifieur bayésien naïf semble dominer pour les grandes valeurs de l'hyperparamètre de lissage alors que l'approche entièrement bayésienne paraît avoir l'avantage pour les valeurs les plus faibles. Cela dit, la plus grande différence entre les scores des deux méthodes est d'environ 0.2%, ce qui ne représente qu'un seul document dans notre corpus de test de 500 documents à chaque jeu de données et n'est donc pas statistiquement significatif.

Nous avons également conduit des tests sur d'autres corpus de référence en classification supervisée, à savoir 20-Newsgroups [Lang, 1995] et Spam Assassin [Mason, 2002], en modifiant notamment le nombre de documents d'apprentissage ou la taille du vocabulaire mais la différence ne devient jamais plus significative. En outre, des études poussées sur la classification supervisée de texte [Yang and Liu, 1999, Sebastiani, 2002] montrent que le classifieur bayésien naïf est de toute façon systématiquement moins performant que les meilleures méthodes. L'avantage probant qu'apporte la formule (4.6) dans le cas de l'échantillonnage de Gibbs ne se retrouve donc pas dans le cadre supervisé.

Chapitre 5

Discussion des performances

Après les études du chapitre précédent concernant uniquement le modèle de mélange de multinomiales et l'inférence de ses paramètres, nous le comparons à présent à d'autres approches, en utilisant toujours le même cadre d'évaluation.

Dans un premier temps, nous testons l'algorithme classique des K-moyennes avec la représentation la mieux adaptée à la fouille de texte. Puis nous reprenons le modèle probabiliste LDA, introduit en section 2.3.4. Nous détaillons dans un premier temps différentes possibilités pour contourner les difficultés théoriques liées à la complexité de la structure. Nous présentons ensuite les résultats obtenus par échantillonnage de Gibbs.

Enfin, dans une dernière partie, nous abordons le problème difficile de l'interprétation a posteriori des regroupements induits, à partir des distributions thématiques sur le vocabulaire. Nous privélégions la piste d'identification de termes représentatifs, en utilisant des méthodes statistiques.

5.1 Comparaison avec l'algorithme des K-moyennes

La description de l'algorithme des K-moyennes figure dans la section 2.2.1. Dans un premier temps, nous travaillons avec la représentation des comptes de mots et la distance cosinus classique (avec normalisation par les normes euclidiennes).

Comme dans le cas de l'EM, nous choisissons d'initialiser l'algorithme des K-moyennes sur une classification particulière, c'est-à-dire une configuration des probabilités de chaque document d'appartenir à un thème donné, pour éluder la difficulté de proposer des positions initiales des centroïdes. En revanche, contrairement au mode d'initialisation de l'EM, pour les K-moyennes, il faut que le partitionnement choisi soit déterministe. C'est déjà le cas pour l'initialisation sur les catégories Reuters. Pour l'initialisation Dirichlet en revanche, nous devons, après tirage, convertir les probabilités d'appartenance en 0 ou 1, en ne conservant pour chaque document que le thème le plus probable.

Nous présentons le résultat de cette première série d'expériences (vecteurs de comptes non modifiés et distance cosinus) en figure 5.1. Par souci de lisibilité, pour cette figure comme pour les deux autres de cette section, nous ne montrons que les 15 premières itérations. Les suivantes ne présentent pas de changements importants sur ce qui nous intéresse, à savoir le positionnement des différentes courbes les unes par rapport aux autres. Nous choisissons ici la mesure des cooccurrences sur l'ensemble de test. L'algorithme des K-moyennes n'étant pas un algorithme probabiliste, il n'y a en effet pas de façon naturelle de calculer sa perplexité.

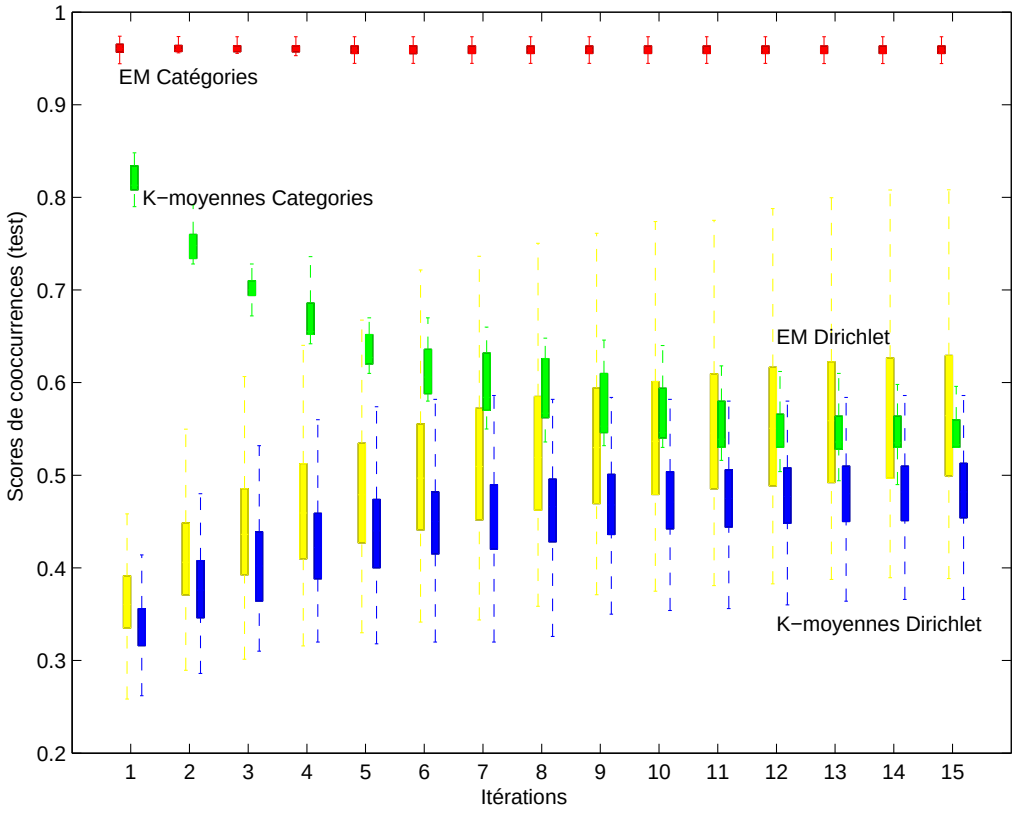

Fig. 5.1 – Scores de cooccurrences des K-moyennes sans pondération idf.

Nous observons sur la figure 5.1 que l'algorithme des K-moyennes appliqué de la sorte donne de mauvais résultats sur notre corpus, relativement aux performances de référence de l'algorithme EM initialisé sur les catégories Reuters. Ainsi, les performances à partir de l'initialisation Dirichlet sont globalement dans la fourchette inférieure (autour de 0.5) de celles réalisées par l'algorithme EM. Notons également que le constat réalisé pour l'algorithme EM reste vrai pour les K-moyennes : les performances présentent une variabilité importante d'une initialisation à une autre. Il est enfin particulièrement révélateur de constater que l'algorithme initialisé sur les catégories Reuters, en partant d'un point satisfaisant, subit une dégradation progressive de la classification pour atteindre en fin de compte un niveau inférieur ou comparable à l'algorithme EM avec initialisation Dirichlet. Le fait que des textes placés au départ dans une « bonne » configuration soient déplacés vers des groupes moins pertinents ne peut conduire qu'à une conclusion : la représentation choisie n'est pas révélatrice des similarités que nous cherchons à mettre en évidence.

Cette remarque incite donc à changer de représentation et nous appliquons à présent l'algorithme des K-moyennes avec la distance cosinus normalisée aux données représentées sous la forme TFIDF, consistant à pondérer chaque mot par sa « rareté » dans le corpus (plus le mot est présent dans de nombreux documents, plus son score de TFIDF est diminué, section 2.2.2).

L'amélioration des performances est spectaculaire, comme en témoigne la figure 5.2. Après quelques itérations, l'algorithme des K-moyennes avec initialisation Dirichlet devient en effet meilleur en moyenne que l'EM avec la même initialisation. La variabilité est légèrement plus forte mais les performances sont supérieures de 10%, ce qui montre

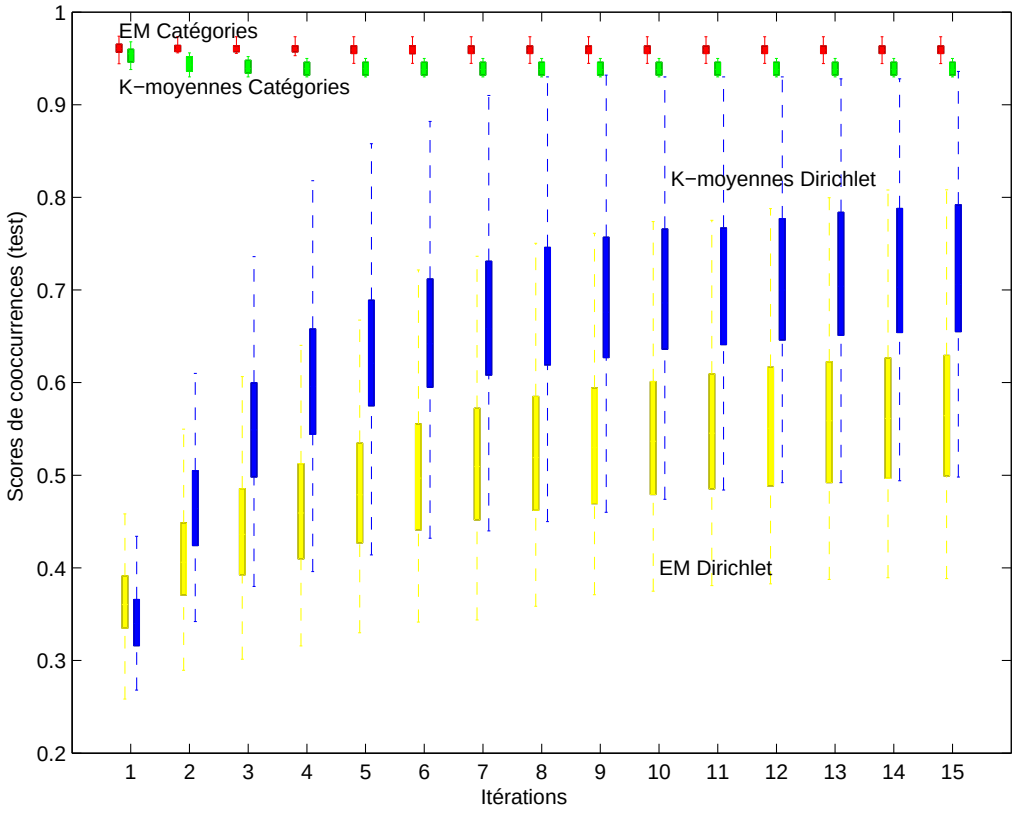

FIG. 5.2 – Scores de cooccurrences des K-moyennes avec pondération idf.

bien la pertinence de cette représentation pour notre corpus. Il faut, à ce sujet, signaler que ces expériences sont légèrement biaisées en faveur des algorithmes de partitionnement déterministe, comme les K-moyennes, dans la mesure où le classement Reuters de référence est lui-même déterministe. L'algorithme EM en revanche sera forcément pénalisé dès lors qu'il assigne un document à deux thèmes, même lorsque cette décision est justifiée... Sur des corpus présentant des recrouvrements partiels des thèmes, les conclusions seraient probablement moins tranchées.

Le terme de fréquence inversée sur les documents a ici principalement l'influence d'une technique de lissage, en diminuant l'importance des termes ayant de nombreuses occurrences (principalement les anti-mots). Il est donc tout-de-même assez remarquable que l'algorithme EM, sans mécanisme de correction similaire, atteigne des performances qui ne soient que modérément inférieures. Nous avons montré en section 4.2.4 que, dans la mesure où les probabilités a posteriori sont presque déterministes, l'algorithme EM est proche, dans ce contexte, d'un algorithme des K-moyennes avec une divergence de Kullback-Leibler (c'est-à-dire qu'il est équivalent de ne considérer que le thème le plus probable pour chaque document). Plutôt que deux algorithmes, cette expérience consiste donc à comparer deux représentations, la représentation TFIDF avec distance cosinus et la représentation des documents en termes de profils probabilistes (sans facteur de lissage réduisant l'influence des mots fréquents).

Pour terminer cette section, nous comparons les performances de l'algorithme des K-moyennes à celles de l'algorithme d'échantillonnage de Gibbs rao-blackwellisé étudié en section 4.4.3. Nous constatons, sur la figure 5.3, que les performances de l'échantillonneur

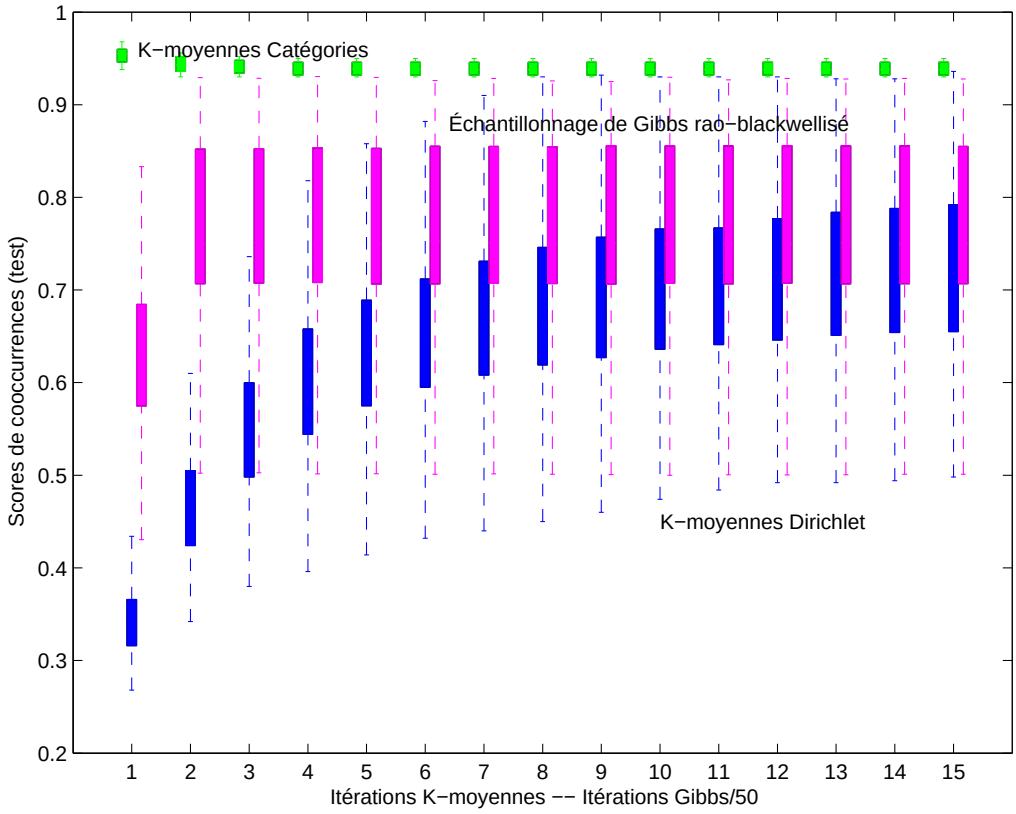

Fig. 5.3 – Comparaison de l'échantillonnage de Gibbs rao-blackwellisé et des K-moyennes avec pondération idf.

de Gibbs sont légèrement meilleures que celles de l'algorithme des K-moyennes, même si elles deviennent assez proches après convergence de ce dernier (la conduite d'un nombre supérieur d'itérations, non représentée sur la figure, n'infirme pas la supériorité de l'échantillonneur de Gibbs). Si les performances médianes diffèrent, il convient néanmoins de noter que la dispersion est à peu près équivalente. Par conséquent, sur un essai particulier, l'utilisation de l'échantillonneur de Gibbs ne garantit pas avec certitude une meilleure classification.

Nous ne menons pas ici de comparaison avec la méthode d'inférence itérative de la section 4.3.2, mais il est évident d'après les résultats précédents que ses performances sont meilleures et plus stables que celles de l'algorithme des K-moyennes. Cela dit, des expériences additionnelles sur l'algorithme des K-moyennes montrent de légères améliorations en réduisant la taille du vocabulaire, ce qui laisse supposer que, de la même façon que nous l'avons appliquée à l'EM et à l'échantillonneur de Gibbs, la méthode itérative donnerait aussi de bons résultats sur l'algorithme des K-moyennes.

5.2 Allocation Dirichlet latente

Nous revenons, dans cette section, sur le modèle LDA (présenté en section 2.3.4), qui est probablement le plus étudié des modèles probabilistes pour la fouille de textes. Avant une présentation plus formelle, rappelons que LDA vise à assouplir le cadre du modèle

de mélange de multinomiales (chapitre 4), en proposant que l'association non-déterministe entre thèmes et documents soit rapportée au niveau des occurrences : *ce sont donc les occurrences qui sont ventilées par thème*, et non plus les documents. Les différentes occurrences au sein d'un même texte restent toutefois globalement liées par une variable latente qui contrôle la distribution des thèmes au niveau du document.

Cette partie est organisée comme suit. Dans un premier temps, nous revenons sur des aspects théoriques du modèle, tels que le calcul de la vraisemblance complète, qui n'a pas d'utilité pratique en soi mais qui se révèle essentiel pour l'objectif abordé dans la section suivante : l'estimation des paramètres du modèle par échantillonnage de Gibbs. Cette étape permet d'obtenir la matrice β mais pas les valeurs de la vraisemblance et des probabilités a posteriori d'un document d'appartenir à un thème. Alors que ces dernières se déduisent trivialement des paramètres avec le modèle de mélange de multinomiales, l'application de nouvelles simulations est ici nécessaire, comme nous le détaillons en section 5.2.3. La dernière section est consacrée à une évaluation pratique, en comparaison avec le modèle de mélange de multinomiales.

5.2.1 Calcul de la vraisemblance *complète*

Nous conservons les notations introduites en section 2.3.4, complétées des définitions suivantes. Si T_{di} est le thème associé à l'occurrence i du document d, $T = (T_{11}, \ldots, T_{n_D l_{n_D}})$ désigne une configuration de variables indicatrices de thèmes pour l'ensemble des occurrences du corpus. T étant connue, notons N_{td} le nombre d'occurrences de d qui sont assignées au thème t ; de manière équivalente, K^d_{tw} désigne le nombre d'occurrences du mot w assignées au thème t dans le document d, $K_{tw} = \sum_{d=1}^{n_D} K^d_{tw}$ la même quantité pour l'ensemble du corpus et $K_t = \sum_{w=1}^{nW} K_{tw}$ le nombre total d'occurrences dans le thème t.

Pour un document d donné, conditionnellement au vecteur μ_d et à la matrice β, la probabilité jointe de deux vecteurs de longueur l_d, l'un, W_d, contenant les indices de mots et l'autre, T_d, les indicatrices de thème, s'exprime alors par (les occurrences étant indépendantes) :

$$\begin{aligned}
\mathrm{P}(W_d, T_d|\mu_d, \beta) &= \prod_{i=1}^{l_d} \mathrm{P}(W_{di}, T_{di}|\mu_d, \beta) \\
&= \prod_{i=1}^{l_d} \mathrm{P}(T_{di}|\mu_d, \beta)\, \mathrm{P}(W_{di}|T_{di}, \mu_d, \beta) \\
&= \prod_{i=1}^{l_d} (\mu_{dT_{di}} \beta_{T_{di}W_{di}}) \\
&= \prod_{t=1}^{n_T} \left(\mu_{dt}^{N_{td}} \prod_{w=1}^{n_W} \beta_{tw}^{K^d_{tw}} \right)
\end{aligned}$$

Les documents étant également supposés indépendants, la vraisemblance du corpus et des indicatrices latentes de thèmes s'exprime comme le produit :

$$P(W, T|\mu, \beta) = \prod_{d=1}^{n_D} \mathrm{P}(W_d, T_d|\mu_d, \beta)$$

L'observation principale de [Griffiths and Steyvers, 2002] est qu'il est possible d'intégrer cette vraisemblance sous la loi *a priori* des paramètres μ et β. Homogénéisons ici la

présentation de LDA en section 2.3.4 et la présentation bayésienne du modèle de mélange de multinomiales (section 4.1.2) en supposant que les distributions a priori de μ et β sont des Dirichlet de paramètres respectifs $(\lambda_\mu, \ldots, \lambda_\mu)$ avec $\lambda_\mu > 0$ et $(\lambda_\beta, \ldots, \lambda_\beta)$ avec $\lambda_\beta > 0$, vecteurs constants de longueurs n_T et n_W. Nous obtenons alors :

$$
\begin{aligned}
\mathrm{P}(W,T) &= \int_\beta \int_\mu \mathrm{P}(W|T,\beta) p(T,\mu,\beta) d\mu d\beta \\
&= \int_\beta \int_\mu \left(\prod_{d=1}^{n_D} \prod_{i=1}^{l_d} \mathrm{P}(W_{di}|T_{di}, \beta_{T_{di}}) \right) \left(\prod_{d=1}^{n_D} \prod_{i=1}^{l_d} \mathrm{P}(T_{di}|\mu_d) \right) \\
&\quad \left(\prod_{t=1}^{n_T} p(\beta_t|\lambda_\beta) \right) \left(\prod_{d=1}^{n_D} p(\mu_d|\lambda_\alpha) \right) d\mu d\beta \\
&= \int_\beta \int_\mu \left(\prod_{d=1}^{n_D} \prod_{i=1}^{l_d} \beta_{T_{di} W_{di}} \right) \left(\prod_{d=1}^{n_D} \prod_{i=1}^{l_d} \mu_{d T_{di}} \right) \\
&\quad \left(\prod_{t=1}^{n_T} \frac{\Gamma(\sum_{w=1}^{n_W} \lambda_\beta)}{\prod_{w=1}^{n_W} \Gamma(\lambda_\beta)} \prod_{w=1}^{n_W} \beta_{tw}^{\lambda_\beta - 1} \right) \left(\prod_{d=1}^{n_D} \frac{\Gamma(\sum_{t=1}^{n_T} \lambda_\alpha)}{\prod_{t=1}^{n_T} \Gamma(\lambda_\alpha)} \prod_{t=1}^{n_T} \mu_{dt}^{\lambda_\alpha - 1} \right) d\mu d\beta \\
&= \prod_{t=1}^{n_T} \left(\frac{\Gamma(n_W \lambda_\beta)}{\Gamma(\lambda_\beta)^{n_W}} \int_\beta \prod_{w=1}^{n_W} \beta_{tw}^{K_{tw} + \lambda_\beta - 1} d\beta \right) \prod_{d=1}^{n_D} \left(\frac{\Gamma(n_T \lambda_\alpha)}{\Gamma(\lambda_\alpha)^{n_T}} \int_\mu \prod_{t=1}^{n_T} \mu_{dt}^{N_{dt} + \lambda_\alpha - 1} d\mu \right) \\
&= \prod_{t=1}^{n_T} \left(\frac{\Gamma(n_W \lambda_\beta)}{\Gamma(\lambda_\beta)^{n_W}} \frac{\prod_{w=1}^{n_W} \Gamma(K_{tw} + \lambda_\beta)}{\Gamma(K_t + n_W \lambda_\beta)} \right) \prod_{d=1}^{n_D} \left(\frac{\Gamma(n_T \lambda_\alpha)}{\Gamma(\lambda_\alpha)^{n_T}} \frac{\prod_{t=1}^{n_T} \Gamma(N_{dt} + \lambda_\alpha)}{\Gamma(l_d + n_T \lambda_\alpha)} \right)
\end{aligned}
$$

Les deux facteurs de ce produit correspondent respectivement à $\mathrm{P}(W|T)$ et à $\mathrm{P}(T)$.

5.2.2 Estimation du modèle

Estimer le modèle LDA consiste à déterminer les valeurs de β à partir des observations. Plusieurs techniques d'estimation ont été proposées : l'inférence variationnelle, dans [Blei et al., 2002] ; la méthode espérance-propagation [Minka and Lafferty, 2002] et l'utilisation d'un échantillonneur de Gibbs [Griffiths and Steyvers, 2002]. Suivant les recommandations convergentes de [Griffiths and Steyvers, 2004, Buntine and Jakulin, 2004], c'est cette dernière méthode que nous avons étudiée. Nous reviendrons brièvement sur les alternatives en section 5.2.4.

Le principe général consiste à construire une séquence de configurations d'indicatrices de thèmes dont la loi stationnaire soit $\mathrm{P}(T|W)$. Pour cela, partant d'une configuration aléatoire, on modifie itérativement les assignations thématiques T_{di} de chaque occurrence du corpus en simulant sous la loi conditionnelle

$$
\mathrm{P}(T_{di}|T_{-di}, W) = \frac{\mathrm{P}(T,W)}{\mathrm{P}(T_{-di}, W_{-di})} \frac{1}{\mathrm{P}(W_{di}|T_{-di} W_{-di})}
$$

où T_{-di} désigne le vecteur T privé de l'élément T_{di}. Le second facteur de l'équation ci-dessus ne dépend pas de T_{di} et peut donc être vu comme un facteur de normalisation. Notons avec un exposant $(-di)$ les quantités K et N obtenues en retirant l'itération i du document d

(qui a pour thème T_{di}). Par définition,

$$\begin{aligned}
\forall t \neq T_{di}, K^{(-di)}_{tW_{di}} = K_{tW_{di}} \quad &\text{et} \quad K^{(-di)}_{T_{di}W_{di}} = K_{T_{di}W_{di}} - 1 \\
\forall t \neq T_{di}, K^{(-di)}_{t} = K_{t} \quad &\text{et} \quad K^{(-di)}_{T_{di}} = K_{T_{di}} - 1 \\
\forall t \neq T_{di}, N^{(-di)}_{dt} = N_{dt} \quad &\text{et} \quad N^{(-di)}_{dT_{di}} = N_{dT_{di}} - 1
\end{aligned}$$

Comme le montre [Griffiths and Steyvers, 2002], en utilisant l'expression précédente de la vraisemblance complète pour (W,T) et (W_{-di},T_{-di}) dans un calcul analogue à celui de la section 4.1.4 pour le modèle de mélange de multinomiales, nous obtenons, après simplifications des termes indépendants de t :

$$\forall t = 1,\ldots,n_T, \mathrm{P}(T_{di}=t|T_{-di},W) \propto \frac{(K^{(-di)}_{tW_{di}}+\lambda_\beta)}{(K^{(-di)}_{t}+n_W\lambda_\beta)}\frac{(N^{(-di)}_{dt}+\lambda_\alpha)}{(l_d-1+n_T\lambda_\alpha)} \tag{5.1}$$

$$\propto \frac{(K^{(-di)}_{tW_{di}}+\lambda_\beta)(N^{(-di)}_{dt}+\lambda_\alpha)}{K^{(-di)}_{t}+n_W\lambda_\beta} \tag{5.2}$$

Cette expression peut être retrouvée plus simplement, en utilisant le fait que T_{di} ne dépend des autres observations qu'à travers les indicatrices de thèmes des autres mots du même document. Il vient alors :

$$\begin{aligned}
\mathrm{P}(T_{di}|T_{-di},W) &= \frac{\mathrm{P}(T_{di},W_{di}|T_{-di},W_{-di})}{\mathrm{P}(W_{di}|T_{-di},W_{-di})} \\
&\propto \mathrm{P}(T_{di},W_{di}|T_{-di},W_{-di}) \\
&\propto \mathrm{P}(W_{di}|T_{di},T_{-di},W_{-di})\,\mathrm{P}(T_{di}|T_{-di},W_{-di}) \\
&\propto \mathrm{P}(W_{di}|T,W_{-di})\,\mathrm{P}(T_{di}|T_{-di}) \qquad (5.3)
\end{aligned}$$

Chaque terme du produit (5.1) peut être vu comme un estimateur des probabilités impliquées dans (5.3). Ainsi $\mathrm{P}(T_{di}|T_{-di})$ est estimé par le nombre de fois où le thème T_{di} a été vu dans le document d (non compris l'occurrence di), auquel s'ajoute un terme d'*a priori*. Après renormalisation par la longueur du document tronqué de l'occurrence courante (l_d-1), on retrouve un estimateur habituel pour $\mathrm{P}(T_{di}|T_{-di})$:

$$\frac{N_{dT_{di}}-1+\lambda_\alpha}{l_d-1+n_T\lambda_\alpha}$$

Le même argument s'applique pour l'occurrence W_{di} : connaissant les variables latentes de thèmes de tous les autres mots du corpus, $\mathrm{P}(W_{di}|T,W_{-di})$ s'estime simplement par :

$$\frac{K_{T_{di}W_{di}}-1+\lambda_\beta}{K_{T_{di}}-1+n_W\lambda_\beta}$$

L'algorithme d'estimation consiste à faire évoluer l'échantillonneur selon (5.2) à partir d'une configuration initiale, puis à collecter des valeurs des paramètres β. Nous avons choisi d'utiliser telles quelles les valeurs de β obtenues. Une autre stratégie est de moyenner l'ensemble des valeurs échantillonnées, dans le but d'obtenir un estimateur plus robuste. Ce moyennage est toutefois problématique, car il est en théorie possible qu'au fil des simulations, les thèmes soient renumérotés. En principe, il serait donc nécessaire d'apparier au

préalable les numérotations des thèmes de deux échantillon M et M' distincts, par exemple par la méthode hongroise décrite en section 3.5.3. En pratique, cependant, l'échantillonneur ne réalise jamais de permutation de thèmes, ce qui montre d'autant plus que son comportement théorique d'exploration complète de l'espace ne se retrouve pas pour notre application. [Pritchard et al., 2000] note à ce sujet que si l'échantillonneur de Gibbs fonctionnait parfaitement, les statistiques moyennées ne seraient d'aucune utilité puisque tous les termes apparaîtraient équiprobables par permutation (notamment, toutes les lignes de la matrice β seraient exactement identiques). On peut dire en forçant un peu le trait que, paradoxalement, l'échantillonneur de Gibbs est d'autant plus intéressant ici que son comportement s'éloigne de celui prédit par la théorie !

5.2.3 Calcul de la vraisemblance, classification des documents

Dans cette section, nous supposons les paramètres connus et nous nous intéressons plus directement à deux questions liées à la tâche initiale de construction d'une classification des documents :

- quelle est la vraisemblance d'un document ?
- comment calculer la distribution des thèmes associée à un document ?

[Griffiths and Steyvers, 2004] mentionne une méthode s'appuyant sur la technique de l'échantillonnage préférentiel, mais ne détaille pas sa mise en œuvre. [Minka and Lafferty, 2002] propose une méthode fondée sur l'algorithme *espérance-propagation*. Enfin, [Buntine and Jakulin, 2004] suggère des méthodes d'échantillonnage d'importance consistant à moyenner la quantité d'intérêt sur différents tirages, une approche assez voisine de celles que nous développons dans cette section.

5.2.3.1 Calcul de la vraisemblance

Considérons le document d et supposons que la distribution de thèmes μ_d pour ce document soit connue. Il vient :

$$\mathrm{P}(W_d|\mu_d) \quad = \quad \prod_{w=1}^{n_W} (\sum_{t=1}^{n_T} \mu_{dt}\beta_{tw})^{C_{wd}}$$

Notons en passant que cette forme a l'intérêt de suggérer un modèle génératif alternatif à celui que nous avons présenté, consistant en n_D tirages Dirichlet pour les μ_d suivis, pour chaque document, de l_d tirages selon une multinomiale de paramètre $\sum_{t=1}^{n_T} \mu_{dt}\beta_{tw}$ [Buntine, 2002]. On remarque que $\log \mathrm{P}(W_d|\mu_d)$ est une fonction concave des μ_{dt} : après reparamétrisation par $\mu_{d,1}, \ldots \mu_{d,n_T-1}$, il vient en effet :

$$\log \mathrm{P}(W_d|\mu_d) = \sum_{w=1}^{n_W} C_{wd} \log \left(\sum_{t=1}^{n_T-1} \mu_{dt}\beta_{tw} + (1 - \sum_{t'=1}^{n_T-1} \mu_{dt'})\beta_{n_T w}) \right)$$

dont le Hessien H est semi-defini négatif[1]. Cette fonction atteint donc un maximum unique sur le simplexe dans lequel évolue $\mu_d = (\mu_{d1}, \ldots, \mu_{dn_T})$.

[1] Plus précisément, pour tout vecteur $u = (u_1, \ldots, u_{n_T-1})$ non nul,

$$u^T H u = -\sum_{w=1}^{n_W} C_{wd} \frac{(\sum_{t=1}^{n_T-1} \beta_{tw}u_t - \beta_{n_T w} \sum_{t=1}^{n_T-1} u_t)^2}{(\sum_{t=1}^{n_T-1} \mu_{dt}(\beta_{tw} - \beta_{n_T w}) + \beta_{n_T w})^2} \leq 0$$

.

Le calcul de la vraisemblance demande de marginaliser cette probabilité conditionnelle par rapport à μ_d, soit de calculer :

$$\mathrm{P}(W_d) \quad = \quad \int_{\mu_d} \prod_{w=1}^{n_W} (\sum_{t=1}^{n_T} \mu_{dt}\beta_{tw})^{C_{wd}} p(\mu_d) d\mu_d$$

Cette intégrale n'admet pas de résolution analytique ; il est en revanche possible de mettre en œuvre une approche de Monte Carlo, consistant à tirer M valeurs $\mu_d^{(m)}$ sous la loi *a priori* (Dirichlet) et à approcher $\mathrm{P}(W_d)$ par : $\frac{1}{M}\sum_{m=1}^{M} \mathrm{P}(W_d|\mu_d^{(m)})$.

5.2.3.2 Classification d'un document

Estimer la distribution des thèmes dans un document revient à calculer l'espérance conditionnelle de μ_d. Pour ce faire, on peut utiliser le même échantillon de Monte Carlo que précédemment en approchant $E[\mu_d|W_d, \beta]$ par :

$$\frac{\sum_{m=1}^{M} \mu_d^{(m)} \mathrm{P}(W_d|\mu_d^{(m)})}{\sum_{m=1}^{M} \mathrm{P}(W_d|\mu_d^{(m)})}$$

Une autre possibilité est de former la moyenne des $E[\mu_d|T_d, W_d]$ qui interviennent naturellement dans l'algorithme d'échantillonnage de Gibbs. Les différentes configurations d'indicatrices étant équiprobables a priori, il s'agit simplement de calculer :

$$\forall t \in \{1, \ldots, n_T\}, \widehat{\mu_{dt}} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}_{\{T_d = t\}}$$

5.2.4 Méthodes d'inférence alternatives

Par la suite, c'est donc l'échantillonneur de Gibbs dont nous évaluerons les performances. Néanmoins, nous décrivons ci-dessous deux autres méthodes d'inférence approchée développées pour LDA. Elles consistent toutes deux à appliquer des algorithmes itératifs en deux étapes similaires à l'EM, la différence résidant dans les approximations réalisées et les quantités maximisées [Minka and Lafferty, 2002].

5.2.4.1 Inférence variationnelle

[Blei et al., 2002] propose de déterminer une vraisemblance approchée par inférence variationnelle, c'est-à-dire en minorant, à d, i fixés, l'expression $\sum_{t=1}^{n_T} \mu_{dt}\beta_{tW_i}$, non-intégrable sur les valeurs possibles de μ_{dt}, par une fonction intégrable $\prod_{t=1}^{n_T} \left(\frac{\mu_{dt}\beta_{tW_i}}{\Phi_{W_i t}}\right)^{\Phi_{W_i t}}$ avec les Φ_{wt}, tels que $\sum_{t=1}^{n_T} \Phi_{wt} = 1$, pouvant être interprétés comme mesurant l'implication du terme w dans le thème t [Minka and Lafferty, 2002].

Les paramètres Φ sont eux-mêmes estimés de façon à maximiser la borne inférieure et donc à se rapprocher le plus possible de l'expression originale. L'autre étape de l'algorithme consiste naturellement à estimer $\hat{\alpha}$ et $\hat{\beta}$, à $\hat{\Phi}$ fixés. En alternant ces deux mises à jour successivement, on obtient après convergence des estimations des paramètres correspondant à une borne inférieure (maximisée localement) de la vraisemblance.

5.2.4.2 Espérance-Propagation

[Minka and Lafferty, 2002] montre que la méthode d'inférence variationnelle produit parfois des estimations peu satisfaisantes et attribue ces imprécisions à des approximations déficientes de la distribution a posteriori de μ. L'alternative proposée par l'algorithme espérance-propagation consiste également à remplacer la quantité $\sum_{t=1}^{n_T} \mu_{dt}\beta_{tW_i}$ par un produit intégrable analogue à l'expression d'une loi de Dirichlet. La différence vient de la procédure d'estimation des paramètres introduits : les termes relatifs à chaque mot sont modifiés successivement par insertions/suppressions pour se rapprocher, en termes de moyenne et variance, de l'expression initiale.

À partir de cette procédure d'approximation des probabilités a posteriori de μ, il est possible d'obtenir dans une autre étape d'approximation/maximisation des valeurs pour les paramètres β. La même technique que dans [Blei et al., 2002] est utilisable mais [Minka and Lafferty, 2002] en suggère une autre, minorant différemment la log-vraisemblance.

Alors que, sur des données peu volumineuses, l'inférence par espérance-propagation semble donner de meilleurs résultats que l'inférence variationnelle [Minka and Lafferty, 2002], il n'est pas évident que cette méthode soit applicable à des cas réels. [Buntine and Jakulin, 2006] souligne en effet que la complexité spatiale de cette méthode est largement supérieure à celles de l'échantillonneur de Gibbs et de l'inférence variationnelle, nécessitant une taille de stockage pour les calculs intermédiaires de l'ordre de ln_T. Nous n'avons donc pas considéré cette méthode d'inférence alternative à l'échantillonnage de Gibbs dans la section expérimentale qui suit.

5.2.5 Comparaison avec le modèle de mélange de multinomiales

5.2.5.1 Résultats

Nous testons à présent l'échantillonnage de Gibbs pour LDA sur le même corpus issu de Reuters que précédemment. La longueur des calculs est ici accentuée par le fait que les mesures de perplexité et de score de cooccurrences nécessitent elles aussi des simulations, comme vu en section 5.2.3, en plus des itérations de Gibbs dédiées à l'estimation de la matrice β. Par conséquent, nous avons restreint ces expériences à un jeu de données (apprentissage/test) fixe.

En appelant itération un ensemble de tirages modifiant une fois la fonction indicatrice associée à chaque occurrence de l'ensemble d'apprentissage, nous conduisons 10000 itérations de l'échantillonneur de Gibbs et mesurons l'évolution de la perplexité et du score de cooccurrences toutes les 50 itérations. Nous estimons la log-vraisemblance par la méthode la plus simple, consistant à moyenner chaque $\mathrm{P}(W_d|\mu_d^{(m)})$ sur 1000 tirages[2] et évaluons les probabilités d'appartenance sur les mêmes échantillons.

L'initialisation nécessite de fournir une configuration complète d'indicatrices. La stratégie la plus immédiate consiste à tirer l fois selon une loi discrète avec équiprobabilité entre les différents thèmes (initialisation « random »). Une autre idée, non applicable dans un cas réel mais permettant d'avoir une idée sur l'influence de l'étape d'initialisation, est d'initialiser toutes les occurrences d'un même document sur un même thème, correspondant

[2]Nous montrons dans l'article [Rigouste et al., 2006b] que cette technique d'estimation tend à sous-estimer nettement la vraisemblance lorsque le nombre de tirages est trop petit devant le nombre de paramètres. Cependant, ces expériences montrent que, pour 5 thèmes, moyenner sur 1000 échantillons donne des résultats raisonnables.

à sa catégorie Reuters. Cette initialisation est analogue à l'initialisation sur les catégories Reuters pratiquée avec l'algorithme EM pour le mélange de multinomiales.

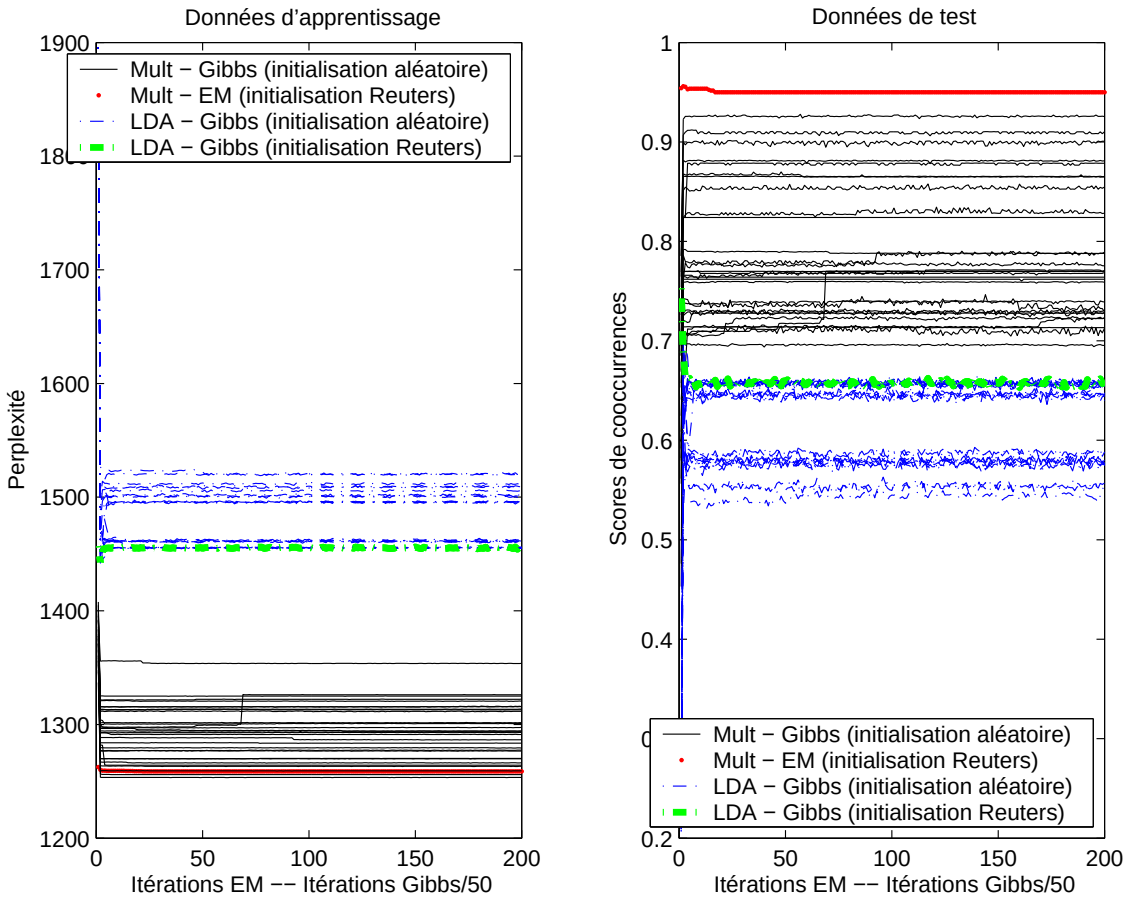

FIG. 5.4 – Évolution de la perplexité et du score de cooccurrence pour LDA.

Les résultats sont présentés en figure 5.4, en perspective avec les meilleures performances obtenues précédemment avec l'échantillonnage de Gibbs rao-blackwellisé et avec l'algorithme EM en initialisant sur les catégories Reuters. Nous constatons que l'échantillonneur de Gibbs pour le modèle LDA atteint rapidement des valeurs dont il ne s'écarte guère. Les variations dans les performances sont moins grandes que pour le modèle de mélange de multinomiales mais l'influence de l'initialisation, bien que plus faible, reste non négligeable. Pour ce qui est de l'applicabilité pratique de l'échantillonneur de Gibbs pour LDA, comme pour le modèle de mélange de multinomiales, il est donc difficile pour un utilisateur d'accorder une confiance aveugle à une méthode souffrant d'une telle instabilité. Sur le plan des performances moyennes, à la fois pour la perplexité d'apprentissage et pour le score de cooccurrences sur l'ensemble de test, l'échantillonneur de Gibbs sur LDA est systématiquement inférieur à l'échantillonneur de Gibbs sur le mélange de multinomiales.

Pour déterminer si la malédiction de la dimensionnalité sur le vocabulaire est la cause des performances moyennes de l'échantillonneur de Gibbs sur LDA, comme c'est le cas pour l'algorithme EM sur le mélange de multinomiales, nous avons reconduit les mêmes calculs avec un vocabulaire limité (les 900 mots les plus fréquents). Les résultats en figure 5.5 montrent que la taille du vocabulaire n'est pas liée directement aux scores obtenus et

la réduction du nombre de mots cause même une dégradation sensible des performances. Le fait que l'inférence ne donne pas de meilleurs résultats en dimension réduite montre que l'échantillonneur de Gibbs ne se comporte par sur LDA comme sur le modèle de mélange de multinomiales, ce que nous soulignons dans la section suivante.

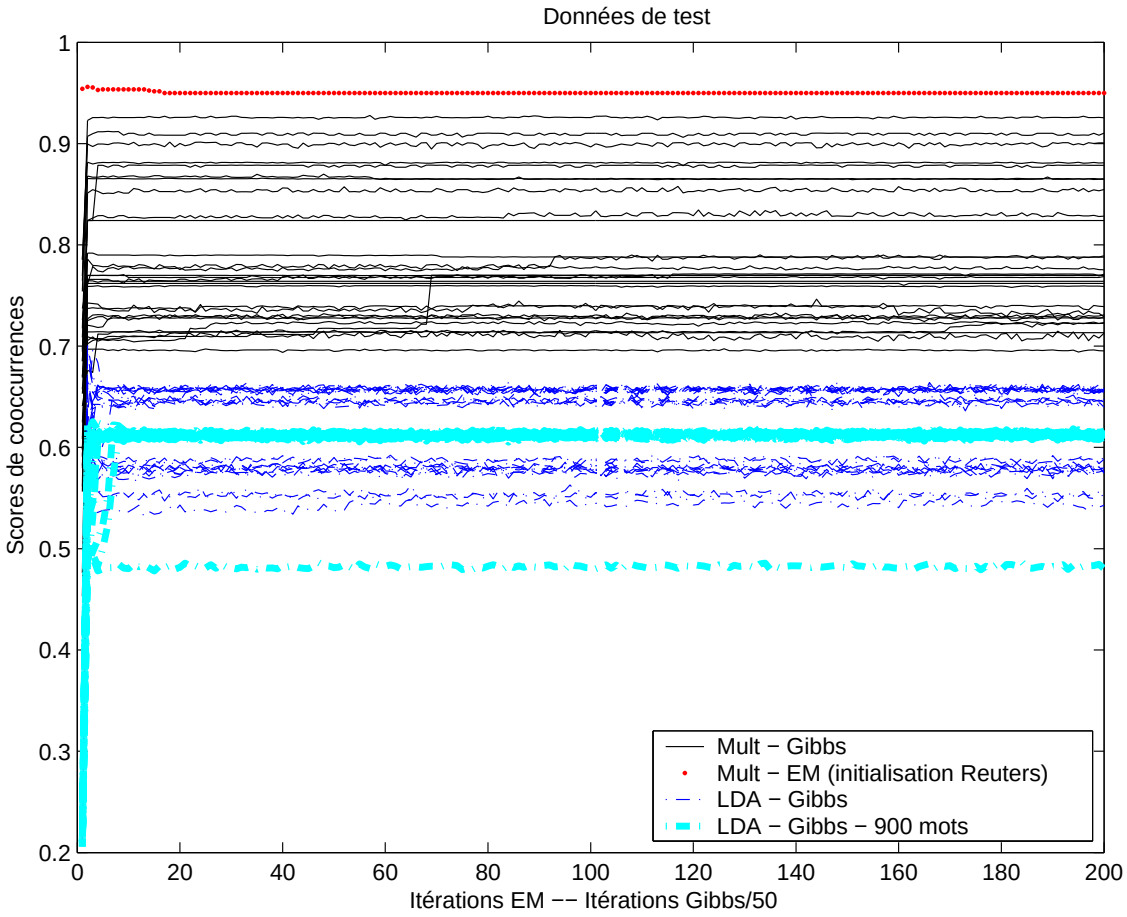

Fig. 5.5 – Évolution du score de cooccurrence pour LDA sur un vocabulaire réduit.

5.2.5.2 Biais d'un corpus de documents monothématiques

Le modèle de mélange de lois multinomiales ne construit qu'en apparence des associations probabilistes entre thèmes et documents. Dans la pratique, la très grande majorité des documents est affectée avec probabilité 1 à un thème unique, comme nous l'avons vu en section 4.2.4. Pour contraster ce comportement avec celui de LDA, nous avons, pour chacun de ces deux modèles, calculé l'entropie moyenne de la distribution des thèmes assignés à un document pour un ensemble de 500 documents de test (avec $n_T = 5$). En conservant les notations de la section 5.2.3, la grandeur mesurée est donc une moyenne (sur 20 simulations) de :

$$\exp\left(\frac{1}{n_{D^*}}\sum_{d=1}^{n_{D^*}} -\mu_d \log(\mu_d)\right)$$

Lorsque l'on construit une classification en n_T thèmes, cette grandeur varie entre 1 (af-

fectation déterministe) et n_T (répartition uniforme). Alors que, pour le modèle de mélange de lois multinomiales, cette grandeur est toujours très proche de 1 (sur les 20 initialisations, la moyenne est : 1.02), pour LDA, cette valeur approche 2 (en moyenne : 1.98), prouvant que ce modèle permet effectivement de construire des classifications probabilistes.

Cet aspect de LDA, inhérent à la multiplication des variables latentes thématiques au sein d'un même document, est fondamental pour expliquer ses contre-performances apparentes illustrées dans la section précédente. Les documents du corpus sur lequel nous travaillons sont des dépêches Reuters, souvent courtes et surtout très fortement « monothématiques ». Par conséquent, une mesure telle que le score de cooccurrence va naturellement pénaliser un modèle, comme LDA, orienté mécaniquement loin des classifications déterministes par son hypothèse de base de plurithématicité. Il ne faut pas, pour autant, en conclure que le modèle de mélange de multinomiales avec inférence par échantillonnage de Gibbs rao-blackwellisé est à préférer pour tous les problèmes de classification non supervisée. Mais ce choix dépend essentiellement des caractéristiques du corpus étudié, et en particulier des variations thématiques au sein des documents. Si de bonnes raisons existent pour considérer les documents comme monothématiques, nous avons montré dans cette section que l'hypothèse de LDA, induisant un modèle plus complexe et donc un traitement numérique plus long, ne permet non seulement pas de réduire la variabilité des résultats mais a, de plus, pour conséquence directe une dégradation des performances.

5.3 Interprétation des thèmes

5.3.1 Position du problème

Nous avons jusqu'ici peu insisté sur un aspect de l'analyse exploratoire capital pour un utilisateur final : la possibilité de caractériser et d'interpréter aisément les regroupements construits. Dans cette section, notre étude sera guidée intuitivement par les deux critères suivants :

- la méthode d'interprétation des thèmes que nous cherchons devra être relativement endogène par rapport au modèle choisi. Il y a dans cette exigence la volonté d'utiliser dans la phase d'analyse la représentation choisie pour la classification, sans avoir à développer une longue série de post-traitements additionnels ;
- d'autre part, pour que la phase d'interprétation soit la plus rapide possible, nous attendons de la méthode qu'elle nous livre une vision particulièrement synthétique chaque thème. La représentation la plus évidente de ce point de vue est une liste de mots caractéristiques. Nous avons par conséquent peu considéré des techniques complémentaires répondant moins à ce critère, telles que la sélection de documents les plus représentatifs d'un groupe.

5.3.2 Interprétation des paramètres β_{tw}

Un avantage indéniable des modèles probabilistes du point de vue de l'interprétabilité est l'estimation d'un paramètre liant les thèmes au vocabulaire. β_{tw} est égal à l'espérance du compte du mot w conditionnellement au thème t divisé par le nombre total de tirages (dans le cas où les comptes sont estimés par des lois multinomiales). Il peut donc être vu comme la fréquence du mot dans le thème en question. Néanmoins, la donnée de la probabilité d'émission des mots dans chaque thème ne résout que partiellement le problème de l'interprétation des groupes. Bien que les β_{tw} soient représentatifs des thèmes, leur

interprétation n'est pas pour autant immédiate car les mots les plus pertinents pour définir un thème ne sont pas nécessairement ceux qui ont les β_{tw} les plus forts. Pour illustrer les différents problèmes qui émergent, considérons une matrice β particulière correspondant au mélange de multinomiales, obtenue avec l'algorithme EM et l'initialisation Dirichlet. Nous étudions un cas où les scores finals sont de 1467.6 pour la perplexité sur l'ensemble d'apprentissage et 0.55 pour le score de cooccurrence sur l'ensemble de test. Il est important en effet que la classification à analyser ne soit pas trop proche des catégories Reuters sinon la tâche paraîtrait, à tort, exagérément simple devant la clarté des regroupements.

L'idée la plus immédiate pour l'analyse des mots importants est de sélectionner ceux dont l'apparition dans un thème donné est la plus probable, c'est-à-dire, littéralement, les plus grandes valeurs de β_{tw}. Cependant, en appliquant cette stratégie au premier thème, nous voyons que la liste obtenue n'est guère informative :

the, of, to, in, and, said, on, for, is, that, was, it, with, by, he, his, at, from, as, be

Les mots les plus probables sont en effet les anti-mots. En les filtrant grâce à une liste pré-définie, nous obtenons la réponse suivante pour le thème 1 :

year, million, world, people, jackson, years, told, film, time, percent, law, health, women, government, week, state, life, group, abortion, company

Étant entendu que nous cherchons des indices nous permettant de retrouver les catégories Reuters initiales[3], faisant toujours l'hypothèse qu'il s'agit de la classification de référence, la plupart des mots de la liste ci-dessus paraissent inutiles. Ainsi, les mots années, million ou monde semblent peu informatifs. Il n'aurait pas été absurde de les faire figurer également sur la liste des anti-mots. Cependant, plus cette liste est élargie, plus le risque d'écarter des mots pertinents est grand. En réalité, même une liste très réduite est susceptible de contenir des mots pertinents : nous verrons par exemple plus loin que le pronom possessif his (ou her) caractérise bien la catégorie Reuters culture, dont les articles évoquent souvent un artiste avant d'aborder ses créations par l'emploi de possessifs. Ces remarques mettent en relief que le degré non nul de subjectivité et d'arbitraire nécessaire à la constitution d'une liste d'anti-mots est rédhibitoire dans cette optique d'analyse des thèmes. C'est la raison pour laquelle nous nous intéressons dans ce qui suit à des méthodes n'ayant pas recours à ce type de filtrage.

Plus que la présence des anti-mots, le facteur qui semble perturber l'analyse est que nous essayons de comparer des termes qui n'ont pas la même fréquence d'apparition dans le corpus. Finalement, ce qui nous intéresse, c'est de caractériser chaque thème par les mots pour lesquels la probabilité s'écarte le plus de la moyenne. C'est pourquoi nous établissons un nouveau classement des mots par ordre décroissant de la valeur suivante :

$$\beta_{tw} - \frac{1}{n_T} \sum_{t=1}^{n_T} \beta_{tw}$$

Le résultat est présenté en tableau 5.1.

Nous obtenons à présent une meilleure idée de la constitution des thèmes, malgré la présence toujours nuisible des anti-mots[4]. Ainsi, les thèmes 3 et 4 semblent correspondre

[3] Rappelons qu'il s'agit des catégories sport, accidents, santé, emploi et culture.

[4] Les retirer améliore légèrement le résultat mais avec l'inconvénient évoqué ci-dessus que, parmi les mots supprimés, certains anti-mots sont en fait caractéristiques des thèmes.

Thème 1	Thème 2	Thème 3	Thème 4	Thème 5
him	this	work	state	metall
kevorkian	world	jobs	killed	released
suicide	aids	plant	from	employers
parliament	pilgrims	its	km	new
she	health	year	had	their
bill	irish	hargrove	hurricane	with
life	that	plants	miles	ig
miss	australian	auto	storm	goal
world	it	at	plane	game
who	race	said	air	who
women	has	september	fire	match
film	have	rate	officials	pay
has	be	unemployment	people	second
law	for	canadian	and	soccer
abortion	tobacco	canada	was	th
is	percent	percent	on	cup
he	year	union	of	sick
to	is	strike	the	league
jackson	to	workers	were	his
his	the	gm	said	first

TAB. 5.1 – Sélection de termes sur les β_{tw} après soustraction de la moyenne.

assez précisément aux catégories Reuters emploi et accidents respectivement. Les catégories Reuters restantes : santé, culture et sports sont en apparence équitablement représentées dans les trois autres thèmes. Sans connaissance supplémentaire à propos du corpus, il est difficile d'aller plus loin. Cela ne signifie cependant pas que ces thèmes 1, 2 et 5 sont totalement dénués de cohérence, et nous poursuivons à présent leur étude grâce à des techniques de caractérisation des thèmes plus évoluées.

5.3.3 Sélection de mots représentatifs

5.3.3.1 Méthode hypergéométrique

L'analyse de regroupements n'est pas une problématique nouvelle : la question s'est posée de façon particulièrement critique avec l'avènement de l'analyse en composantes principales (ACP) [Lebart and Salem, 1988]. En effet, bien que cette technique garantisse de conserver une grande partie de la covariance des données originales sur un nombre restreint de directions propres, la compréhension de ce que sont ses sous-espaces propres était également un enjeu majeur pour l'acceptation de l'ACP par les utilisateurs.

Parmi l'éventail de techniques exposées par [Lebart and Salem, 1988], nous avons retenu le calcul des spécificités [Lebart and Salem, 1988, section 4.2] qui consiste en la recherche des *mots* les plus discriminants de chaque thème (positivement ou négativement), de façon un peu analogue à l'exemple élémentaire ci-dessus.

Le calcul des spécificités consiste à comparer les nombres d'occurrences attendu et réel d'un mot dans un groupe, le nombre d'objets attendu étant modélisé par une loi hypergéométrique. En considérant un thème t et un mot w, notons l_t^T la « longueur » du

thème t et l_w^W le nombre d'occurrences du mot w dans le corpus définis par :

$$\begin{aligned} l_t^T &= \sum_{d=1}^{n_D} l_d \, \mathrm{P}(T_d = t|C) \\ l_w^W &= \sum_{d=1}^{n_D} C_{wd} \end{aligned}$$

De façon analogue à la classification déterministe, il est possible de voir le thème t comme un ensemble de l_t^T occurrences et la question à se poser est alors « le nombre $n_{tw} = \lfloor \beta_{tw} l_t^T \rfloor$ d'occurrences du mot w au sein de ce groupe est-il exceptionnel[5] ? ». Supposons donc que nous tirions l_t^T occurrences parmi les l occurrences du corpus (chaque mot est compté autant de fois qu'il apparaît) sans ordre et sans remise. Selon la loi hypergéométrique, la probabilité de voir k fois le mot w s'obtient en dénombrant les combinaisons de k individus parmi les l_w^W occurrences du mot considéré et les combinaisons de $m-k$ objets sur toutes les autres occurrences. Nous obtenons par conséquent :

$$\forall k \in \{0, \ldots, l_w^W\}, \mathrm{P}(X = k) = \frac{\binom{l_w^W}{k}\binom{l - l_w^W}{l_t^T - k}}{\binom{l}{l_t^T}},$$

cette probabilité étant naturellement nulle pour $k > l_w^W$ (il est impossible de tirer plus de fois le terme w que son nombre d'occurrences dans le corpus entier). Plus précisément, les deux quantités que nous allons mesurer sont :

$$\begin{aligned} S_-(t,w) &= \mathrm{P}(X \leq n_{tw}) && (5.4) \\ &= \sum_{k=0}^{n_{tw}} \mathrm{P}(X = k) \\ S_+(t,w) &= \mathrm{P}(X \geq n_{tw}) && (5.5) \\ &= \sum_{k=n_{tw}}^{l_w^W} \mathrm{P}(X = k) \\ &= 1 - S_-(t,w) + \mathrm{P}(X = n_{tw}) \,. && (5.6) \end{aligned}$$

Nous obtenons ainsi, pour chaque mot, son degré de spécificité pour chaque thème : plus $S_+(t,w)$ est petit, plus w est positivement spécifique (sensiblement plus d'occurrences que prévu) ; inversement, plus $S_-(t,w)$ est petit, plus w est négativement spécifique (sensiblement moins d'occurrences que prévu). La caractérisation d'un thème s'effectue en gardant les mots avec les S_+ et S_- les plus faibles.

Sur le plan pratique, nous évaluons ces probabilités en log, grâce à la fonction `gammaln`. Il est en général judicieux de commencer par (5.4), qui contient souvent moins de termes à calculer que (5.5), et d'évaluer la spécificité positive par la formule de complémentarité (5.6). D'autre part, comme c'est souvent le cas avec les formules impliquant des factorielles,

[5] Ce nombre peut être exceptionnel de deux façons : exceptionnellement petit (mot très rare dans le thème par rapport à son apparition dans le reste du corpus) ou exceptionnellement grand (mot très fréquent par rapport au reste du corpus).

il n'est pas utile de recalculer pour chaque valeur k les trois coefficients du binôme car il existe une formule de récurrence simple :

$$\forall k \in \{1, \ldots, l_w^W\}, \mathrm{P}(X = k-1) \quad = \quad \mathrm{P}(X = k)\frac{k(l - l_w^W - l_t^T + k)}{(l_w^W - k + 1)(l_t^T - k + 1)}$$

Dans la table 5.2, nous montrons les résultats des calculs de spéficité pour la même classification que précédemment. Nous sélectionnons au total 20 mots par thème, de même que précédemment, mais cette fois répartis entre 10 positivement spécifiques et 10 négativement spécifiques. Une première remarque est que les spécificités négatives sont ici d'une utilité limitée dans la mesure où les 5 catégories Reuters initiales sont relativement bien délimitées et les spécificités négatives des thèmes sont généralement des spécificités positives d'un autre[6]. Néanmoins, dans des cas réels où les spécificités positives seraient particulièrement difficiles à interpréter, les spécificités négatives peuvent fournir des indications utiles pour délimiter le groupe.

Les listes obtenues par cette méthode sont globalement plus cohérentes et dépourvues de mots non-informatifs. La cohérence du thème 3 avec le thème emploi n'est pas remise en cause (le mot le plus représentatif, gm, signifie General Motors) mais on décèle une influence lourde d'une série d'articles traitant d'une crise au sein de General Motors Canada, dans laquelle le syndicat CAW (Canadian Auto Workers), présidé par Buzz Hargrove, a joué un rôle important. Le thème 3 n'est donc peut-être qu'un sous-groupe de la catégorie Reuters sur l'emploi. Le thème 4 ne pose pas de questions particulières. Un examen des textes permet d'expliquer mieux le thème 1, qui semblait être un mélange des articles de culture et de santé. En réalité, il regroupe des actualités touchant à diverses affaires de mœurs : accusations de pédophilie pour Michael Jackson, questions relatives à l'euthanasie et au suicide assisté pour le médecin américain Kevorkian, lois sur l'avortement, menaces de mort et censure pour Salman Rushdie et manifestations anti-« concours de Miss monde » dans la ville indienne de Bangalore. Les thèmes 2 et 5 sont plus nébuleux et constitués de dépêches sur le sport, la santé, l'emploi ou les catastrophes. Le thème 2 regroupe plus spécifiquement les articles multi-thématiques : des travailleurs dans un élevage d'autruches (ostrich) atteints de la fièvre congolaise, les morts de pélerins (pilgrims) hindous dans une grotte (cave) lors d'une randonnée dans le Kashmir, des incidents internes à la SAA (South African Airways) ainsi que des articles généraux sur le tabac (tobacco) et le sport (race, ARL - Australian Rugby League). La présence du mot irlandais (irish) reste en revanche inexplicable mais cela n'est pas surprenant dans la mesure où il s'agit d'un terme très multicatégoriel dans ce corpus. Le thème 5 semble être la fusion d'articles généraux sur le sport (league, soccer, cup, goal, th -venant de 4th, 5th, par exemple) et de dépêches très particulières couvrant un conflit en Allemagne entre le syndicat IG Metall et la fédération d'employeurs Gesamtmetall sur la rémunération (pay) des jours-maladies (sick).

5.3.3.2 Méthode bêta

Dans le même ordre d'idée, il est possible de déterminer, de façon bayésienne, la distribution des comptes de mots et de la comparer aux différents β_{tw}. Ainsi, en considérant un mot particulier w, le nombre de ses apparitions pour un nombre de tirages fixé sur le vocabulaire suit une loi binomiale avec, sur le corpus d'apprentissage, l_w^W « succès » (nombre d'apparitions du mot) et $l - l_w^W$ « échecs » (nombre d'apparitions d'un autre

[6]Il est d'ailleurs remarquable que les spécificités négatives des thèmes les plus « flous » (1, 2 et 5) se constituent quasi-exclusivement de mots représentatifs des thèmes les mieux délimités (3 et 4).

Thème 1		Thème 2		Thème 3	
+	-	+	-	+	-
jackson	strike	pilgrims	gm	gm	fire
abortion	workers	tobacco	plane	hargrove	plane
rushdie	union	ostrich	hurricane	workers	storm
kevorkian	wage	race	storm	strike	hurricane
suicide	storm	arl	crash	auto	people
bangalore	officials	irish	fire	canadian	crash
miss	gm	cave	sick	canada	killed
law	plane	congo	cuba	plants	air
abortions	were	saa	air	caw	tobacco
his	hurricane	kashmir	strike	unemployment	homes

Thème 4		Thème 5	
+	-	+	-
plane	his	sick	said
fire	beat	ig	strike
storm	cup	metall	state
were	th	league	gm
hurricane	league	soccer	were
air	gm	cup	government
miles	first	gesamtmetall	people
winds	film	goal	fire
officials	soccer	pay	plane
homes	won	th	air

Tab. 5.2 – Termes retenus par la méthode « hypergéométrique ».

Thème 1		Thème 2		Thème 3	
+	-	+	-	+	-
jackson	strike	pilgrims	gm	gm	people
abortion	workers	tobacco	plane	hargrove	fire
rushdie	union	ostrich	storm	auto	plane
kevorkian	were	race	hurricane	strike	storm
suicide	officials	arl	fire	workers	hurricane
bangalore	wage	cave	strike	canadian	air
miss	storm	congo	air	canada	killed
abortions	plane	irish	crash	plants	crash
law	gm	saa	were	caw	homes
concert	said	kashmir	sick	unemployment	tobacco

Thème 4		Thème 5	
+	-	+	-
plane	his	sick	said
fire	th	ig	were
were	cup	metall	strike
storm	beat	league	state
hurricane	first	soccer	government
air	league	cup	people
miles	he	gesamtmetall	gm
winds	gm	goal	officials
officials	film	liverpool	fire
km	world	employers	air

Tab. 5.3 – Termes retenus par la méthode « χ^2 ».

mot). Si nous faisons l'hypothèse d'un a priori conjugué bêta (loi Dirichlet en dimension 2) de paramètre $(\lambda_\beta, \lambda_\beta)$, la variable aléatoire X_w d'apparition du mot w suit encore une loi bêta de paramètre $(l_w^W + \lambda_\beta, l - l_w^W + \lambda_\beta)$ (il s'agit d'un cas particulier de la conjugaison Dirichlet-Multinomiale vue en section 4.1.1). Considérons donc la fonction de répartition de la loi bêta, appelée fonction bêta incomplète régularisée :

$$\mathrm{P}(X_w \leq x) \quad = \quad \frac{\int_0^x t^{l_w^W + \lambda_\beta}(1-t)^{l - l_w^W + \lambda_\beta} dt}{\int_0^\infty t^{l_w^W + \lambda_\beta}(1-t)^{l - l_w^W + \lambda_\beta} dt}$$

En calculant cette fonction par des méthodes itératives approchées [Press et al., 1992], il est possible d'évaluer $\mathrm{P}(X_w \leq \beta_{tw})$. Si cette probabilité est très proche de 0, β_{tw} est étonnamment petit et est donc négativement spécifique du thème. En revanche, si cette probabilité est très proche de 1, β_{tw} est plus grand que prévu et est donc positivement représentatif du thème. Là encore, il est possible de conserver la liste des termes crédités des scores les plus forts, dans un sens ou dans l'autre.

Néanmoins, pour cette méthode, un problème numérique se pose compte tenu de la forme « pointue » des distributions bêta obtenues (typiquement les paramètres sont de l'ordre de $(10, 40000)$). La plage de valeurs pour lesquelles la fonction de répartition ne renvoie pas exactement 0 ou 1 est très réduite (pour les paramètres ci-dessus, il s'agit approximativement de l'intervalle $[10^{-4}, 5.10^{-4}]$) et, par conséquent, un trop grand nombre de mots (précisément ceux qui nous intéressent) obtiennent exactement les scores 0 ou 1 et ne peuvent pas être classés par ordre de pertinence. Il serait donc nécessaire afin de pouvoir utiliser cette méthode d'implémenter un algorithme plus précis de calcul de la fonction de répartition de la distribution bêta ou bien de lisser cette dernière, tout en préservant l'ordre des spécificités, des solutions que nous n'avons pas menées à leurs termes pour l'instant.

5.3.3.3 Méthode χ^2

Finalement, le problème posé revient à déterminer si la distribution $\beta_{t\bullet}$ est ou non équivalente à la distribution unigramme inférée à partir du corpus entier. Le test statistique usuel permettant de répondre à cette question est le test du χ^2 [Cochran, 1952].

Nous cherchons à comparer deux distributions discrètes sur le vocabulaire, le coefficient unigramme correspondant au mot w étant l_w^W/l, sur la base des l observations séparées en n_W groupes (1 par mot). La théorie nous indique alors que lorsque le nombre l d'occurrences tend vers l'infini, sous l'hypothèse H_0 que $\beta_{t\bullet}$ provient d'un échantillon de la distribution unigramme, la quantité

$$l \sum_{w=1}^{n_W} \frac{(\beta_{tw} - l_w^W/l)^2}{l_w^W/l} \tag{5.7}$$

converge en loi vers une distribution χ^2 à $n_W - 1$ degrés de liberté.

L'hypothèse de cette stratégie de détection de spécificités est que cette propriété asymptotique globale est exploitable au niveau local, en supposant que les termes les plus grands de la somme (5.7) sont précisément ceux qui, au sein de la distribution $\beta_{t\bullet}$, dévient le plus de la distribution unigramme, et sont donc les plus spécifiques du thème. L'idée est donc de calculer la formule pour tous les w et de classer les mots par cette mesure. Comme dans la section précédente, il peut être utile de distinguer les spécificités positives ($\beta_{tw} \gg l_w^W/l$)

des spécificités négatives ($\beta_{tw} \ll l_w^W/l$) puisque les deux cas sont susceptibles d'apparaître dans la liste[7]. Nous pouvons donc créer deux listes en gardant les scores les plus importants positivement puis négativement de la quantité suivante :

$$\text{signe}(\beta_{tw} - l_w^W/l)\frac{(\beta_{tw} - l_w^W/l)^2}{l_w^W/l}$$

Le résultat de cette sélection est présenté dans le tableau 5.3. Nous constatons que l'accord avec la méthode précédente est excellent. Dans la mesure où la formule correspondant à la méthode du « χ^2 » est beaucoup plus simple et rapide à calculer, nous la préférerons à la méthode « hypergéométrique ».

Pour conclure cette section, nous montrons que lorsqu'une classification obtient de meilleurs scores qu'une autre, la différence en termes de cohérence est également manifeste à l'examen des groupes. Ainsi, nous nous intéressons au résultat d'une trajectoire de la méthode d'inférence itérative de la section 4.3.2 telle que celles représentées en figure 4.9. Les scores exacts de la classification analysée sont de 1283.8 pour la perplexité sur l'ensemble d'apprentissage et de 0.82 pour le score de cooccurrence sur l'ensemble de test. Contrairement aux précédentes, les listes du tableau 5.4 sont suffisamment claires pour ne pas nécessiter d'explications annexes. Elles montrent que les regroupements sont très cohérents et proches de la catégorisation Reuters de reférence. Notons que ces résultats constituent un argument solide contre la suppression des anti-mots, dans la mesure où le thème 5 (culture) est très fortement caractérisé par les pronoms à la troisième personne pour les raisons déjà évoquées en section 5.3.1.

5.3.4 Autres méthodes d'interprétation

En section 5.3.3.3, nous avons utilisé le test du χ^2 de la façon la plus immédiate possible, thème par thème. En réalité, ce test peut être adapté à une situation proche de la notre, dans laquelle la population globale peut être divisée en plusieurs sous-populations et l'hypothèse à tester est la similarité des distributions des différentes sous-populations avec une distribution globale. La somme formée ressemble à celle étudiée en (5.7) dans la mesure où la distribution de fond est constituée de l'ensemble des sous-populations et correspond donc à nouveau à la distribution unigramme. Les différentes sous-populations contribuent à la somme du χ^2 en proportion de leur nombre d'individus ce qui, dans notre cas, peut être adapté de la façon suivante :

$$l\sum_{t=1}^{n_T}\alpha_t\frac{(\beta_{tw} - l_w^W/l)^2}{l_w^W/l}$$

qui converge également vers une loi du χ^2 lorsque l tend vers l'infini. L'utilisation d'une telle formule pour la sélection de termes représentatifs n'est en fait pas très différente de (5.7) : elle introduit simplément une pondération par le poids relatif du thème α_t qui permet d'effectuer une recherche globale de spécificités sur tous les thèmes et de les ordonner par importance indépendamment du thème auquel ils sont rattachés. Mais d'autres applications sont envisageables, notamment dans la sélection du nombre de thèmes : le score de χ^2 mesure, dans un certain sens, la différentiation des thèmes par rapport au fond commun,

[7] Cependant, en pratique, nous constatons que lorsque la distinction n'est pas faite, ce sont les spécificités positives qui ressortent aux premières places, l'écart à la distribution unigramme étant plus important en valeur absolue que pour les spécificités négatives.

Thème 1		Thème 2		Thème 3	
+	-	+	-	+	-
health	his	strike	his	beat	said
drug	strike	union	world	cup	of
disease	workers	workers	her	league	that
study	was	gm	who	th	the
aids	her	pay	th	soccer	people
eu	cup	unions	people	first	percent
market	beat	wage	was	match	it
patients	he	talks	cup	goal	government
beef	after	sick	beat	second	were
tobacco	th	employers	disease	game	workers

Thème 4		Thème 5	
+	-	+	-
were	his	his	percent
fire	percent	her	workers
plane	will	film	strike
storm	for	he	union
miles	strike	she	health
km	is	jackson	were
hurricane	union	you	officials
killed	first	him	government
winds	with	miss	gm
officials	he	book	said

TAB. 5.4 – Termes retenus pour l'inférence itérative (méthode « χ^2 ».)

une statistique riche d'enseignement si nous hésitons sur le nombre de composantes à conserver.

Notons l'existence d'approches relevant d'une optique différente : en plus de chercher les mots les plus représentatifs, il est également envisageable de caractériser les groupes par leurs *éléments* les plus représentatifs. [Lebart and Salem, 1988] donne deux exemples de ces méthodes avec sélection par un critère lié au calcul des spécificités d'une part ou par un critère de type χ^2 d'autre part. Cette méthode s'applique assez bien au cas que Lebart et Salem considèrent, dans la mesure où leurs observations (ce que nous avons appelé les *documents*) sont très courtes et peuvent donc être visionnées rapidement. Dans notre cas, les documents sont plus longs et en lire plusieurs en intégralité pour chaque thème peut être fastidieux. Une solution est alors d'en obtenir une synthèse, par exemple en sélectionnant les paragraphes de chaque document jugés les meilleurs représentants du thème. Mais cette technique soulève d'autres problèmes, tels que l'absence de cohésion des passages extraits, dont le traitement est abordé dans les travaux sur le résumé automatique [Mani, 1999].

Pour les méthodes probabilistes, la sélection de documents représentatifs devrait pouvoir se faire naturellement à partir des résultats des algorithmes, en considérant pour le thème t les documents d pour lesquels la probabilité $\mathrm{P}(T_d = t|C_d)$ est la plus forte. Cependant, en pratique, cette stratégie se heurte au problème identifié en section 4.2.4 que ces probabilités sont souvent très proches de 0 ou de 1, si bien qu'un grand nombre d'éléments ont la probabilité maximale de 1 d'appartenir à un thème donné et l'exigence d'obtenir une liste synthétique des documents les plus représentatifs n'est ici pas respectée.

Il existe enfin des techniques non probabilistes, tirant leur inspiration de techniques

d'apprentissage (sélection de variables) ou d'analyse de données classique (analyse de la variance), voire des deux. Des exemples d'utilisation de ce type de techniques sont [Clérot et al., 2004] ou [Rigouste et al., 2006a], reproduit en annexe B.

5.4 Conclusion

Dans ce chapitre, nous avons d'abord comparé les performances de l'échantillonneur de Gibbs appliqué au modèle de mélange de lois multinomiales à deux algorithmes classiques en fouille de textes : l'algorithme des K-moyennes et l'échantillonneur de Gibbs appliqué au modèle LDA.

- L'algorithme des K-moyennes avec pondération « fréquence des documents inversée » se comporte globalement bien sur le problème Reuters, mais avec une grande sensibilité aux conditions initiales comme l'algorithme EM. Ses performances sont légèrement inférieures en moyenne à celles de l'échantillonneur de Gibbs rao-blackwellisé.
- Les performances du modèle LDA sont très nettement moins bonnes et en particulièrement systématiquement inférieure à celles de l'échantillonneur de Gibbs sur le modèle de mélange de multinomiales. En revanche, l'algorithme d'échantillonnage de Gibbs pour le modèle LDA semble moins souffrir des problèmes de variabilité dans les performances que les précédents. La diminution de la dimensionnalité n'apporte dans ce cas pas d'amélioration.

Il est important de relativiser la portée de ces comparaisons, qui n'ont été effectuées que sur un corpus, par ailleurs selon une méthode d'évaluation fortement biaisée en faveur de la classification déterministe. Le protocole de comparaison a donc naturellement tendance à sur-estimer les K-moyennes et à sous-estimer l'échantillonneur de Gibbs pour LDA. Néanmoins, une conclusion utile est qu'il n'est pas souhaitable de privilégier systématiquement LDA au modèle de mélange de multinomiales, a fortiori dans les cas où les documents traités ont peu de chances d'être très pluri-thématiques.

Nous avons également abordé brièvement le sujet de l'interprétation des thèmes, en proposant plusieurs méthodes de sélection de termes représentatifs grâce à la matrice β. Il est rassurant de constater que les deux méthodes les plus convaincantes donnent des résultats équivalents, ce qui conduit naturellement à privilégier la plus rapide, à savoir la méthode du χ^2. Des pistes pour prolonger ce travail sont :

- déterminer la similarité théorique entre les méthodes « hypergéométriques » et « χ^2 » justifiant la similarité des résultats ;
- justifier l'utilisation de la méthode χ^2 dans un cas « non asymptotique » (l du même ordre de grandeur que n_W)
- utiliser la statistique du χ^2 à plusieurs sous-populations, à d'autres fins, notamment pour la sélection de modèles.

Conclusion

Atouts du modèle de mélange de multinomiales

Cette thèse a abordé le problème de la classification non supervisée de documents par l'étude approfondie d'un modèle particulier : le mélange de lois multinomiales avec variables latentes thématiques au niveau des documents.

Dans l'état de l'art, nous avons présenté un panorama détaillé de la classification non supervisée de données textuelles, méthodes vectorielles (K-moyennes, analyse sémantique latente, factorisation en matrices non négatives, goulot d'information) et méthodes probabilistes (mélange de multinomiales, analyse sémantique latente probabiliste, allocation Dirichlet latente, Gamma-Poisson). Nous avons mis l'accent sur les raisons qui nous conduisent à préférer l'étude des modèles probabilistes, à savoir les facilités de généralisation et d'interprétation des résultats et l'adaptabilité naturelle du cadre statistique à des situations variées, notamment aux cadres supervisé et semi-supervisé. Parmi ces derniers, nous justifions notre intérêt pour le modèle le plus simple par notre sentiment qu'une étude détaillée des différentes stratégies d'inférence n'a pas été réalisée.

De fait, nous montrons ensuite qu'il y a beaucoup plus à dire sur le modèle de mélange de lois multinomiales que le simple fait que l'algorithme espérance-maximisation est très sensible aux optimums locaux et donne globalement de mauvaises performances. Par une étude complète des effets de la variation de la taille du vocabulaire, nous montrons que le nombre d'optimums locaux diminue nettement pour un nombre de mots réduit et qu'il est possible de tirer partie de ce constat pour proposer une stratégie heuristique d'inférence. Cette méthode itérative par ajout de mots rares affiche d'excellentes performances de classification non supervisée sur notre corpus et également une stabilité beaucoup plus importante que l'EM non modifié.

Nous avons également appliqué l'inférence par échantillonnage de Gibbs à ce modèle. Nous montrons en quoi l'approche bayésienne d'intégration des paramètres permet d'accélérer l'inférence par échantillonneur de Gibbs tout en améliorant ses performances. Cette version de l'échantillonneur, dite rao-blackwellisée, donne des résultats moyens aussi bons que la méthode d'inférence itérative, mais avec une plus grande sensibilité à l'initialisation, du niveau de celle que nous avons observée avec l'algorithme EM.

Dans le dernier chapitre, nos expériences sur l'algorithme des K-moyennes et le modèle d'allocation Dirichlet latente montrent que ces méthodes sont également très sensibles à l'initialisation. Sur le jeu de données Reuters, la stratégie d'inférence itérative par ajout de mots rares obtient des performances meilleures et plus stables que toutes les autres méthodes considérées. Ce résultat illustre, à notre avis, que le modèle de mélange de multinomiales devrait rester un choix prioritaire pour la classification non supervisée de documents, à moins d'avoir une raison objective de privilégier la multithématicité à l'intérieur d'un même document, une caractéristique du modèle LDA.

Enfin, nous avons proposé des critères endogènes aux modèles probabilistes pour la sélection de mots représentatifs des thèmes. En particulier, nous avons montré qu'un critère de type χ^2 permet de calculer facilement les éléments qui dévient le plus de la représentation unigramme pour un thème, et donc de comprendre les résultats de la classification.

Perspectives

La direction la plus naturelle pour poursuivre ce travail nous semble être de considérer d'autres cas que celui du corpus Reuters étudié. La section 4.5 a donné une idée de la difficulté de la tâche, nous détaillons à présent les différents axes possibles.

L'augmentation du nombre de thèmes semble générer des comportements très différents de ceux étudiés. Nous pensons que les problèmes d'inférence y sont a fortiori plus difficiles, dans la mesure où le nombre de paramètres augmente alors que le nombre d'observations reste le même. Une limitation de la taille du vocabulaire est probablement inévitable dans ce contexte. D'un point de vue pratique, réduire la complexité des algorithmes en jeu est important dans ce cas, de même que pour l'application à des corpus de grande taille.

Nous n'avons pas étudié le problème du choix du nombre de thèmes. Or les travaux concernant la sélection de modèles peuvent se révéler très utiles dans ce contexte. Les idées de stabilité de la classification, de variations de la perplexité en fonction du nombre de paramètres ou de différenciation par rapport à la distribution unigramme sont autant de pistes pouvant guider la recherche dans cette voie.

Lorsque les documents du corpus ne sont pas du tout monothématiques, l'idée de placer les variables latentes au niveau des paragraphes et de les lier par une chaîne de Markov, présentée pour le modèle de mélange de multinomiales en annexe B et, de façon voisine, pour PLSA dans [Mei and Zhai, 2005], mériterait d'être étudiée en détail, en proposant un algorithme d'inférence plus abouti. Dans tous les cas, l'idée de considérer le paragraphe, et non plus le document, comme unité de base, comme le suggèrent [Clérot et al., 2004] et [Antonellis and Gallopoulos, 2006] notamment, atténuerait peut-être certaines difficultés soulevées dans cette thèse, notamment les problèmes numériques liés à la dimensionnalité (section 4.2.4). Il serait alors nécessaire de proposer et comparer des stratégies de fusion pour traduire la classification sur les paragraphes en une classification sur les documents finalement exploitable par l'utilisateur.

Le problème de la modélisation des comptes mis en évidence en section 2.3.7.2 reste ouvert. Le modèle de mélange de lois binomiales négatives A.1 obtiendrait probablement, à condition de venir à bout de l'étape d'inférence, de meilleurs scores de perplexité pour la modélisation des données. Il reste à savoir si cette meilleure adéquation se traduirait par des capacités de classification accrues.

Glossaire

Nous présentons dans cette section les conditions d'emploi dans la thèse de certains termes courants. Ce glossaire n'a pas vocation à donner des définitions précises mais plutôt des clarifications succinctes et des renvois vers des sections plus détaillées.

Analyse exploratoire, classification non supervisée : Nous désignons sous ces expressions la tâche constituant l'objectif applicatif de la thèse et consistant à diviser un corpus en groupes de documents similaires. Cet enjeu est détaillé en section 1.2.

Catégorie, classe, étiquette : Dans un cadre *supervisé*, un ensemble de documents a été divisé au préalable en différents groupes, en général par des observateurs humains. *Catégories*, *classes* et *étiquettes* font indifféremment référence à ces groupes. Le mot *thème* est également parfois utilisé dans ce sens mais nous nous astreignons ici à le réserver aux groupes formés par un algorithme de classification.

Classification déterministe, classification probabiliste : Nous parlons de *classification déterministe* si un texte n'est pas autorisé à appartenir simultanément à plusieurs thèmes, c'est-à-dire que chaque thème peut se définir de manière unique comme un ensemble de documents et l'ensemble des thèmes constitue une partition du corpus. Dans le cas contraire, le résultat de la classification est une affectation *probabiliste* des documents à chaque groupe (il y a donc, pour chaque document, n_T valeurs comprises entre 0 et 1 qui somment à 1). Nous revenons sur cette distinction, centrale pour les modèles probabilistes, tout au long du document.

Classification supervisée : La caractéristique essentielle qui distingue la *classification supervisée* de l'*analyse exploratoire* (section 1.2)est que les groupes de documents ont été définis à l'avance sur des *données d'apprentissage* par un observateur humain. Le but est alors de généraliser la classification à un autre ensemble, celui des *données de test*.

Corpus : Nous utilisons le terme *corpus* (ou parfois aussi le pluriel latin *corpora*) dans sa plus grande généralité, pour désigner un ensemble de documents. Une branche du TAL a pour but d'exploiter informatiquement les *corpus* pour en extraire des propriétés globales non accessibles à un observateur humain.

Document, texte : Nous employons indifféremment ces deux expressions pour désigner les données d'étude, vues comme des ensembles de mots. Nous parlons également de « **sac de mots** », pour insister sur l'aspect non-ordonné (voir la section 2.1.1).

Données d'apprentissage, données de test : Pour appliquer un algorithme d'apprentissage, il peut être nécessaire d'avoir à disposition un groupe de données sur lequel le modèle sera ajusté, les *données d'apprentissage*, et d'évaluer ses performances sur un autre ensemble, celui des *données de test*. Ces deux corpus sont nécessairement disjoints à un instant donné mais un même document peut se trouver indifféremment dans le premier ou le deuxième, d'une séquence d'évaluation à l'autre. C'est le cas par exemple lors de la *validation croisée*.

Groupes, partition : Il s'agit des ensembles de documents formés lors de la classification non supervisée.

Mot, terme, forme, type : Nous désignons ainsi l'unité linguistique de base à laquelle nous nous intéresserons, en général une suite de lettres entre deux symboles de ponctuation ou espaces. Une définition plus précise dépend de la technique de **lemmatisation** employée (section 2.1.1) et est au-delà des objectifs de cette étude. L'ensemble des mots considérés constitue le **vocabulaire**.

Occurrence, token : Ces expressions sont moins connotées linguistiquement que *mot* et *terme* et réfèrent à une apparition particulière d'un *mot* dans un texte. Ainsi un document peut contenir plusieurs *occurrences* d'un même *mot*.

Plan de classement : C'est un type de structure obtenue en sortie de la phase d' *analyse exploratoire*, qui permet de ranger la plupart des documents sans ambiguïté sous un thème. Elle peut être applicable à des textes non vus au moment de l'apprentissage.

Plan hiérarchique : La classification hiérarchique consiste à diviser de façon récursive les thèmes établis en sous-thèmes. Le résultat obtenu est un arbre dont les feuilles sont les thèmes les plus spécifiques. Avec un tel plan, dit **plan hiérarchique**, le classement est alors plus facilement lisible pour un utilisateur que lorsque tous les thèmes sont au même niveau, dans un **plan à plat**.

Thème : Le *thème* est le trait commun aux textes que nous classons ensemble. L'idée est assez intuitive mais nous ne précisons pas le niveau de granularité attendu (voir la note de bas de page 2 du chapitre 1) car il dépend du corpus et de l'algorithme choisis.

Traitement Automatique de la Langue, TAL : Le *TAL* peut être caractérisé comme l'ensemble des techniques destinées à lire et interpréter par ordinateur des textes en vue d'une application donnée. La définition de ce domaine est nécessairement vague vue la diversité des utilisations. [Cori and Léon, 2002] dresse un historique de la discipline notamment à travers ses nombreux changements d'appellation.

Validation croisée, jeu : Lorsqu'un ensemble de données unique est disponible, il est d'usage de le diviser en n parties égales (n est un entier, souvent 10) et de conduire n expériences, en écartant à chaque fois une des n parties qui servira pour le test et en utilisant les $n-1$ autres pour l'apprentissage. Ce processus est connu sous le nom de *validation croisée* et permet de conduire les tests sur n *jeux* de données différents.

Notations

n_T	nombre de thèmes
n_D	nombre de documents
n_W	taille du vocabulaire
C_{wd}	termes généraux de la matrice de comptes
l_d	longueurs du document d
l	nombre d'occurrences dans le corpus d'apprentissage
$n_D{}^\star, C^\star_{wd^\star}, l^\star_{d^\star}, l^\star$	équivalents des précédents pour le corpus de test

---

$V \sim \text{Mult}(k, p)$	k tirages suivant une loi multinomiale de probabilités $(p_1, \ldots, p_n)$, avec $\sum_{i=1}^n p_i = 1$, le résultat étant placé dans un vecteur V de dimension n et de somme k
Cas particuliers :	
$V \sim \text{Mult}(k, (p_1, p_2))$	k tirages suivant une loi binomiale
$V \sim \text{Mult}(1, (p_1, p_2))$	tirage selon une loi de Bernoulli
$p \sim \text{Dir}(a)$	tirage selon une loi de Dirichlet, les a_i, ainsi que les p_i, sont n réels positifs, les p_i somment à 1
Cas particulier :	
$(p_1, p_2) \sim \text{Dir}(a_1, a_2)$	tirage selon une loi bêta
$x \sim \text{Gamma}(a, b)$	tirage selon une loi Gamma de paramètres $a, b > 0$ avec $x \in [0, \infty)$
$k \sim \text{Poisson}(a)$	tirage selon une loi de Poisson de paramètre $a > 0$ avec $k \in \mathbb{N}$

U^T	transposée de la matrice U
Γ	fonction Gamma : $\forall x \in (0, +\infty), \Gamma(x) = \int_0^\infty t^{x-1} e^{-x} dx$ (propriété : $\forall n \in \mathbb{N}, \Gamma(n+1) = n!$)
$\binom{n}{k}$	coefficient du binome $\frac{n!}{k!(n-k)!}$ pour $k \leq n$
$\lfloor x \rfloor$	partie entière de $x \in \mathbb{R}$: $\max\{n \in \mathbb{N} \mid n \leq x\}$
signe(x)	fonction signe définie pour $x \neq 0$: 1 si $x > 0$, -1 sinon
$\mathbb{1}_A$	fonction indicatrice pour un événement A : 1 si A est vrai, 0 sinon.

Annexe A

Compléments au chapitre 2

A.1 Modèle de mélange de lois binomiales négatives

Dans cette section, nous présentons un modèle de mélange de lois binomiales négatives, inspirées par les lacunes de la modélisation par loi de Poisson mises en évidence en section 2.3.7.2.

Les documents sont supposés indépendants et chaque texte $d \in \{1, \ldots, n_D\}$ résulte de l_d tirages indépendants sur le vocabulaire suivant une distribution liée au thème, lequel est une variable cachée tirée une fois par texte. D'où le modèle génératif pour un vecteur de comptes C_d :

1. Tirer un thème $T_d \sim \text{Mult}(1, (\alpha_1, \ldots, \alpha_{n_T}))$ où les α_t sont des paramètres tels que $\sum_{t=1}^{n_T} \alpha_t = 1$.
2. Pour chaque mot $w \in \{1, \ldots, n_W\}$ du vocabulaire,
   (a) tirer le compte correspondant C_{wd} selon une loi binomiale négative de paramètres (a_{wT_d}, b_{wT_d}) telle que :

$$\mathrm{P}(C_d | T_d = t) = \frac{\Gamma(C_{wd} + a_{wt})}{\Gamma(a_{wt}) C_{wd}!} b_{wt}^{C_{wd}} (1 + b_{wt})^{-C_{wd} - a_{wt}} ,$$

   a_{wt} et b_{wt} appartenant à $(0, \infty)$ pour tous $w \in \{1, \ldots, n_W\}, t \in \{1, \ldots, n_T\}$.

Les paramètres du modèle sont donc :

$$\Theta = ((\alpha_t)_{t=1,\ldots,n_T}, (a_{wt})_{w=1,\ldots,n_W, t=1,\ldots,n_T}, (b_{wt})_{w=1,\ldots,n_W, t=1,\ldots,n_T})$$

La probabilité d'un document est alors :

$$\begin{aligned} \mathrm{P}(C_d; \Theta) &= \sum_{t=1}^{n_T} \mathrm{P}(T_d = t; \Theta) \, \mathrm{P}(C_d | T_d = t; \Theta) \\ &= \sum_{t=1}^{n_T} \alpha_t \prod_{w=1}^{n_W} \frac{\Gamma(C_{wd} + a_{wt})}{\Gamma(a_{wt}) C_{wd}!} b_{wt}^{C_{wd}} (1 + b_{wt})^{-C_{wd} - a_{wt}} \end{aligned}$$

La vraisemblance du corpus est ensuite obtenue en effectuant le produit des probabilités sur tous les documents. Comme pour le modèle de mélange de multinomiales, nous souhaitons utiliser l'algorithme EM pour l'inférence. C'est la raison pour laquelle nous calculons la log-vraisemblance complète $\mathcal{L}^c$:

$$\begin{aligned}
\mathcal{L}^c &= \sum_{d=1}^{n_D} \log \mathrm{P}(C_d, T_d) \\
&= \sum_{d=1}^{n_D} \Bigg(\log \alpha_{T_d} + \sum_{w=1}^{n_W} \Bigg(\log \frac{\Gamma(C_{wd} + a_{wT_d})}{\Gamma(a_{wT_d}) C_{wd}!} \\
&\qquad + C_{wd} \log b_{wT_d} - (C_{wd} + \alpha_{wT_d}) \log(1 + b_{wT_d}) \Bigg) \Bigg) \\
&= \sum_{d=1}^{n_D} \sum_{t=1}^{n_T} \mathbb{1}_{\{T_d = t\}} \Bigg(\log \alpha_t + \sum_{w=1}^{n_W} \Bigg(\sum_{k=1}^{C_{wd}} \log(a_{wt} - 1 + k) + K \\
&\qquad + C_{wd} \log b_{wt} - (C_{wd} + a_{wt}) \log(1 + b_{wt}) \Bigg) \Bigg),
\end{aligned}$$

où nous avons appliqué la simplification suivante :

$$\begin{aligned}
\log \frac{\Gamma(C_{wd} + a_{wT_d})}{\Gamma(a_{wT_d}) C_{wd}!} &= \log(\Gamma(C_{wd} + a_{wT_d})) - \log(\Gamma(a_{wT_d})) - \log(C_{wd}!) \\
&= \sum_{k=1}^{C_{wd}} \log(a_{wt} - 1 + k) + K \,,
\end{aligned}$$

$K = -\log(C_{wd}!)$ étant une constante indépendante des paramètres que nous oublierons par la suite. Quant au premier terme, il pourra être exprimé soit comme somme de `log`s soit comme différence de `log`Γs, fonctions `gammaln` en Matlab.

L'espérance des fonctions indicatrices étant égale à la probabilité de l'événement, nous obtenons :

$$\begin{aligned}
Q &= E[\mathcal{L}^c | C; \Theta'] \\
&= \sum_{d=1}^{n_D} \sum_{t=1}^{n_T} \mathrm{P}(T_d = t | C_d; \Theta') \Bigg(\log \alpha_t + \sum_{w=1}^{n_W} \Bigg(\sum_{k=1}^{C_{wd}} \log(a_{wt} - 1 + k) \\
&\qquad + C_{wd} \log b_{wt} - (C_{wd} + a_{wt}) \log(1 + b_{wt}) \Bigg) \Bigg) \qquad \text{(A.1)}
\end{aligned}$$

Les probabilités a posteriori sont obtenues par la formule de Bayes. D'où, pour $t \in \{1, \ldots, n_T\}, d \in \{1, \ldots, n_D\}$:

$$\begin{aligned}
\mathrm{P}(T_d = t | C_d; \Theta') &= \frac{\mathrm{P}(C_d | T_d = t; \Theta') \, \mathrm{P}(T_d = t; \Theta')}{\sum_{t'=1}^{n_T} \mathrm{P}(C_d | T_d = t'; \Theta') \, \mathrm{P}(T_d = t'; \Theta')} \\
&= \frac{\alpha'_t \prod_{w=1}^{n_W} \frac{\Gamma(C_{wd} + a'_{wt})}{\Gamma(a'_{wt})} {b'_{wt}}^{C_{wd}} (1 + b'_{wt})^{-C_{wd} - a'_{wt}}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \prod_{w=1}^{n_W} \frac{\Gamma(C_{wd} + a'_{wt'})}{\Gamma(a'_{wt'})} {b'_{wt'}}^{C_{wd}} (1 + b'_{wt'})^{-C_{wd} - \alpha'_{wt'}}}
\end{aligned}$$

L'expression (A.1) n'est maximisable directement qu'en α et b, à a fixé. Nous déterminons donc les équations de réestimation de ces deux paramètres en maximisant la quantité

de l'EM[1]. Donc, pour $t \in \{1, \ldots, n_T\}$ et $w \in \{1, \ldots, n_W\}$:

$$\begin{aligned} \alpha_t &= \frac{1}{n_D} \sum_{d=1}^{n_D} \mathrm{P}(T_d = t | C_d; \Theta') \\ b_{wt} &= \frac{\sum_{d=1}^{n_D} C_{wd} \mathrm{P}(T_d = t | C_d; \Theta')}{a'_{wt} \sum_{d=1}^{n_D} \mathrm{P}(T_d = t | C_d; \Theta')} \end{aligned}$$

Pour a, la maximisation directe n'est pas possible. On applique donc simplement un pas de Newton, c'est-à-dire pour $t \in \{1, \ldots, n_T\}$ et $w \in \{1, \ldots, n_W\}$:

$$\begin{aligned} a_{wt} &= a'_{wt} - \frac{\frac{\partial Q}{\partial a_{wt}}}{\frac{\partial^2 Q}{\partial a_{wt}^2}} \\ &= a'_{wt} - \frac{\sum_{d=1}^{n_D} \mathrm{P}(T_d = t | C_d; \Theta') \left(\sum_{k=1}^{C_{wd}} \frac{1}{a'_{wt} - 1 + k} - \log(1 + b'_{wt}) \right)}{\sum_{d=1}^{n_D} \mathrm{P}(T_d = t | C_d; \Theta') \left(\sum_{k=1}^{C_{wd}} - \frac{1}{(a'_{wt} - 1 + k)^2} \right)} \end{aligned}$$

Ces formules sont appliquées de façon itérative, jusqu'à convergence. Il faut vérifier que Q ne diminue pas (ce qui garantit alors la non diminution de la log-vraisemblance) puisque la mise à jour de a n'est pas issue d'une maximisation explicite. Par ailleurs, comme dans le modèle de mélange de multinomiales, le lissage de Laplace permet d'éviter l'annulation des paramètres.

A.2 Algorithme itératif du goulot d'information

Dans cette section, nous présentons le détail de la démonstration de l'algorithme itératif du goulot d'information.

Rapellons d'abord les notations de la section 2.2.6 : W, D et T sont des variables aléatoires indicatrices respectivement des mots, des documents ou des thèmes (définis au niveau des documents) et à valeurs dans $\{1, \ldots, n_W\}$, $\{1, \ldots, n_D\}$ et $\{1, \ldots, n_T\}$. Les notations suivantes sont également introduites :

$$\begin{aligned} \alpha_t &= P(T = t) \\ \beta_{tw} &= P(W = w | T = t) \\ \mu_{dt} &= P(T = t | D = d) \end{aligned}$$

Nous supposons la probabilité jointe mots/documents connue. Pour un λ fixé, la quantité $\mathcal{Q}$, définie par :

$$\mathcal{Q} = I(T; D) - \lambda I(T; W) \,,$$

est minimisée par l'application itérative alternante des mises à jour suivantes :

$$\mu_{dt} = \frac{\alpha_t}{Z(\lambda, d)} e^{-\lambda D_{KL}(P(W=w|D=d) || P(W=w|T=t))} \tag{A.2}$$

avec $Z(\lambda, d) = \sum_{t=1}^{n_T} \alpha_t e^{-\lambda D_{KL}(P(W=w|D=d) || P(W=w|T=t))}$ constante de normalisation et $D_{KL}(P(W = w|D = d) || P(W = w|T = t))$ divergence de Kullback-Leibler :

$$\sum_{w=1}^{n_W} P(W = w|D = d) \log \frac{P(W = w|D = d)}{P(W = w|T = t)} \,.$$

[1] La normalisation des α est assurée par la technique des multiplicateurs de Lagrange.

$$\alpha_t = \sum_{d=1}^{n_D} \mu_{dt} P(D=d) \tag{A.3}$$

$$\beta_{tw} = \frac{1}{\alpha_t} \sum_{d=1}^{n_D} P(W=w, D=d) \mu_{dt} \tag{A.4}$$

Démonstration [Tishby et al., 1999] On développe d'abord l'expression $\mathcal{Q}$:

$$\begin{aligned}\mathcal{Q} &= \sum_{d=1}^{n_D}\sum_{t=1}^{n_T} P(D=d)P(T=t|D=d)\log\frac{P(T=t|D=d)}{P(T=t)} \\ &\quad -\lambda\sum_{w=1}^{n_W}\sum_{t=1}^{n_T} P(W=w)P(T=t|W=w)\log\frac{P(T=t|W=w)}{P(T=t)}\end{aligned}$$

$\alpha_t = P(T=t)$ et $\beta_{tw} = P(W=w|T=t)$ sont à présent supposées fixées. Il faut minimiser $\mathcal{Q}$ en $\mu_{dt} = P(T=t|D=d)$ selon les contraintes $\forall d \in \{1,\ldots,n_D\}, \sum_{t=1}^{n_T} \mu_{dt} = 1$, introduites avec les multiplicateurs de Lagrange supplémentaires λ_d :

$$\begin{aligned}\mathcal{Q}' &= \sum_{d=1}^{n_D}\lambda_d\Big(\sum_{t=1}^{n_T}\mu_{dt}-1\Big) + \sum_{d=1}^{n_D}\sum_{t=1}^{n_T} P(D=d)\mu_{dt}\log\frac{\mu_{dt}}{\alpha_t} \\ &\quad -\lambda\sum_{w=1}^{n_W}\sum_{t=1}^{n_T} P(W=w)P(T=t|W=w)\log\frac{P(T=t|W=w)}{\alpha_t} \\ &= \sum_{d=1}^{n_D}\lambda_d\Big(\sum_{t=1}^{n_T}\mu_{dt}-1\Big) + \sum_{d=1}^{n_D}\sum_{t=1}^{n_T} P(D=d)\mu_{dt}\log\mu_{dt} - \sum_{t=1}^{n_T}\alpha_t\log\alpha_t \\ &\quad -\lambda\sum_{w=1}^{n_W}\sum_{t=1}^{n_T} P(W=w)P(T=t|W=w)\log P(T=t|W=w) + \lambda\sum_{t=1}^{n_T}\alpha_t\log\alpha_t\end{aligned}$$

Or, par définition, pour tous t, d et w, $\alpha_t = \sum_{d=1}^{n_D}\mu_{dt}P(D=d)$ et $P(T=t|W=w) = \sum_{d=1}^{n_D}\mu_{dt}P(D=d|W=w)$. Par conséquent, en dérivant par rapport à μ_{dt} pour d et t fixés,

$$\begin{aligned}\frac{\partial\alpha_t}{\partial\mu_{dt}} &= P(D=d) \\ \frac{\partial P(T=t|W=w)}{\partial\mu_{dt}} &= P(D=d|W=w)\end{aligned}$$

On obtient donc, toujours pour d et t fixés,

$$\begin{aligned}\frac{\partial\mathcal{Q}'}{\partial\mu_{dt}} &= \lambda_d + P(D=d)(1+\log\mu_{dt}) - P(D=d)(1+\log\alpha_t) \\ &\quad -\lambda\Big(\sum_{w=1}^{n_W} P(W=w)P(D=d|W=w)(1+\log P(T=t|W=w)) \\ &\quad -P(D=d)(1+\log\alpha_t)\Big)\end{aligned}$$

$$
\begin{aligned}
&= \lambda_d + P(D=d)\log\frac{\mu_{dt}}{\alpha_t} - \lambda\sum_{w=1}^{n_W} P(W=w, D=d)\log\frac{P(T=t|W=w)}{\alpha_t} \\
&= \lambda_d + P(D=d)\left(\log\frac{\mu_{dt}}{\alpha_t} - \lambda\sum_{w=1}^{n_W} P(W=w|D=d)\log\frac{P(W=w|T=t)}{P(W=w)}\right) \\
&= \lambda_d + P(D=d)\Bigg(\log\frac{\mu_{dt}}{\alpha_t} + \lambda\sum_{w=1}^{n_W} P(W=w|D=d)\log\frac{P(W=w|D=d)}{P(W=w|T=t)} \\
&\qquad -\lambda\sum_{w=1}^{n_W} P(W=w|D=d)\log\frac{P(W=w|D=d)}{P(W=w)}\Bigg) \\
&= \lambda_d - \lambda\sum_{w=1}^{n_W} P(W=w, D=d)\log\frac{P(W=w|D=d)}{P(W=w)} \\
&\qquad +P(D=d)\left(\log\frac{\mu_{dt}}{\alpha_t} + \lambda D_{KL}(P(W=w|D=d)||P(W=w|T=t))\right),
\end{aligned}
$$

avec la définition classique de la divergence de Kullback-Leibler. L'annulation de cette dérivée partielle donne :

$$
\mu_{dt} = \frac{\alpha_t e^{-\lambda D_{KL}(P(W=w|D=d)||P(W=w|T=t))}}{\exp\frac{\lambda_d - \lambda\sum_{w=1}^{n_W} P(W=w,D=d)\log\frac{P(W=w|D=d)}{P(W=w)}}{P(D=d)}}
$$

L'expression au dénominateur étant indépendante de t, il s'agit bien d'une constante de normalisation fonction de d et λ, ce qui permet de déduire l'expression (A.2).
(A.3) et (A.4) s'obtiennent en sommant sur les documents.

Annexe B

Participation au DÉfi Fouille de Textes DEFT'05

Nous reproduisons ici l'article [Rigouste et al., 2006a] publié dans la Revue des Nouvelles Technologies de l'Information. Il traite de notre participation au DÉfi Fouille de Textes DEFT'05 [Alphonse et al., 2005] (voir aussi le site web `http://http://www.lri.fr/ia/fdt/DEFT05/`). Il montre comment le modèle de mélange de multinomiales peut être utile et conduire à une amélioration des performances sur des tâches qui ne sont pas directement liées à la classification non supervisée, mais dans lesquelles il peut s'insérer comme un module d'une chaîne de traitement plus globale.

Est également exploitée l'idée d'introduire plus d'une variable latente thématique par document, avec, ici, une chaîne de Markov sur les phrases, suggérée naturellement par la structure des données (le découpage en phrases était effectué par les organisateurs du défi).

Auteurs : Loïs Rigouste⋆, Olivier Cappé⋆, François Yvon⋆ et Fabrice Clérot†

Affiliations :
- ⋆ GET – Télécom Paris & CNRS – LTCI
  46 rue Barrault, 75634 Paris Cédex 13
  `rigouste,cappe,yvon` à `enst.fr`
- † France Télécom Division R & D TECH/SUSI/TSI
  2 Avenue Pierre Marzin, 22307 Lannion Cédex
  `fabrice.clerot` à `francetelecom.com`

Résumé

Dans cet article, nous montrons comment des outils génériques de la fouille statistique de textes peuvent être utilisés pour résoudre une tâche d'apprentissage supervisée : le DÉfi Fouille de Textes 2005. Dans un premier temps, nous étudions comment capturer une partie des spécificités de la tâche à l'aide de modèles de Markov cachés. Nous détaillons ensuite une modélisation des textes par un mélange de distributions multinomiales sur les comptes de mots, dans laquelle chaque composante correspond à un thème particulier. Les paramètres des distributions thématiques sont estimés grâce à l'algorithme EM. Ce modèle est utilisé pour diviser en sous-thèmes les discours des deux présidents. Nous discutons finalement des performances obtenues en combinant ces deux outils.

Abstract

In this contribution, we show how we used generic probabilistic text mining tools to solve a supervised task : the DÉfi Fouille de Textes 2005. We first explain how the specificities of the task can be captured in the form of Hidden Markov Models. Then we present a probabilistic approach for text clustering, which models texts by a mixture of multinomial distributions over the word counts, where each component corresponds to a different theme. We apply the EM algorithm to estimate the parameters of these thematic distributions. This model is used to thematically subdivide the available training corpus in an unsupervised manner. We finally present and discuss the performance obtained using the combination of these tools.

B.1 Introduction

La tâche DEFT, introduite plus en détail dans ce même numéro, consiste à analyser un pseudo-document construit en insérant, dans un discours de Jacques Chirac, un fragment de discours de François Mitterrand. Il s'agit, pour les participants, de séparer le document original de l'insert éventuel. Ils peuvent, à cette fin, s'appuyer sur un corpus de pseudo-documents annotés par les organisateurs.

Cette tâche se prête *a priori* à plusieurs approches :

- l'identification *non-supervisée* de segments thématiquement homogènes dans les pseudo-documents, problème pour lequel de multiples méthodes sont disponibles (voir, par exemple, [Hearst, 1997, Choi, 2000]), qui toutes essaient de tirer parti de l'organisation séquentielle du texte. Cette démarche est confortée par la méthodologie de constitution de la base de données, selon laquelle les insertions de discours de François Mitterrand traitent de thématiques différentes de celles des discours de Jacques Chirac. Cette stratégie a pour inconvénient de ne pas réellement exploiter les données de supervision disponibles.
- la classification *supervisée* des phrases dans deux catégories, une pour chaque président. Cette tâche est bien documentée dans la littérature et il existe de nombreux outils permettant de la résoudre (voir, par exemple, [Sebastiani, 2002] pour une revue). Cette approche, qui permet de tirer effectivement parti des données de supervision, se heurte à une double difficulté. D'une part, les fragments à catégoriser sont courts (des phrases isolées), ce qui fragilise les systèmes de catégorisation ; d'autre part, on peut s'attendre à ce que les deux classes soient « proches », dans la mesure où, d'un point de vue purement thématique, les échantillons de phrases des deux présidents abordent globalement des sujets très voisins. Une variante, permettant de répondre partiellement à la seconde objection consisterait à fonder la discrimination sur des attributs purement stylistiques (longueur des phrases, fréquence d'emplois de certains marqueurs lexicaux ou syntaxiques), en faisant abstraction des mots sémantiquement pleins. Cette approche est caractéristique des travaux portant sur l'identification d'auteurs en lexicométrie [Benzécri et al., 1981].

La plupart des participants à DEFT'05 ont proposé des méthodes empruntant à ces deux démarches. La cohérence de la segmentation est assurée soit par des modèles de Markov cachés (Hidden Markov Models, HMM) [Jelinek, 1997], soit par les outils de segmentation cités plus haut, alors que la tâche de classification supervisée fait l'objet de traitements plus variés : classifieurs bayésiens avec extraction de divers attributs [El-Bèze et al., 2005, Labadié et al., 2005], analyse de dépendances syntaxiques [Maisonnasse and

Tambellini, 2005] ou machines à vecteurs supports [Kerloch and Gallinari, 2005] sont des exemples parmi d'autres.

L'approche que nous avons mise en œuvre suit également cette idée de combinaison de segmentation et de classification, en exploitant des outils *génériques* de la fouille de texte, réutilisables dans de nombreux autres contextes. Elle utilise, d'une part, des techniques de catégorisation et de classification probabilistes, qui se fondent sur une représentation en « sac-de-mots » des phrases. Ainsi, la phase d'apprentissage consiste à inférer les paramètres de modèles multi-thématiques des discours de chaque président. Notre approche vise, d'autre part, à tirer profit des différentes contraintes de la tâche, et en particulier celles qui se déduisent de la méthode de constitution des pseudo-documents, qui sont exprimées par des automates finis. L'ensemble est combiné sous la forme de HMMs. La figure B.1 illustre l'approche dans sa globalité.

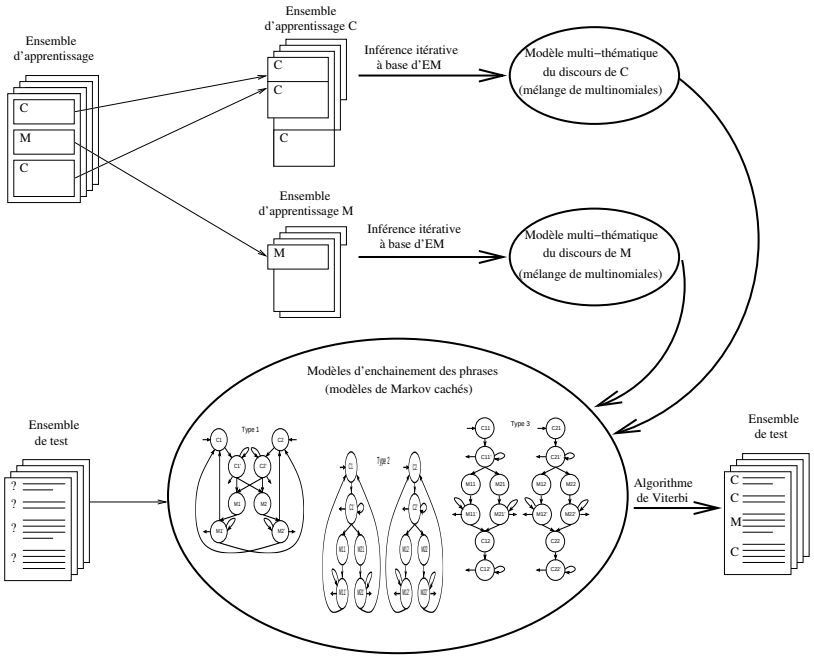

FIG. B.1 – Schéma global de l'approche

Cet article est organisé comme suit. Nous décrivons à la section B.2 le principe général de l'algorithme de segmentation, qui repose essentiellement sur la technologie des HMMs. Nous montrons comment, à partir d'un modèle simple, il est possible d'intégrer progressivement les différentes contraintes de la tâche en utilisant des structures de plus en plus élaborées. Cette section se termine par l'adaptation des HMMs au cas où les discours de chaque président peuvent être représentés par plusieurs thèmes chacun et nous illustrons l'incorporation de cette multithématicité aux HMMs. La section B.3 présente le modèle utilisé pour identifier de manière non-supervisée des sous-thèmes au sein des discours des deux présidents. Dans cette même section, nous analysons les problèmes d'estimation que pose ce modèle et proposons une solution originale pour y faire face. La section B.4 contient une présentation complète du système utilisé pour la tâche DEFT et des performances qu'il nous a permis d'obtenir. Une analyse plus détaillée des résultats, permettant d'apprécier la contribution des différents outils utilisés aux résultats est également proposée. Cet article se conclut par une discussion des voies d'amélioration du système, dans la perspective

d'autres tâches de fouille de texte.

B.2 Modèles de Markov pour la segmentation

L'approche que nous avons retenue combine deux outils de base de la fouille de textes, à savoir d'une part les modèles de Markov pour les séquences ; d'autre part les modèles de classification probabilistes non-supervisés. Dans cette section, nous discutons la conception et la mise en œuvre de modèles de Markov dont la topologie est spécifiquement adaptée à la tâche de segmentation DEFT. Partant d'un modèle très simple envisageant la segmentation sous l'angle de la catégorisation, nous introduisons progressivement des modèles plus complexes, qui prennent en compte les différentes contraintes de la tâche.

B.2.1 Un modèle simpliste de catégorisation

Le modèle le plus simple implantant les principes que nous avons retenus considère chaque pseudo-document comme une succession ordonnée de vecteurs C_p multidimensionnels (un par phrase). Chaque vecteur C_p contient le nombre d'occurrences $C_p(w)$, dans la phrase p (de longueur l_p), de chacun des mots w d'un vocabulaire d'indexation prédéfini (de taille n_W). La tâche consiste alors à répartir ces vecteurs en deux classes : la classe C (les phrases attribués à J. Chirac) et la classe M (celles de F. Mitterrand).

Le classifieur dit « Bayésien Naïf » [Lewis, 1998, McCallum and Nigam, 1998] permet d'effectuer, phrase par phrase, cette affectation à partir du modèle probabiliste suivant. Il s'agit de déterminer la classe $y^* \in \{C, M\}$ qui est la plus probable compte tenu de l'observation courante C_p, soit :

$$y^* = \arg\max_y P(Y = y|C_p) = \arg\max_y P(C_p|Y = y)P(Y = y) \tag{B.1}$$

Chaque vecteur est considéré comme la réalisation d'un tirage d'une loi multinomiale. Si les valeurs des paramètres $\{\beta_{wy}, w = 1 \dots n_W, y \in \{C, M\}\}$, pour chacune des lois associées à ces deux classes, sont supposées connues, il vient :

$$P(C_p|Y = y) = \frac{l_p!}{\prod_{w=1}^{n_W} C_p(w)!} \beta_{wy}^{C_p(w)}$$

Cette formule exprime simplement le fait que chaque occurrence d'un mot w dans la phrase C_p contribue à la probabilité globale par un facteur β_{wy}. Intuitivement, plus ce facteur est grand, plus le mot w est "important" pour la classe y. Le rapport de factorielles est le facteur de normalisation classique des lois multinomiales (qui ne joue aucun rôle dans la classification car il ne dépend pas des paramètres de classes).

Si la valeur des probabilités *a priori* $P(Y = y), y \in \{C, M\}$, pour chacune des deux classes est également connue, il devient possible d'utiliser (B.1) pour ventiler les phrases.

Ce classifieur fournit exactement la même segmentation que la mise en œuvre du décodage par l'algorithme de Viterbi dans le Modèle de Markov représenté à la figure B.2, sous l'hypothèse que :

- la loi d'émission associée à chaque état est $P(C_p|Y = y)$, calculée selon les principes exposés ci-dessus.
- les probabilités de transition entre états, ainsi que les probabilités initiales et finales ne dépendent que des probabilités a priori $P(Y = y)$[1].

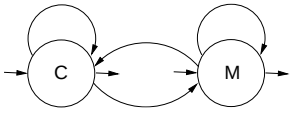

FIG. B.2 – HMM de base
Les états ayant une transition entrante depuis les états initiaux sont identifiés par une flèche entrante ; ceux qui ont une transition vers l'état final ont une flèche sortante.

Nous allons voir dans ce qui suit qu'il est possible de préserver ce lien entre classifieurs et HMMs pour l'enchaînement des phrases tout en ajoutant de la complexité dans les deux étapes, avec l'ajout de sous-thèmes dans la classification (Section B.3) d'une part et avec l'intégration de contraintes dans les transitions entre états (Sous-sections B.2.2 et B.2.3) d'autre part.

B.2.2 Intégration des contraintes de la tâche

L'approche décrite ci-dessus ignore entièrement le caractère séquentiel de la tâche, en posant l'hypothèse que chaque phrase est statistiquement indépendante des autres phrases du même pseudo-document. Cette hypothèse est très inappropriée, dans la mesure où les documents s'organisent comme une succession de passages relativement longs d'un même auteur. En jouant sur les probabilités de transition entre états de manière à (i) favoriser une des deux classes et (ii) pénaliser les changements de classes, il est aisé d'améliorer un peu le modèle de la figure B.2 pour aboutir à un segmenteur fournissant un étiquetage plus cohérent temporellement.

Cette approche ne permet toutefois pas de respecter différentes contraintes qui constituent pourtant des informations intéressantes sur la tâche :

1. Un texte commence toujours par une phrase de la classe C.
2. Chaque fragment contient toujours au moins deux phrases de la même classe (pas de phrase de F. Mitterrand isolée et au moins deux phrases de J. Chirac en début de texte).
3. Chaque pseudo-document contient au plus une insertion d'un bloc de phrases de la classe M[2].

Il est possible d'intégrer progressivement ces contraintes en conservant l'architecture générale du système. La première s'exprime simplement par le fait que la probabilité initiale de l'état C est 1. La seconde se modélise en dupliquant chaque état (et les lois d'émission associées) pour donner lieu au modèle de la figure B.3.(a), dans lequel les états C et C' (respectivement M et M') sont des clones de l'état C (respectivement M) de la figure B.2. Dans ce nouveau modèle, tout fragment est ainsi contraint à « consommer » au moins deux phrases. La troisième contrainte est rendue explicite dans le modèle représenté Figure B.3.(b), qui contient maintenant 4 clones de l'état C : $C1$ et $C1'$, qui modélisent les phrases de la classe C pré-insertion, et $C2$ et $C2'$ pour les phrases post-insertion.

[1]Ceci est vrai à un détail technique près, qui est que les probabilités finales sont liées de façon plus ou moins directe à la longueur du texte. Mais cette dépendance est accessoire dans l'analogie que nous établissons ici.

[2]Cette contrainte ne figure pas explicitement dans la description de la tâche DEFT. Il nous a semblé intéressant de l'inclure, dans la mesure où tous les documents du corpus d'apprentissage la respectent.

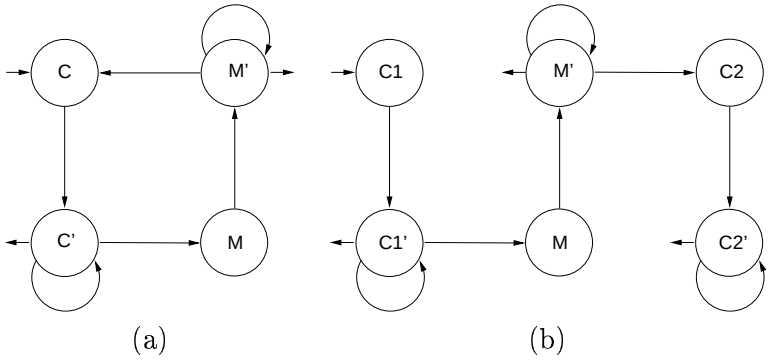

Fig. B.3 – HMMs avec contraintes

B.2.3 Subdivision des classes C et M

Un examen attentif de la méthode de constitution des corpus d'apprentissage et de test révèle une autre information, qui est que l'ensemble des discours se ventile globalement en deux grands thèmes : 'national' et 'international' et que la procédure de constitution des corpus a veillé à ce que chaque fragment ajouté (classe M) diffère thématiquement du discours dans lequel il s'insère : soit un fragment M-'national' dans un discours C-'international', soit le contraire. Cette autre clé de répartition des discours n'est malheureusement pas fournie par les organisateurs : pour utiliser cette information, il faudra donc, d'une certaine manière, la reconstruire de façon non-supervisée. Nous verrons précisément comment cela est possible à la section B.3.

Supposons pour l'instant que cette information soit disponible, modélisée par quatre (et non plus deux) classes différentes : CN, CI, MN, et MI, à chacune desquelles est associé un modèle multinomial différent. La segmentation du texte est alors opérée par le modèle de la figure B.4. En intégrant les trois contraintes décrites dans la section précédente, nous aboutirions à un modèle dont la topologie contiendra deux copies du modèle B.3.(b), l'une pour les séquences $CN - MI - CN$, l'autre pour les séquences $CI - MN - CI$.

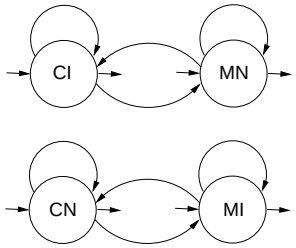

Fig. B.4 – HMMs avec contraintes et thèmes

Dans la mesure où la répartition des documents en thèmes est inconnue, les modèles que nous avons finalement utilisés sont moins contraints. Ils reposent en effet sur l'hypothèse que les discours de J. Chirac se répartissent en n_C thèmes, et ceux de F. Mitterrand en n_M thèmes, l'observation selon laquelle 'national' et 'international' sont alternés se traduisant plus généralement par :

4. Toute transition directe entre deux thèmes du même auteur est interdite (à l'exception des insertions, chaque texte est donc supposé monothématique).
5. L'insertion d'un fragment de F. Mitterrand sépare deux fragments de discours de

J. Chirac qui appartiennent au même thème.

Ces contraintes s'ajoutent à celles que nous avons énoncées dans la section précédente.

Pour valider l'apport en termes de performances de ces différentes hypothèses, nous avons finalement développé et testé 3 modèles, appelés "types" 1, 2 et 3 dans ce qui suit. Ils respectent tous les conditions 1, 2 et 4 mais :

- le type 1 ne tient pas compte des contraintes 3 (au plus une insertion M) et 5 (cohérence sous-thématique) ;
- le type 2 respecte la contrainte 5 mais pas 3 ;
- le type 3 respecte l'ensemble des contraintes discutées ci-dessus.

Ces modèles sont représentés sur les figures B.5 et B.6 dans le cas simple où $n_M = n_C = 2$.

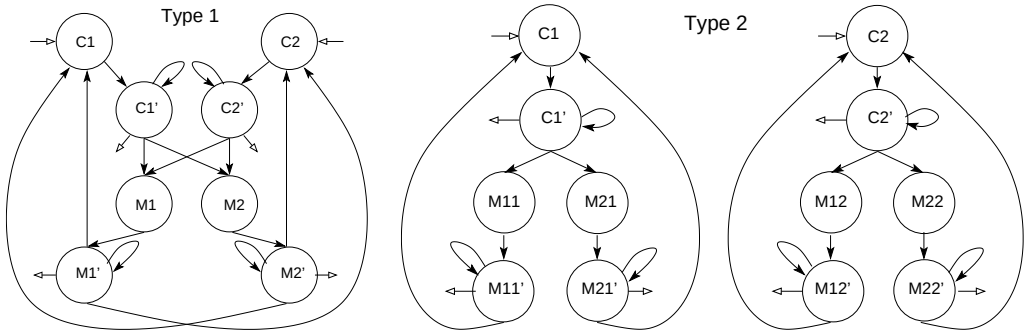

FIG. B.5 – Contraintes de "mono-thématicité" et de "retour" dans un même thème
Modèles incluant les contraintes de non transition d'un thème à un autre pour un passage donné d'un interlocuteur (1) et d'identité des thèmes pré et post-insertion (2).

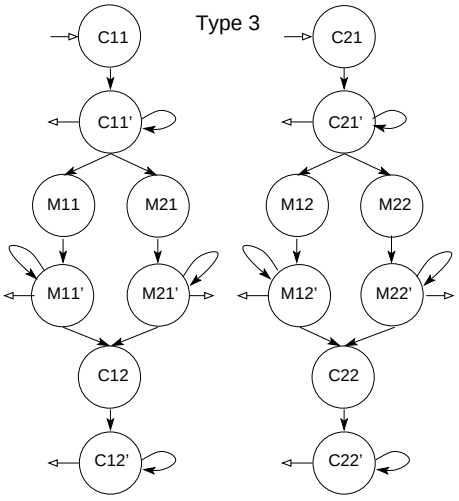

FIG. B.6 – Contrainte d'insertion unique
Modèle respectant, en plus de toutes les autres, la contrainte d'insertion unique.

La mise en œuvre de ces modèles requiert deux types de ressources : les paramètres des lois d'émission multinomiales, dont l'estimation est discutée à la section B.3, et les probabilités de transition du HMM, qui sont calculées comme suit. Pour les probabilités de transition, nous multiplierons les constantes de changement d'auteur (p_{C2M} et p_{M2C}, fixées

à 0.3) par le paramètre α_t correspondant au thème vers lequel la transition s'effectue[3]. Les probabilités de sortie ont été en général fixées à 0, à l'exception notable du type 3 où les probabilités de sortie des états C post-insertion doivent être augmentées et fixées à p_{C2M}. En effet, dans le cas contraire, on favorise les états post-insertion par rapport aux états pré-insertion et la vraisemblance est alors presque toujours maximisée en affectant les deux premières phrases à J. Chirac, les deux suivantes à F. Mitterrand et toutes les autres à J. Chirac (seule configuration admissible qui maximise le nombre de paragraphes dans les états post-insertion). Les probabilités de rester dans le même état (boucle) sont calculées pour que la somme des probabilités de transition pour chaque état soit égale à 1.

B.3 Modèle de mélange de multinomiales

Nous exposons et discutons à présent la méthodologie utilisée pour construire, de manière non-supervisée, des sous-thèmes caractérisés par des distributions multinomiales à partir d'une collection de documents. Nous utilisons dans cette section de façon générique le terme document, mais il se peut très bien que chaque vecteur de comptes représente un paragraphe ou une phrase. Après une rapide présentation du modèle, nous discutons plus longuement des problèmes d'estimation et des solutions mises en œuvre.

Aux côtés des méthodes classiques de classification non-supervisée, telles que l'algorithme des K-moyennes ou l'Analyse en Composantes Principales (ou une variante proche : l'Analyse Sémantique Latente [Deerwester et al., 1990]), des méthodes probabilistes ont trouvé leur place pour l'analyse exploratoire de données textuelles, les modèles les plus populaires étant probablement *Probabilistic Latent Semantic Analysis* [Hofmann, 2001] et *Latent Dirichlet Allocation* [Blei et al., 2002]. À l'instar de ces auteurs, nous nous plaçons ici dans le domaine des statistiques paramétriques et utilisons un modèle de mélange dont les variables latentes ont une signification *thématique*. Les paramètres de ces modèles ont ainsi une interprétation simple et l'on peut associer à chaque thème une distribution sur le vocabulaire qui identifie les mots les plus représentatifs pour ce thème. De manière duale, le résultat du partitionnement est, pour chaque document, un vecteur probabilisé d'appartenance aux différents thèmes, qui peut s'interpréter comme une projection du document dans un espace de faible dimension.

Nous considérons ici le plus simple de ces modèles probabilistes [Nigam et al., 2000, Clérot et al., 2004], dans lequel chaque document est supposé monothématique : ce modèle est ainsi directement compatible avec la tâche DEFT, dans lequel chaque phrase est supposée appartenir à un thème unique. Après avoir présenté ce modèle, nous donnons les équations d'estimation du Maximum A Posteriori, via l'algorithme Expectation Maximization (EM). Nous évoquons ensuite quelques résultats qui montrent l'importance de l'initialisation et suggérons une méthode heuristique pour l'inférence des paramètres, qui est celle que nous avons finalement utilisée pour les évaluations.

B.3.1 Le modèle génératif

Nous supposons toujours que les textes sont représentés par des « sacs-de-mots », c'est-à-dire que le vocabulaire est connu et fini et que chaque document est représenté par un

[3]Les α_t correspondent aux poids du mélange ; le problème de leur estimation sera également traité en section B.3. Pour le moment, précisons simplement qu'il existe un α_t par sous-thème, donc $n_M + n_C$ au total, et que les n_M paramètres correspondant aux sous-thèmes M somment à 1, ainsi que les n_C paramètres correspondant aux sous-thèmes C.

vecteur de comptes sur cet ensemble. Conformément aux notations introduites en B.2.1, n_D, n_T et n_W représentent respectivement le nombre de documents dans le corpus d'apprentissage, le nombre de thèmes (i.e. le nombre de composantes du modèle de mélange) et la taille du vocabulaire.

Pour $d \in \{1, \ldots, n_D\}$ et $w \in \{1, \ldots, n_W\}$, on note $C_d(w)$ le terme général de la matrice de comptes, c'est-à-dire le nombre d'occurrences du mot w dans le document d'indice d. Désignons également par $l_d = \sum_{w=1}^{n_W} C_d(w)$ le nombre d'occurrences dans le texte d et par $l = \sum_{d=1}^{n_D} l_d$ le nombre total d'occurrences dans le corpus, somme de tous les termes de la matrice de comptes.

Contrairement au cadre de la catégorisation supervisée, une hypothèse faite ici est qu'aucune information sur les affectations des documents aux différents thèmes n'est à notre disposition. Le modèle de génération du corpus présenté ci-dessous permet, après estimation des paramètres, de proposer une ventilation des documents suivant les différentes composantes du mélange.

On suppose que les textes sont indépendants. Le document d ($d \in \{1, \ldots, n_D\}$) résulte de l_d tirages indépendants sur le vocabulaire selon une distribution dépendant du thème, ce dernier étant défini par une variable cachée tirée une fois par texte. Notons $\text{Mult}(k, (p_1, \ldots, p_n))$ l'opération consistant à tirer k fois suivant une multinomiale de probabilités $(p_1, \ldots, p_n)$. D'où le modèle génératif pour un document :

- Tirer un thème $T \sim \text{Mult}(1, (\alpha_1, \ldots, \alpha_{n_T}))$ où les α_t sont des paramètres tels que $\sum_{t=1}^{n_T} \alpha_t = 1$.
- Tirer l_d mots $C_d = (C_{d1}, \ldots, C_{dn_W}) \sim \text{Mult}(l_d, (\beta_{1t}, \ldots, \beta_{n_W t}))$, β étant une matrice $n_W \times n_T$ de paramètres telle que $\forall t \in \{1, \ldots, n_T\}, \sum_{w=1}^{n_W} \beta_{wt} = 1$.

La probabilité d'un document est alors, en notant T_d la variable indicatrice du thème latent :

$$\begin{aligned} P(C_d; \alpha, \beta) &= \sum_{t=1}^{n_T} P(T_d = t; \alpha, \beta) P(C_{d1}, \ldots, C_{dn_W} | T_d = t; \alpha, \beta) \\ &= \sum_{t=1}^{n_T} P(T_d = t; \alpha, \beta) l_d! \prod_{w=1}^{n_W} \frac{P(w | T_d = t; \alpha, \beta)^{C_d(w)}}{C_d(w)!} \\ &= \frac{l_d!}{\prod_{w=1}^{n_W} C_d(w)!} \sum_{t=1}^{n_T} \alpha_t \prod_{w=1}^{n_W} \beta_{wt}^{C_d(w)} \end{aligned}$$

Ainsi chaque thème contribue à la probabilité globale du document par sa probabilité a priori α_t et, pour chaque occurrence du texte, par la probabilité β_{wt} d'émission du mot w dans le thème en question.

La probabilité du corpus, ou vraisemblance des observations, est obtenue en réalisant le produit de l'expression ci-dessus pour l'ensemble des documents étudiés. Cependant, il n'est pas possible d'établir directement une expression d'un estimateur de maximum de vraisemblance. C'est pourquoi nous faisons appel à l'algorithme EM (Expectation Maximization) qui repose sur le calcul de l'espérance, conditionnellement aux observations, de la log-vraisemblance *complète* $\mathcal{L}^c$, c'est-à-dire la log-vraisemblance des couples vecteur de comptes C_d et thème T_d, définie par :

$$\mathcal{L}^c = \sum_{d=1}^{n_D} \log P(C_d, T_d)$$

$$= \sum_{d=1}^{n_D} \left(\log P(T_d) + \log P(C_d|T_d) \right)$$
$$= \sum_{d=1}^{n_D} \sum_{t=1}^{n_T} \mathbb{1}_{\{T_d=t\}} \left(\log \alpha_t + \sum_{w=1}^{n_W} C_d(w) \log \beta_{wt} \right) + K$$

où K est une constante indépendante des paramètres (que nous oublierons par la suite). La notation $\mathbb{1}_A$ désigne la fonction indicatrice définie par :

$$\mathbb{1}_A = \begin{cases} 1 & \text{si } A \text{ est vrai ;} \\ 0 & \text{sinon.} \end{cases}$$

L'espérance, conditionnellement aux observations, et tenant compte des paramètres α', β' issus de l'itération précédente, s'écrit :

$$E[\mathcal{L}^c] = \sum_{d=1}^{n_D} \sum_{t=1}^{n_T} P(T_d = t|C_d; \alpha', \beta') \times \left(\log \alpha_t + \sum_{w=1}^{n_W} C_d(w) \log \beta_{wt} \right)$$

Les probabilités a *posteriori* sont données par la formule de Bayes, conduisant, pour $t \in \{1, \ldots, n_T\}, d \in \{1, \ldots, n_D\}$, à :

$$\begin{aligned} P(T_d = t|C_d; \alpha', \beta') &= \frac{P(C_d|T_d = t; \alpha', \beta') P(T_d = t; \alpha', \beta')}{P(C_d; \alpha', \beta')} \\ &= \frac{P(C_d|T_d = t; \alpha', \beta') P(T_d = t; \alpha', \beta')}{\sum_{t'=1}^{n_T} P(C_d|T_d = t'; \alpha', \beta') P(T_d = t'; \alpha', \beta')} \\ &= \frac{\alpha'_t \prod_{w=1}^{n_W} {\beta'_{wt}}^{C_d(w)}}{\sum_{t'=1}^{n_T} \alpha'_{t'} \prod_{w=1}^{n_W} {\beta'_{wt'}}^{C_d(w)}} \end{aligned} \quad \text{(B.2)}$$

Le numérateur, produit de la probabilité a priori du thème t et de l'"importance" dans le thème de chaque occurrence du mot w dans le texte considéré, a déjà été identifié précédemment comme mesurant intuitivement la probabilité jointe du document C_d et du thème t. Le dénominateur vient de l'opération de normalisation correspondant à $\sum_{t=1}^{n_T} P(T_d = t|C_d; \alpha', \beta')$.

Il est alors possible de déterminer les équations de ré-estimation des paramètres en maximisant la quantité de l'EM, avec la technique des multiplicateurs de Lagrange pour normaliser de façon appropriée les paramètres α (le vecteur somme à 1) et β (chaque colonne somme à 1). On obtient, pour $t \in \{1, \ldots, n_T\}$ et $w \in \{1, \ldots, n_W\}$:

$$\alpha_t = \frac{1}{n_D} \sum_{d=1}^{n_D} P(T_d = t|C_d; \alpha', \beta') \quad \text{(B.3)}$$

$$\beta_{wt} = \frac{\sum_{d=1}^{n_D} C_d(w) P(T_d = t|C_d; \alpha', \beta')}{\sum_{w=1}^{n_W} \sum_{d=1}^{n_D} C_d(w) P(T_d = t|C_d; \alpha', \beta')} \quad \text{(B.4)}$$

Ces formules ont elles aussi une interprétation intuitive simple si les probabilités d'appartenance $P(T_d = t|C_d; \alpha', \beta')$ sont exactement 0 ou 1 (chaque texte "appartient" alors à un thème et un seul) :

- On obtient α_t en comptant le nombre de documents dans le thème t puis en normalisant.

– On détermine la nouvelle valeur de β_{wt} en dénombrant le nombre d'occurrence du mot w dans les textes correspondant au thème t, puis on normalise sur l'ensemble des mots.

Cette interprétation peut être étendue au cas où les probabilités d'appartenance ne sont pas binaires. Chaque texte contribue alors au renouvellement des paramètres en proportion de son "implication" dans le thème.

L'algorithme EM consiste à appliquer les formules (B.2), (B.3) et (B.4) de façon itérative jusqu'à convergence.

Lorsqu'un mot w n'est jamais observé dans un thème t, ces formules conduisent à une estimation nulle pour β_{wt}. Il est alors nécessaire de recourir à des techniques de lissage des estimateurs, pour rendre compte du fait que, même si, dans l'ensemble d'apprentissage, un mot n'a jamais été vu en conjonction avec un thème donné, son apparition dans ce thème n'est pas totalement impossible (elle est néanmoins de probabilité très faible). Dans la suite, nous utilisons un lissage de Laplace, consistant à augmenter tous les comptes de 0.1. Dans le cadre probabiliste, ce lissage peut être interprété comme correspondant à l'algorithme EM associé au maximum *a posteriori* (et non plus au maximum de vraisemblance) lorsque les paramètres β sont munis d'une distribution *a priori* de type Dirichlet, de paramètre 1.1 [Rigouste et al., 2005a].

B.3.2 Méthode d'inférence itérative par ajout de mots rares

Les équations de ré-estimation posées, il reste encore une marge de manœuvre importante pour un expérimentateur désirant inférer les paramètres du modèle. Des questions pertinentes concernent notamment :

– le choix du vocabulaire : faut-il considérer le vocabulaire en entier ou retirer les mots trop rares ou trop fréquents ?
– l'initialisation du modèle

Ces interrogations sont étudiées en détail dans [Rigouste et al., 2005b] : nous résumons dans un premier temps les conclusions de ces travaux qui nous semblent pertinentes pour DEFT, avant de présenter la méthode d'inférence finalement utilisée pour les tests.

B.3.2.1 Expérimentations préliminaires

Le corpus qui a servi de base à ces expérimentations préliminaires est un corpus raisonnablement simple, issu de Reuters 2000[4] et composé de 5000 textes équirépartis dans 5 catégories (arts, sports, emploi, catastrophes, santé) [Reuters, 2000]. En plus de la log-vraisemblance (ou de manière équivalente, de la perplexité[5]) calculée lors de l'apprentissage, nous considérons également une autre mesure des performances : *l'information mutuelle* entre le classement produit par le modèle et les catégories du corpus Reuters. Ce critère mesure plus directement la faculté de l'algorithme à retrouver les regroupements d'origine.

Nos expériences nous ont permis de mettre en évidence le fait que la phase d'initialisation de l'algorithme EM est cruciale pour obtenir des regroupements pertinents des documents. Elles ont également confirmé l'intuition suivante : en l'absence d'information *a priori* sur les thèmes à trouver, la meilleure initialisation consiste à partir de regroupements qui se recoupent largement, l'apprentissage se chargeant en général de les séparer. Dans cet

[4]Le corpus est en anglais. Savoir si les conclusions de notre étude se transposent à un autre corpus, en français, comme nous l'avons supposé, reste une question ouverte.

[5]La perplexité est définie comme l'exponentielle de la valeur moyenne (par mot) de la log-vraisemblance [Jelinek, 1997].

esprit, l'algorithme est initialisé en fixant les probabilités *a posteriori* pour un document d'appartenir à un thème – équation (B.2) – très proches de l'équiprobabilité entre tous les thèmes. Pour chaque essai, nous tirons donc ces valeurs selon une distribution Dirichlet dont la moyenne est la distribution uniforme et dont la variance est faible. C'est-à-dire que si l'on fixe un paramètre $\lambda > 0$ grand, les probabilités a posteriori sont tirées pour chaque document selon une densité g telle que : $g(p_1, \ldots, p_{n_T}) \propto \prod_{t=1}^{n_T} {p_t}^{\lambda-1}$

Ainsi, plus λ est grand, plus on obtient des probabilités proches de l'équiprobabilité : on peut en effet montrer que la moyenne de chaque p_t est $\frac{1}{n_T}$ et que sa variance est équivalente à $\frac{1}{\lambda}$ lorsque $\lambda \gg n_T$. Par la suite, nous désignerons ce procédé sous le nom d'initialisation "Dirichlet".

Afin d'avoir une idée de la meilleure performance possible, nous avons également essayé d'introduire l'information de supervision disponible, consistant à baser l'initialisation sur les catégories Reuters. Pour ce faire, l'étape d'initialisation donne à un document d de catégorie Reuters t une valeur de 1 à la probabilité *a posteriori* d'appartenir au thème t, et une valeur 0 pour tous les autres thèmes.

La comparaison systématique de ces différentes procédures d'initialisation a conduit à établir les constats suivants :

- la variabilité entre les deux initialisations est très forte pour les deux mesures : log-vraisemblance et information mutuelle.
- la log-vraisemblance mesurée sur l'ensemble d'apprentissage est un indicateur raisonnable de la qualité finale du regroupement produit, dans le sens où ses variations sont comparables à celles de la log-vraisemblance sur les données de test ou de l'information mutuelle sur les données d'apprentissage ou de test.
- à moins de pouvoir les initialiser correctement (ce qui est impossible sans information de supervision), les mots rares nuisent en général à l'apprentissage et l'écart entre les deux initialisations diminue lorsque l'on réduit la taille du vocabulaire en ne conservant que les mots les plus fréquents.

B.3.2.2 Une nouvelle stratégie d'inférence

Sur la base de ces observations, la méthode d'inférence finalement retenue repose sur le principe d'une augmentation progressive de la taille du vocabulaire d'indexation (une présentation plus complète est donnée dans [Rigouste et al., 2005b]). Son principe est le suivant : partant d'un vocabulaire extrêmement réduit (constitué des 1000 mots les plus fréquents, soit 2% du vocabulaire total), une première estimation des paramètres du modèle est obtenue avec l'initialisation "Dirichlet". Ce procédé est répété plusieurs fois et seul le meilleur ensemble de paramètres (au sens de la log-vraisemblance finale) est conservé. Au terme de cette étape, nous disposons donc d'une valeur pour un (petit) sous-ensemble des paramètres β_{wt} correspondant aux mots les plus fréquents. Il est possible d'en déduire, par application de l'étape M de l'algorithme EM (équation (B.2)), des probabilités a posteriori d'appartenance aux thèmes pour *tous les documents*. Après augmentation de la taille du vocabulaire, nous prenons alors soin d'utiliser une initialisation "Dirichlet" dont l'espérance n'est plus maintenant uniforme, mais égale aux probabilités a posteriori de l'étape précédente. L'algorithme EM peut alors être mis en œuvre pour ré-estimer les paramètres β_{wt} pour un plus grand ensemble de mots, desquels seront déduites de nouvelles probabilités a posteriori, etc. Cette procédure est itérée jusqu'à ce que le vocabulaire complet soit finalement pris en compte.

Les résultats présentés dans [Rigouste et al., 2005b] montrent que l'algorithme d'inférence itératif parvient à atteindre les mêmes valeurs de vraisemblance que celles obtenues en initialisant avec les informations de supervision. L'information mutuelle est un peu moins bonne, montrant que la corrélation entre les deux indicateurs n'est pas absolue, mais se situe dans des plages de valeurs beaucoup plus satisfaisantes qu'avec l'initialisation "Dirichlet" simple.

B.4 Utilisation du segmenteur en thèmes pour DEFT

L'idée directrice de notre méthode est qu'il devrait être plus facile d'identifier les ruptures thématiques entre les phrases prononcées par J. Chirac et celles de F. Mitterrand si l'on connaît précisément les différents sujets abordés par chaque locuteur. Nous pensons (et cela se confirme dans la dernière section) que le résultat sera meilleur en modélisant les discours de chaque président par plusieurs thèmes, qui lui sont propres, plutôt qu'en utilisant seulement un thème pour chaque personne.

Ainsi, nous utilisons les données d'apprentissage pour estimer les paramètres relatifs aux thèmes abordés par J. Chirac et à ceux abordés par F. Mitterrand. Une fois ces paramètres identifiés, nous utilisons l'algorithme de Viterbi sur les phrases du corpus de test pour déterminer le thème (et donc l'auteur) le plus vraisemblable pour chaque phrase.

B.4.1 Prétraitements

Rappelons que la campagne DEFT inclut trois tâches, soit, par ordre de difficulté croissante : la tâche 3 consiste à segmenter des textes "bruts" ; la tâche 2 des textes dans lesquels les dates sont remplacées par un tag unique <date> ; la tâche 1 des textes dans lesquels ont également été normalisés et remplacés par le tag <nom> les noms de personnes. Pour chacune des trois tâches, la même série de prétraitements des corpus a été utilisée, consistant à segmenter chaque phrase en mots, à normaliser les chiffres, à mettre tous les mots en minuscules et à supprimer toutes les marques de ponctuation.

À l'issue de ces traitements, le vocabulaire utilisé dans les modèles statistiques peut être identifié : il contient toutes les formes qui apparaissent dans le corpus, y compris les mots-outils et les mots rares, soit environ 30 000 formes graphiques. Lorsqu'un document du corpus de test contient un mot qui n'apparaît pas dans le corpus d'entraînement, ce mot est simplement ignoré.

Nous formulons l'hypothèse que, dans un fichier de l'ensemble d'entraînement, toutes les phrases prononcées par un président donné font partie du même thème. Par conséquent, le corpus d'entraînement pour apprendre les sous-thèmes de J. Chirac est constitué en supprimant les insertions de F. Mitterrand et en agrégeant les parties de texte séparées par ces insertions. Deux passages qui appartiennent à deux documents différents dans le corpus original ne sont jamais concaténés dans le même texte. De la même manière, chaque fragment attribué à F. Mitterrand constitue un document distinct.

B.4.2 Description de l'algorithme

L'algorithme itératif d'estimation des paramètres décrit en section B.3.2 est utilisé pour obtenir les coefficients β_{wt} correspondant aux thèmes récurrents des discours de J. Chirac. Le nombre de thèmes n_C est fixé *a priori* en effectuant des mesures de F-score (tel que défini par les organisateurs pour l'évaluation de la tâche DEFT) en validation croisée sur

l'ensemble d'apprentissage. Nous déterminons de même un nombre de thèmes n_M pour F. Mitterrand, ce qui donne au total $n_C + n_M$ distributions sur le vocabulaire, qui sont représentatives des différents sujets abordés dans les discours de J. Chirac et F. Mitterrand.

Sur les textes du corpus de test, nous n'avons d'autre choix que d'affecter une variable latente à chaque phrase puisque nous n'avons pas d'information a priori sur les ruptures thématiques. Le problème est alors d'évaluer la séquence thématique la plus probable pour chaque nouveau texte. Pour cela, nous utilisons les modèles de Markov cachés qui ont été présentés à la section B.2, en utilisant le modèle de mélange de la section B.3 pour calculer la vraisemblance des phrases examinées dans chaque thème. Pour chaque document du corpus de test, l'état (c'est-à-dire le thème) le plus probable de chaque phrase est ainsi déterminé par application de l'algorithme de Viterbi.

B.4.3 Évaluation du segmenteur

Nous étudions ici uniquement les résultats sur la tâche 1 de DEFT : dans la mesure où nous n'avons pas cherché à tirer profit des informations spécifiques liées aux noms et aux dates, nos performances sur les autres tâches sont quasiment identiques à celles obtenues sur la tâche 1. Sur la base des résultats du Défi [Alphonse et al., 2005], il semble que, comparativement aux autres méthodes, notre modèle soit plus efficace sur cette tâche que sur les deux autres, dans la mesure où nos scores sont sensiblement les mêmes sur les trois tâches alors que d'autres équipes améliorent leurs performances sur les tâches 2 et 3. Pour faire de même, il nous aurait fallu ajuster les poids des dates et des noms de personnes dans le calcul de la vraisemblance de chaque phrase.

Pour la campagne de test officielle, nous avons soumis sur cette tâche les types 1 et 2 avec $n_C = 10$ et $n_M = 4$, le type 2 obtenant les meilleures performances. La table B.1 montre qu'il est possible d'atteindre des performances légèrement supérieures en utilisant le type 3, toujours pour les mêmes nombres de thèmes.

	F	Préc.	Rappel	Corr.	M → C	C → M
Type 1 (soumission DEFT 1)	80.33	88.01	73.87	94.94	3.65	1.41
Type 2 (soumission DEFT 2)	86.04	86.44	85.65	96.12	2.01	1.88
Type 3	86.96	84.32	89.78	96.24	1.43	2.33
Vainqueurs DEFT	87.0	88.3	85.8	–	–	–

Tab. B.1 – Évaluation des différents modèles
Résultats pour $n_C = 10$ et $n_M = 4$ et comparaison avec la meilleure équipe. En plus des valeurs de F-score (pour $\beta = 1$), pour chaque modèle sont donnés la précision, le rappel (également tels que définis par les organisateurs de DEFT), ainsi que le pourcentage de phrases correctes et les pourcentages d'erreurs par type (phrases de J. Chirac attribuées à F. Mitterrand, et phrases de F. Mitterrand attribuées à J. Chirac).

Notons que le meilleur résultat parmi tous nos essais, autour de $F = 88$ (non présenté dans la table B.1), est obtenu avec le type 3 en fixant $n_C = 10$ et $n_M = 3$.

De façon générale, les résultats de DEFT [Alphonse et al., 2005] valident largement l'approche consistant à combiner méthodes de classification et de segmentation. Plus spécifiquement, on remarque que les équipes utilisant des HMMs [El-Bèze et al., 2005, Labadié et al., 2005, Kerloch and Gallinari, 2005] obtiennent toutes des résultats satisfaisants.

Une analyse plus fine des résultats obtenus avec le type 3 permet de constater que les erreurs ne sont pas également réparties, mais se concentrent sur un nombre relativement

faible de documents. Elles correspondent à des situations dans lesquelles le modèle a choisi de considérer qu'un fragment significatif du texte était une insertion, alors que le texte n'en comprenait pas ; ou bien, au contraire, a failli à détecter la moindre insertion. Près de 90% des erreurs sont ainsi localisées dans les 80 documents de test les moins bien étiquetés, alors qu'à l'inverse plus de la moitié des documents ne contiennent aucune erreur de segmentation.

B.4.4 Analyse des résultats

La figure B.7 permet d'apprécier quantitativement l'apport de chacune des contraintes ainsi que de l'augmentation du nombre de sous-thèmes. Il apparaît ainsi que multiplier les sous-thèmes est d'autant plus efficace que le modèle utilisé est contraint : pour le type 1, les résultats obtenus en prenant un thème par classe sont pratiquement les meilleurs ; alors que pour le type 3, nous observons une augmentation sensible des performances lorsque nous multiplions les sous-divisions thématiques. Dans tous les cas, la solution consistant à ne garder que deux sous-thèmes par classe est clairement sous-optimale. Il semble que les nombres optimaux de sous-thèmes soient de 3 pour la classe M et de 8 ou 10 pour la classe C. Cette différence s'explique probablement par la quantité de données d'apprentissage, bien plus importante pour un auteur que pour l'autre, et qui permet par conséquent d'estimer de façon fiable un plus grand nombre de paramètres.

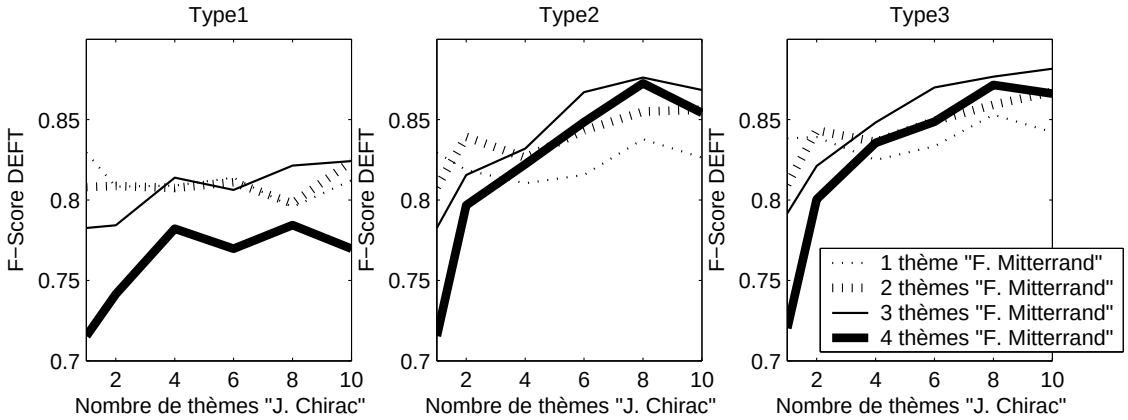

FIG. B.7 – F-Score obtenu sur la tâche 1 de DEFT en fonction de n_C

En dehors de ces évaluations portant sur la tâche DEFT, une question légitime, et particulièrement pertinente en ce qui concerne l'utilité pratique de ce type de modèles, concerne la nature des thèmes déterminés de façon non supervisée lors de l'apprentissage. La procédure d'interprétation des thèmes que nous avons adoptée repose sur une heuristique en deux étapes :

- partant de la représentation complète des documents en sacs de mots, nous utilisons l'approche de sélection de variables décrite dans [Boullé, to appear, 2006], qui permet de ne retenir que les attributs (les mots) qui, individuellement, apportent une information vis-à-vis de la classification des documents sur les 14 thèmes. Cette étape vise à réduire fortement la taille du vocabulaire « discriminant » ;
- à partir de cette représentation réduite et pour chacun des thèmes, nous calculons la matrice de co-occurrences entre les mots des documents du thème, qui est interprétée comme une matrice de similarité relativement au thème. Pour caractériser le thème,

nous retenons finalement les N mots les plus centraux au sens de la centralité spectrale, consistant à considérer la projection du mot sur la première direction propre de la matrice de similarité.

Les tables B.2 et B.3 listent les 15 mots les plus saillants ainsi déterminés pour chaque thème. On peut constater que, dans la plupart des cas, ces mots-clés suffisent à donner une idée assez précise de la thématique sous-jacente. C'est en particulier toujours le cas pour les discours de J. Chirac (thèmes C1 à C10) ; l'interprétation de certains thèmes des discours de F. Mitterrand est moins nette (voir le thème M2 en particulier). Mais, même dans ce dernier cas, le classement suit une bipartition entre thèmes plus nationaux (M1 et M2) et internationaux (M3 et M4).

Thème C1	Thème C2	Thème C3	Thème C4	Thème C5
amitié	école	cultures	mondialisation	états
peuples	familles	diversité	accord	union
paix	famille	côtés	amérique	conseil
continent	concitoyens	partenariat	croissance	européen
peuple	public	coopération	lutte	membres
histoire	égalité	paix	guerre	européenne
europe	citoyens	dialogue	marché	aide
deux	autorité	soutien	puissance	institutions
union	enfants	amis	peuples	internationale
ambition	nation	engagement	union	mise
relation	service	monsieur	indispensable	sommet
construire	publique	cher	règles	sécurité
vision	sociale	sommet	continent	défense
nations	services	projets	européen	européens
liberté	société	votre	emplois	mondiale
Thème C6	Thème C7	Thème C8	Thème C9	Thème C10
afrique	devoir	république	combat	moi
paris	protection	peuple	liberté	maire
moi	citoyens	cher	guerre	succès
partenariat	combat	emplois	général	veut
relation	autorité	compatriotes	fut	façon
choses	société	maire	valeurs	justice
guerre	public	mes	jamais	puis
fut	mission	amitié	homme	puissance
vivre	lutte	choses	paris	domaine
institutions	droit	veux	nom	lors
problèmes	sociale	europe	honneur	paix
membres	école	justice	forces	union
peuples	répondre	union	contre	démocratie
messieurs	famille	nation	justice	europe
mesdames	agit	construction	devoir	indispensable

Tab. B.2 – Mots caractéristiques pour les thèmes de J. Chirac

Cette analyse permet de mettre en évidence que la méthode de classification employée, bien que n'exploitant aucune information de nature sémantique, permet effectivement de construire des groupes de documents présentant une unité thématique.

Thème M1	Thème M2	Thème M3	Thème M4
sociaux	marché	europe	guerre
paix	on	cent	nations
renforcer	effort	communauté	états
coopération	cinq	puissance	peuples
sommet	entendu	marché	comment
économiques	simplement	trois	forces
devons	moins	serait	droit
niveau	plan	construction	équilibre
compatriotes	cent	institutions	peuple
hommage	quand	européen	droits
liens	faut	peu	paix
économie	plusieurs	était	amérique
notamment	mille	moins	sécurité
membres	là	après	cas
afrique	est-à-dire	autres	dès

TAB. B.3 – Mots caractéristiques pour les thèmes de F. Mitterrand

Enfin, pour évaluer l'apport de l'algorithme itératif d'initialisation utilisé pour apprendre les paramètres des thèmes, nous avons comparé ses performances avec celles obtenues en utilisant une procédure d'initialisation plus simple (initialisation « Dirichlet »), qui considère d'emblée tout le vocabulaire. Les mesures utilisées sont la perplexité (capacité de prédiction du modèle probabiliste) et l'information mutuelle (coïncidence des étiquettes C/M proposées avec les "vraies" classes). Comme le montrent les résultats de la table B.4, moyennés sur 50 tirages, les performances nettement meilleures obtenues par la méthode itérative telles que les mesure la perplexité se traduisent également par un gain, d'ampleur plus modeste, sur la tâche d'évaluation extrinsèque de DEFT.

Méthode	Perplexité - corpus C	Perplexité - corpus M	Information Mutuelle
Init. Dirichlet	755.7±3.4	775.5±2.2	0.83±0.01
Init. Itérative	733.3±2.2	760.8±2.2	0.85±0.01

TAB. B.4 – Résultats comparés pour deux méthodes d'inférence des paramètres
Les mesures d'évaluation sont calculées sur un ensemble de test, puis moyennées sur plusieurs essais et en validation croisée.

B.5 Conclusion

Nous avons présenté dans cet article la méthode utilisée pour répondre au problème posé dans le cadre du DÉfi Fouille de Textes 2005. Notre approche s'appuie sur l'utilisation de deux outils de base de la fouille de textes : d'une part les modèles de Markov cachés, d'autre part un modèle de classification non supervisée. En particulier, nous avons montré qu'en identifiant les distributions thématiques qui sous-tendent les discours de J. Chirac et F. Mitterrand dans le corpus d'entraînement, nous sommes mieux à même de segmenter le corpus de test, en déterminant l'enchaînement thématique le plus probable. Il est remarquable que ces deux modules n'aient pas été spécifiquement conçus pour ce travail, mais

que leur assemblage permette d'aboutir à des résultats très satisfaisants.

Dans le cadre des tâches DEFT, de nombreuses améliorations de cette stratégie sont envisageables, consistant, par exemple, à apprendre de manière conjointe plutôt que séparée les différents paramètres (lois d'émission et probabilités de transition) des modèles de Markov. Au-delà de ces aménagements immédiats, peu justifiés au vu du caractère un peu artificiel de la tâche proposée, nous envisageons d'explorer d'autres voies pour améliorer ces modèles.

Concernant les modèles séquentiels, nous prévoyons de tester sur d'autres tâches la combinaison de ce modèle de mélange thématique et d'un modèle de Markov caché modélisant l'enchaînement des variables latentes associées aux phrases ou aux paragraphes, dans un cadre dans lequel l'apprentissage de l'ensemble du modèle s'effectue de manière intégralement non-supervisée. Nous espérons ainsi montrer que les bons résultats obtenus ici, qui s'appuient en partie sur les spécificités structurelles du corpus (informations sur les règles d'insertion), sont généralisables à des applications moins contraintes. Concernant plus spécifiquement le modèle de classification probabiliste non-supervisée, nous envisageons de continuer à travailler sur l'amélioration des méthodes d'inférence, en comparant la méthode itérative heuristique présentée dans cet article avec des algorithmes de simulation (échantillonneur de Gibbs).

Remerciements Nous adressons nos remerciements aux organisateurs de ce Défi Fouille de Textes pour leur travail considérable, ainsi qu'aux rapporteurs dont les commentaires ont grandement contribué à l'amélioration de l'article.

Annexe C

Le programme C++ *Textclust*

C.1 Introduction

Les expériences réalisées dans cette thèse ont été implémentées dans un cadre unique : *Textclust* est un ensemble de classes en C++ permettant d'appliquer des modèles probabilistes d'analyse exploratoire (essentiellement le mélange de lois multinomiales et l'allocation Dirichlet latente).

Les prétraitements (lecture de fichiers, constitution du vocabulaire, lemmatisation, représentation matricielle) sont effectués par des modules de la bibliothèque *Bag-Of-Word* (*BOW*) [McCallum, 1996]. Nous implémentons à partir de ces éléments un cadre d'évaluation, permettant notamment de mesurer la perplexité ou la similarité avec une catégorisation existante et de conduire des expériences en validation croisée.

Les algorithmes d'inférence principalement étudiés sont les suivants :

- pour le mélange de multinomiales, l'algorithme d'espérance-maximisation (et certaines variantes, telles que la méthode d'inférence itérative) et l'échantillonnage de Gibbs ;
- pour l'allocation Dirichlet latente, l'échantillonnage de Gibbs et l'algorithme d'espérance-propagation.

Du point de vue de la structure, nous distinguons donc les classes *BOW*, qui assurent la conversion des textes en objets sur lesquels s'appliquent les algorithmes (matrices de comptes par exemple) et les classes *Textclust*, qui sont chargées des étapes d'inférence, d'évaluation et d'interprétation de la classification. Nous résumons cette répartition sur la figure C.1.

C.2 La bibliothèque *BOW*

La bibliothèque *BOW* [McCallum, 1996], telle que nous l'avons transformée, s'appuie toujours sur des routines non orientées objet : les utilitaires *BOW*. Ils sont dédiés à des opérations bas niveau, telles que le contrôle des informations affichées (verbose), la gestion des messages d'erreur, les entrées/sorties fichiers et la génération de nombres premiers. Nous ne documentons pas ici ces fonctions de base qu'un utilisateur de *Textclust* a peu de raisons de modifier.

La partie de *BOW* codée en C++ concerne les classes suivantes :

- vocabulaire (`bow_int4word`) ;
- lemmatisation (`bow_lexer`) ;

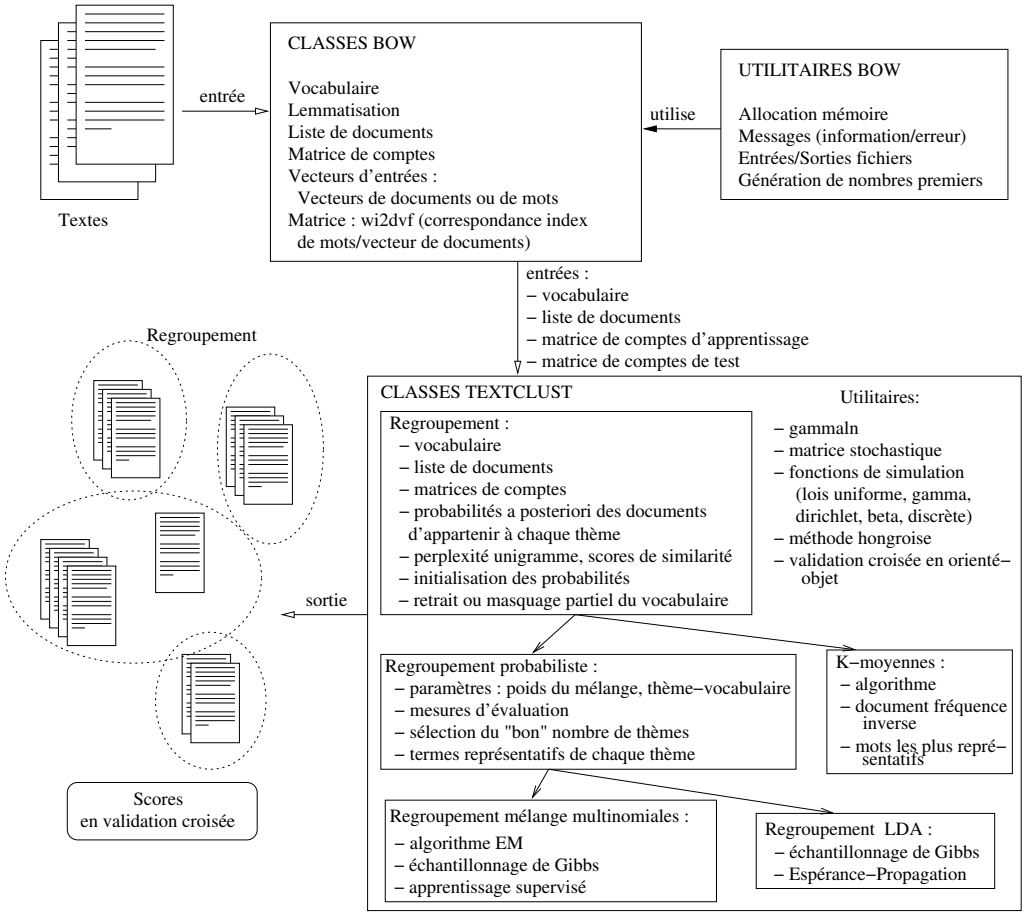

FIG. C.1 – Structure globale de *Textclust*.

– listes de documents (`bow_cdocs`, utilisées lorsque la matrice de comptes doit être traitée document par document) ;
– matrices de comptes (`bow_wi2dvf`, traitement de la matrice de compte mot par mot).

Ces classes sont déclarées au sein d'un fichier d'en-tête unique : `bow.h`. Il faut préciser ici qu'un certain nombre d'options *BOW* ont été simplement copiées/collées mais pas testées (en particulier parmi la multitude d'options de lemmatisation). Ces fonctions sont donc susceptibles de contenir des bugs et les utilisateurs potentiels de ces options sont encouragés à vérifier le comportement du programme (grâce aux méthodes `print`).

C.2.1 Vocabulaire

La classe générale de la bibliothèque *BOW* en charge de la correspondance mots/indices sur le vocabulaire se nomme `bow_int4str` et est construite à l'aide d'une table de hachage. Dans *BOW*, le vocabulaire en est ensuite dérivé sans modification majeure. Nous avons jugé que cela ne correspondait pas totalement à nos besoins, dans la mesure où il existe des cas où nous souhaitons trier les mots dans le vocabulaire selon un ordre particulier (cela n'est pas possible avec `bow_int4str` puisque les indices dépendent directement de la table de hachage). Par conséquent, nous avons conçu une classe interface `bow_int4word` qui donne les signatures des fonctions à implémenter par toute classe de type vocabulaire.

La classe `bow_int4word_with_hash` est la classe *BOW* de vocabulaire originale et éga-

lement la classe par défaut. La seconde classe dérivée `bow_int4word_with_list` est moins efficace en termes de temps de calcul mais plus flexible puisqu'autorisant à associer n'importe quel mot à n'importe quel indice. Dans le futur, il faudrait améliorer la rapidité de l'implémentation de `bow_int4word_with_list` en la dérivant de `bow_int4word_with_hash` par combinaison de la table de hachage et d'une bijection de l'espace des indices dans lui-même.

C.2.2 Lemmatisation

Les options de lemmatisation offertes par la bibliothèque *BOW* sont les suivantes :
- pour l'anglais, deux listes d'anti-mots sont prédéfinies dans la classe `bow_stoplist` et il est possible d'en ajouter aisément ;
- toujours pour l'anglais, l'algorithme de racinisation de Porter est implémenté au sein de la classe `bow_porter_stemmer`, dérivée de la classe interface `bow_stemmer` ;
- les différents paramètres de lemmatisation (ignorer certains mots, chiffres, préfixes, suffixes, etc) sont regroupés dans une classe `bow_lexer_parameters` avec des valeurs par défaut. Des classes dérivées implémentent des lemmatiseurs particuliers : `bow_alpha_lexer_parameters`, `bow_alphanum_lexer_parameters`, `bow_alpha_only_lexer_parameters`. Ces derniers permettent, entre autres, de préciser des délimiteurs pour les textes, les mots et de préciser ce qu'il faut faire des chiffres.

Un lemmatiseur, défini en général dans la classe `bow_lexer` et implémenté par défaut dans la classe `bow_lexer_simple`, inclut les options décrites ci-dessus et une structure appelée `bow_lex` permettant de stocker le document courant. Les fonctions les plus utiles de la classe `bow_lexer_simple` sont en pratique `open_text_fp` et `get_raw_word`. Il faut noter que des lemmatiseurs sont définis par défaut en statique dans la classe `clustering` pour les utilisateurs ne souhaitant pas passer du temps à régler les différents paramètres.

C.2.3 Documents

Dans le modèle du sac-de-mots, un document est vu comme un vecteur de comptes sur l'espace du vocabulaire. La classe `bow_wv` implémente les vecteurs de mots en tableaux de `bow_we` (structure contenant notamment l'indice du mot et le compte). Les autres informations relatives aux documents (catégorie, nom du fichier correspondant, mots à ignorer, étiquette apprentissage ou test) figurent dans la classe `bow_cdoc` dont `bow_wv` est une variable membre. Au final, les éléments `bow_cdoc` sont regroupés au sein d'une classe `bow_cdocs`, qui est un tableau de `bow_cdocs` à réallocation dynamique, permettant de naviguer facilement entre documents d'apprentissage et de test. `bow_cdocs` est donc dans ce qui suit la structure utilisée lorsqu'il faut parcourir le corpus document par document.

C.2.4 Matrices de comptes

Les matrices termes-documents sont utilisées pour parcourir le corpus mot par mot. Elles sont implémentées par la classe `bow_wi2dvf`. Le rôle de cette classe est d'associer chaque indice de mots à un vecteur de documents `bow_dv` : étant donné un indice sur l'ensemble du vocabulaire, un `bow_wi2dvf` renvoie un vecteur `bow_dv` contenant l'ensemble des documents dans lesquels apparaît le mot en question. Plus précisément, `bow_dv` est un tableau d'éléments de type `bow_de`, une structure analogue à `bow_we` mais pour le cas des vecteurs de documents (par conséquent, chaque cellule est liée à un document particulier). `bow_wi2dvf` propose également des fonctions permettant d'écarter ou d'ignorer les mots

de multiples façons. Les matrices de comptes peuvent être chargées et sauvegardées dans des fichiers dans de nombreux formats, et notamment le format `sparse` de Matlab.

C.3 *Textclust*

La classe principale de *Textclust* est `clustering` (regroupement). Elle englobe les données, consultables document par document (via `cdocs`) ou mot par mot (via `wi2dvf`), et propose des fonctions d'accès aux éléments de base, aux algorithmes et aux mesures d'évaluation. `clustering` a recours aux classes *BOW* pour quatre éléments importants : le vocabulaire, la liste des documents et les matrices de comptes d'apprentissage et de test.

Les autres classes dérivent de `clustering` notamment `clustering_with_kmeans` (algorithme des k-moyennes), `probabilistic_clustering` (regroupement probabiliste, avec gestion des paramètres) et les classes dérivées de ce dernier :

- `clustering_with_multinom_model` (modèle de mélange de lois multinomiales)
- `clustering_with_lda` (allocation Dirichlet latente)

Dans la section des utilitaires, nous présentons la classe `stoch_matrix` pour la gestion des matrices dont les lignes somment à 1, ainsi que les fonctions de la bibliothèque `alea` pour la simulation de lois de probabilités et la méthode hongroise déterminant le meilleur appariement entre deux regroupements. La classe `cross_validation` et ses descendantes constituent une interface entre les paramètres de l'algorithme et l'objet `clustering` et prennent en charge les problèmes de division du corpus. Enfin, `textclust.c` est le programme à lancer par l'utilisateur. Il prend un fichier de configuration en entrée, en fait la lecture et exécute la validation croisée adéquate.

C.3.1 Regroupement

`clustering` a pour variables membres les matrices de comptes (`train_wi2dvf`, `test_wi2dvf`), les tableaux de documents (`cdocs`) et le vocabulaire. Les éléments représentant le regroupement proprement dit sont le nombre de thèmes et les probabilités pour chaque document d'appartenir à un thème donné (0 ou 1 pour une classification déterministe).

Dans la mesure où nous avons pris le parti d'ignorer les mots qui ne figurent pas dans l'ensemble d'apprentissage, `train_wi2dvf` constitue l'objet de référence pour la constitution du vocabulaire. Insistons sur le fait que l'objet `vocabulary` permet d'obtenir l'équivalence entre indices sur le vocabulaire et mots mais n'a aucune information sur les mots à ignorer (présents uniquement dans le corpus de test) et donc sur les mots à considérer pour l'application des algorithmes.

Tous les documents, qu'ils appartiennent à l'ensemble d'apprentissage ou à l'ensemble de test, figurent dans le même tableau `cdocs` (de 0 à `bow_cdocs_length`-1), l'étape d'indexage ayant été réalisée par la fonction `bow_cdocs_register_doc`. Par conséquent, certains indices de documents sont absents de la matrice d'apprentissage `train_wi2dvf` ou de la matrice de test `test_wi2dvf`. Pour ne parcourir que les documents d'un type, par exemple uniquement ceux de l'ensemble d'apprentissage, il faut utiliser les fonctions dédiées, dans ce cas `bow_cdocs_index_of_train_doc`. De même, l'objet `proba` contient à la fois les informations de classification relatives aux documents d'apprentissage et aux documents de test.

Nous pouvons implémenter dans la classe `clustering` les fonctions qui ne dépendent pas du modèle de classification sous-jacent. C'est le cas par exemple de la construction

d'un objet `clustering` à partir d'un répertoire de fichiers textes, avec un lemmatiseur donné, ou à partir de fichiers de données, par exemple des fichiers représentant des matrices au format `sparse` de Matlab. Les fonctions `init_postprob` permettent d'initialiser de façon aléatoire les probabilités d'appartenance des documents aux thèmes. La réduction ou la modification du vocabulaire entraîne la modification de toutes les variables membres. Ces changements sont effectués dans la fonction `hide_or_remove_words`. La perplexité dépendant du modèle considéré, elle n'est disponible que dans les classes dérivées. Il est néanmoins possible d'implémenter à ce niveau le calcul d'une perplexité de référence : la perplexité unigramme (`compute_unigram_perplexity`). Les mesures d'évaluation ne dépendant que de la classification sont également calculables dans cette classe, par exemple l'information mutuelle (`compute_mutual_information`) ou les scores de cooccurrences (`compute_cooccurrences`). Enfin, l'objet `clustering` est également constitué de fonctions permettant d'afficher, d'enregistrer ou de charger les différents composants.

C.3.2 Regroupement probabiliste

La classe `probabilistic_clustering` regroupe les caractéristiques communes aux modèles probabilistes implémentés : le modèle de mélange de multinomiales (`clustering_with_multinom_model`) et l'allocation Dirichlet latente (`clustering_with_lda`). Les variables membres importantes sont les paramètres communs, à savoir les poids des thèmes `wghtthm` (α), la probabilité des mots conditionnellement aux thèmes `wghtvoc` (β), qui est en fait un pointeur sur une `stoch_matrix` et les valeurs courantes des log-vraisemblances sur les ensembles d'apprentissage et de test.

Les méthodes implémentées à ce niveau concernent l'évaluation (la perplexité est calculable en fonction des valeurs courantes de la log-vraisemblance), la sélection du nombre de thèmes par critères d'information (Akaike, Schwartz, dépendant du nombre de paramètres et de la log-vraisemblance) et l'identification des mots les plus caractéristiques de chaque thème (dépendant de la valeur de `wghtvoc` et des comptes).

C.3.2.1 Regroupement avec mélange de multinomiales

La classe `clustering_with_multinom_model`, dérivée de `probabilistic_clustering`, implémente des algorithmes liés au modèle de mélange de lois multinomiales.

Pour la classification non supervisée, l'inférence peut être conduite via l'algorithme EM (selon différents modes : simple, avec classification déterministe, itératif ou en contexte semi-supervisé) `run_em_algorithm` ou bien via l'échantillonnage de Gibbs (complet ou rao-blackwellisé) `run_gibbs_sampling_algorithm`. Dans un contexte de classification supervisée, la fonction `compare_with_naive_bayes` a permis d'établir la comparaison présentée en section 4.6.2.

C.3.2.2 Regroupement avec le modèle LDA

La classe `clustering_with_lda` est également dérivée de `probabilistic_clustering` et est liée au modèle allocation Dirichlet latente que nous avons décrit en section 5.2.

L'EM n'étant pas applicable pour l'inférence pour ce modèle, les paramètres sont appris par échantillonnage de Gibbs, via `learn_wghtvoc_with_gibbs_sampling`. D'autre part, le calcul de la log-vraisemblance peut être effectué par des simulations Monte Carlo ou bien par espérance-propagation.

C.3.3 K-moyennes

`clustering_with_kmeans`, dérivée de `clustering`, implémente l'algorithme des K-moyennes. `run_kmeans_algorithm` consiste simplement à alterner la fonction permettant de calculer l'affectation de chaque document à un groupe, `compute_assignments_from_centroids`, et celle qui détermine les centroïdes à partir des appartenances thématiques de chaque document, `compute_centroids_from_assignments`.

Des variables membres supplémentaires sont nécessaires, telles que les coordonnées des centroïdes ou l'idf associée à chaque mot (section 2.2.2), que l'utilisateur peut décider ou non de prendre en compte.

C.3.4 Utilitaires

La classe `stoch_matrix` permet de traiter les matrices dont les lignes somment à 1 et peuvent donc être assimilées à des distributions de probabilités. C'est le cas par exemple des matrices `wghtvoc` et `proba`. `alea` est une bibliothèque de générateurs aléatoires [Cappé et al., 2003], que nous utilisons en particulier pour simuler des lois Dirichlet et discrètes. Le code de la classe `hungarian` est repris de [Stachniss, 2004], il permet de calculer le score de cooccurrences entre deux classifications.

C.3.5 Validation croisée et lecture des fichiers de configuration

Le mécanisme d'appel des algorithmes, avec les paramètres appropriés, en validation croisée, est géré par héritage en partant d'une classe `cross_validation`. L'idée est de transmettre les paramètres de l'algorithme dès la construction de l'objet et de les conserver en variables membres. L'algorithme principal (EM, échantillonnage de Gibbs, etc) est alors appelé avec pour arguments en entrée l'objet `clustering` concerné et le fichier de sortie où écrire les résultats.

Les fichiers de configuration sont lus et traités par `textclust.c`, source de l'exécutable final qui vérifie que tous les paramètres requis pour lancer un algorithme particulier sont présents. Le format des fichiers de configuration est classique : attribut = valeur. Voir la liste des attributs et des dépendances dans le fichier `params.h`.

`cross_validation` et `textclust.c`, dans leur forme actuelle, sont utilisables pour effectuer des tests sur des corpus « fermés » et obtenir des scores d'évaluation pour des configurations particulières. Cette approche est adaptée aux expériences décrites dans cette thèse où nous nous posons globalement la question de l'influence des différents paramètres sur la performance finale, sans avoir le besoin d'enregistrer les résultats de la classification, d'appliquer cette classification à d'autres documents ou de conserver durablement les valeurs des paramètres. Un utilisateur souhaitant se servir de *Textclust* dans une situation « réelle », en construisant la classification non supervisée sur l'ensemble des données disponibles afin de procéder à l'analyse exploratoire d'un corpus ou d'utiliser les résultats pour d'autres applications, ne fera pas appel à la classe `cross_validation` mais procédera d'une des deux façons suivantes :

- soit en adaptant `textclust.c` : plutôt que de créer et appeler un objet `cross_validation`, appeler directement l'algorithme d'inférence souhaité sur l'objet `clustering` créé et sauvegarder les paramètres ou les probabilités a posteriori dans un fichier ;
- soit en créant directement un objet `clustering` depuis un autre fichier, en copiant/collant les passages adéquats de `textclust.c`.

Bibliographie

E. M. Airoldi, W. W. Cohen, and S. E. Fienberg. Bayesian methods for frequent terms in text : Models of contagion and the delta square statistic. In *Proceedings of the Joint Annual Meeting of the Interface and the Classification Society of North America*, 2005.

E. Alphonse, A. Amrani, J. Azé, T. Heitz, A.-D. Mezaour, and M. Roche. Préparation des données et analyse des résultats de DEFT'05. In *"Traitement Automatique des Langues Naturelles" (TALN 2005) - Atelier DEFT'05*, volume 2, pages 99–111, 6-10 juin 2005.

I. Antonellis and E. Gallopoulos. Exploring term-document matrices from matrix models in text mining, 2006. SIAM Text Mining Workshop.

L. Azzopardi, M. Girolami, and K. van Risjbergen. Investigating the relationship between language model perplexity and ir precision-recall measures. In *Proceedings of the 26th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 369–370, 2003.

H. Baayen. *Word Frequency Distributions*. Kluwer, 2001.

F. R. Bach and M. I. Jordan. Learning spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, 2004.

L. D. Baker, T. Hofmann, A. K. McCallum, and Y. Yang. A hierarchical probabilistic model for novelty detection in text. URL `citeseer.ist.psu.edu/article/baker99hierarchical.html`. submitted to NIPS'99, 1999.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. In *Proceedings of the fourth SIAM International Conference on Data Mining*, 2004.

J.-P. Benzécri et al. *Pratique de l'analyse des données, tome 3. Linguistique et lexicologie*. Dunod, 1981.

M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM review*, 37(4) :573–595, 1995.

C. M. Bishop and M. E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3) :281–293, 1998.

G. Bisson. Évaluation et catégorisation. Journée de travail "Applications, Apprentissage et Acquisition des Connaissances à partir de Textes" (A^3CTE), 2001.

D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, pages 537–544, 2004.

D. M. Blei and J. D. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18, pages 147–154, 2006.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 601–608, 2002.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT)*, pages 144–152, 1992.

M. Boullé. MODL : a Bayes optimal discretization method for continuous attribute. *Machine Learning*, to appear, 2006.

M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, pages 707–720, 2002.

W. Buntine. Variational extensions to EM and multinomial PCA. In *Proceedings of the 13th European Conference on Machine Learning (ECML)*, pages 23–34, 2002.

W. Buntine and A. Jakulin. Applying discrete PCA in data analysis. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 59–66, 2004.

W. Buntine and A. Jakulin. Discrete component analysis. In *Proceedings of the Subspace, Latent Structure and Feature Selection : Statistical and Optimization Perspectives Workshop (SLSFS)*, pages 1–33, 2006.

W. Buntine, J. Löfström, S. Perttu, and K. Valtonen. Topic-specific link analysis using independent components for information retrieval. In *Proceedings of American Association for Artificial Intelligence 2005 Workshop : Link Analysis*, pages 47–52, 2005.

J. F. Canny. GaP : A factor model for discrete data. In *Proceedings of the 27th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 122–129, 2004.

O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag New York, Inc., 2005.

O. Cappé, C. P. Robert, and T. Rydén. CT/RJ-Mix : Transdimensional MCMC for Gaussian mixtures (available as C source code), 2003. URL `http://www.tsi.enst.fr/~cappe/ctrj_mix/`.

C. Chatfield and A. J. Collins. *Introduction to multivariate analysis*. Chapman & Hall, 1980.

P. Cheeseman and J. Stutz. Bayesian classification (autoclass) : Theory and results. In *Advances in Knowledge Discovery and Data Mining*, pages 153–180. MIT Press, 1996.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 310–318, 1996.

F. Y. Y. Choi. Advances in domain independant linear text segmentation. In *Proceedings of the Conference of North American Chapter of the ACL*, Seattle, WA, 2000.

K. W. Church and W. A. Gale. Poisson mixtures. *Journal of Natural Language Engineering*, 1(2) :163–190, 1995.

CLEF. Cross Language Evaluation Forum, 2000–2006. URL `http://www.clef-campaign.org/`.

F. Clérot, O. Collin, O. Cappé, and E. Moulines. Le modèle "monomaniaque" : un modèle statistique simple pour l'analyse exploratoire d'un corpus de textes. In *Colloque International sur la Fouille de Texte (CIFT'04)*, La Rochelle, 2004.

W. G. Cochran. The χ^2 test of goodness of fit. *Annals of Mathematical Statistics*, 23 : 315–345, 1952.

D. J. Cohen. From babel to knowledge : Data mining large digital collections. *D-Lib Magazine*, 12(3), 2006.

M. Cori and J. Léon. La constitution du TAL, étude historique des dénominations et des concepts. *Traitement automatique des langues*, 43(3) :21–55, 2002.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1990.

G. Crane. What do you do with a million books ? *D-Lib Magazine*, 12(3), 2006.

N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*. Morgan Kaufman, 2001.

D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather : A cluster-based approach to browsing large document collections. In *Proceedings of the 15th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 318–329, 1992.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6) :391–407, 1990.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39 :1–38, 1977.

C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, pages 606–610, 2005.

C. Ding, T. Li, and W. Peng. Nonnegative Matrix Factorization and Probabilistic Latent Semantic Indexing : Equivalence, chi-square statistic, and a hybrid method. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*, 2006.

D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts ? In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, 2004.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

S. T. Dumais. Enhancing performance in latent semantic indexing (LSI) retrieval. Technical Report TM-ARH-017527, Bellcore, 1990.

S. T. Dumais. Latent semantic indexing (LSI) and TREC-2. In *The Second Text RE-trieval Conference (TREC2)*, pages 105–116, 1994.

M. El-Bèze, J.-M. Torres-Moreno, and F. Béchet. Peut-on rendre automatiquement à César ce qui lui appartient ? Application au jeu du Chirand-Mitterrac. In *Dans les actes de la conférence "Traitement Automatique des Langues Naturelles" (TALN 2005) - Atelier DEFT'05*, volume 2, pages 125–134, 6-10 juin 2005.

A. Frank. On Kuhn's Hungarian method - a tribute from Hungary. Technical Report TR-2004-14, Egerváry Research Group, Budapest, 2004. URL `http://www.cs.elte.hu/egres/www/tr-04-14.html`.

W. Gale. Good-turing smoothing without tears. Technical Report 94.5, AT & T Bell Laboratories, 1994. URL `http://cm.bell-labs.com/cm/ms/departments/sia/doc/94.5.ps`.

J. R. Galliers and K. S. Jones. *Evaluating Natural Language Processing Systems.* Lecture Notes in Artificial Intelligence. Springer, 1995.

E. Gaussier and C. Goutte. Relation between PLSA and NMF and implications. In *Proceedings of the 28th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 601–602, 2005.

E. Gaussier, C. Goutte, K. Popat, and F. Chen. A hierarchical model for clustering and categorising documents. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research*, pages 229–247, 2002.

P. V. Gehler, A. D. Holub, and M. Welling. The rate adapting poisson model for information retrieval and object recognition. In *Proceedings of ICML International Conference on Machine Learning*, 2006.

M. Girolami and A. Kabán. On an equivalence between PLSI and LDA. In *Proceedings of the 26th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 433–434, 2003.

C. Goutte. Automatic evaluation of machine translation quality, 2006. URL `http://www.xrce.xerox.com/Publications/Attachments/2006-002/MTeval.pdf`.

C. Goutte and E. Gaussier. A probabilistic interpretation of precision, recall and *f*-score, with implication for evaluation. In *Proceedings of the European Colloquium on IR Research (ECIR)*, pages 345–359, 2005.

T. L. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, 2002.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101 (supl 1) :5228–5235, 2004.

M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Intelligent Information Systems Journal*, 12(2–3) :107–145, 2001.

E.-H. S. Han, G. Karypis, V. Kumar, and B. Mobasher. Hypergraph based clustering in high-dimensional data sets : A summary of results. *IEEE Bulletin of the Technical Committee on Data Engineering*, 21(1), 1998.

M. Hearst. TextTiling : Segmenting texts into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1) :33–64, 1997.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1) :177–196, 2001.

T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. Technical report, Massachussets Institute of Technology, Artificial Intelligence Laboratory, 1998. URL `ftp://publications.ai.mit.edu/ai-publications/1500-1999/AIM-1625.ps`.

P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5 :1457–1469, 2004.

D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 329–338, 1993.

A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Wiley-Interscience, 2001.

A. K. Jain, M. N. Murphy, and P. Flynn. Data clustering : a review. *ACM Computing surveys*, 31(3) :264–323, 1999.

J.-N. Jeanneney. Le débat autour des projets de numérisation : revue de presse, 2005. URL `http://www.bnf.fr/pages/dernmin/com_google.htm`.

F. Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, 1997.

T. Joachims. Text categorization with suport vector machines : Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, pages 137–142, 1998.

D. Jurafsky and J. H. Martin. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall PTR, 2000.

R. Kannan, S. Vempala, and A. Vetta. On clusterings : Good, bad and spectral. In *Proceedings of the 41st Annual Symposium on the Foundation of Computer Science (FOCS)*, pages 367–377, 2000.

S. M. Katz. Distribution of content words and phrases in text and language modelling. *Journal of Natural Language Engineering*, 2(1) :15–59, 1996.

F. Kerloch and P. Gallinari. Extraction d'information à partir de modèles de Markov cachés. In *Dans les actes de la conférence "Traitement Automatique des Langues Naturelles" (TALN 2005) - Atelier DEFT'05*, volume 2, pages 145–153, 6-10 juin 2005.

T. Kohonen, editor. *Self-organizing maps.* Springer-Verlag, 1997.

T. Kolenda and L. K. Hansen. Independent components in text, 1999.

J. B. Kruskal. Nonmetric multidimensional scaling : a numerical method. *Psychometrika*, 29(2) :115–129, 1964.

H. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2 :83–97, 1955.

A. Labadié, Y. Romero, and L. Sitbon. Segmentation et classification : deux politiques complémentaires. In *Dans les actes de la conférence "Traitement Automatique des Langues Naturelles" (TALN 2005) - Atelier DEFT'05*, volume 2, pages 183–192, 6-10 juin 2005.

K. Lagus, T. Honkela, S. Kaski, and T. Kohonen. WEBSOM for textual data mining. *Artificial Intelligence Review*, 13(5/6) :345–364, 1999.

T. K. Landauer and S. T. Dumais. A solution to Plato's problem : The latent semantic analysis : Theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104 :211–240, 1997.

T. K. Landauer, P. W. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, 25 :259–284, 1998.

K. Lang. NewsWeeder : Learning to filter netnews. In *Proceedings of the 12th ICML International Conference on Machine Learning*, pages 331–339, 1995.

T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6) :1299–1323, 2004.

L. Lebart and A. Salem. *Analyse Statistique des Données Textuelles*. Dunod, 1988.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 556–562, 2001.

D. D. Lewis. Reuters-21578, distribution 1.0, 1997. URL `http://www.daviddlewis.com/resources/testcollections/reuters21578`.

D. D. Lewis. Naive (Bayes) at forty : The independence assumption in Information Retrieval. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, pages 4–15, 1998.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 281–296, 1967.

L. Maisonnasse and C. Tambellini. Dépendances syntaxiques et méthodes de détection de passages pour une segmentation sur le locuteur et le thème. In *Dans les actes de la conférence "Traitement Automatique des Langues Naturelles" (TALN 2005) - Atelier DEFT'05*, volume 2, pages 155–164, 6-10 juin 2005.

I. Mani. *Advances in Automatic Text Summarization*. MIT Press, 1999.

C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2007. To appear. `www.informationretrieval.org`.

J. Mason. SpamAssassin corpus, 2002. URL `http://spamassassin.apache.org/publiccorpus/`.

U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE transactions on pattern analysis and machine intelligence*, 24 (12) :1650–1654, 2002.

A. McCallum and K. Nigam. A comparison of event models for Naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.

A. K. McCallum. Bow : A toolkit for statistical language modeling, text retrieval, classification and clustering, 1996. URL `http://www.cs.cmu.edu/~mccallum/bow`.

Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text : An exploration of temporal text mining. In *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 198–207, 2005.

M. Meila. Comparing clusterings by the variation of information. In *Proceedings of the 16th Conference on Learning Theory (COLT)*, pages 173–187, 2003.

T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the conference on Uncertainty in Artificial Intelligence (UAI)*, pages 352–359, 2002.

T. M. Mitchell. *Machine Learning*. McGraw-Hill Higher Education, 1997.

J. E. Mosimann. On the compound multinommial distribution, the multivariate β-distribution, and correlations among proportions. *Biometrika*, 49(1–2) :65–82, 1962.

R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. Technical report, University of Toronto, Department of Computer Science, 1993.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering : Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 849–856, 2002.

K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3) :103–134, 2000.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Dover Publications, 1998.

C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing : A probabilistic analysis. In *Proceedings of the ACM Conference on Principles of Database Systems (PODS)*, pages 159–168, Seattle, 1998.

J. Peltonen, J. Sinkkonen, and S. Kaski. Discriminative clustering of text documents. In *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02)*, pages 1956–1960, 2002.

A. Popescu-Belis. Évaluation numérique de la résolution de la référence : Critiques et propositions. *T.A.L. : Traitement automatique de la langue*, 40(2) :117–146, 2000.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1992.

J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2) :945–959, 2000.

J. Quesada, W. Kintsch, and E. Gomez. A computational theory of complex problem solving using latent semantic analysis. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society (COGSCI'02)*, pages 750–755, 2002.

J.-M. Renders. Application des méthodes à noyaux à la fouille de données textuelles. In *International Conference on Text Mining (CIFT'04)*, 2004. URL `http://spip.univ-poitiers.fr/cift/articles_publies/RendersJean-Michel_CIFT_4.ppt`.

Reuters. Reuters corpus volume 1 (RCV 1), 2000. URL `http://about.reuters.com/researchandstandards/corpus/`.

L. Rigouste, O. Cappé, and F. Yvon. Evaluation of a probabilistic method for unsupervised text clustering. In *Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis (ASMDA)*, Brest, France, 2005a.

L. Rigouste, O. Cappé, and F. Yvon. Inference for probabilistic unsupervised text clustering. In *Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP'05)*, Bordeaux, France, 2005b.

L. Rigouste, O. Cappé, and F. Yvon. Modèles multi-thématiques markoviens pour la segmentation de textes. *RNTI : revue des nouvelles technologies de l'information*, 2006a.

L. Rigouste, O. Cappé, and F. Yvon. Quelques observations sur le modèle LDA. In J.-M. Viprey, editor, *Actes des IXe JADT*, pages 819–830, Besançon, 2006b.

C. P. Robert. *Le choix bayésien, Principes et pratique*. Springer, 2006.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 1999.

N. Rooney, D. Patterson, M. Galushka, and V. Dobrynin. A scaleable document clustering approach for large document corpora. *Information Processing and Management*, 42(5) : 1163–1175, 2006.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11) :613–620, 1975.

L. Saul and S. Roweis. An introduction to locally linear embedding, 2001. URL `http://www.cs.toronto.edu/~roweis/lle/papers/lleintro.pdf`.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1) :1–47, 2002.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 731–737, 1997.

N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 129–136, 2002.

N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Research and Development in Information Retrieval*, pages 208–215, 2000.

N. Slonim and N. Tishby. The power of word clusters for text classification. In *Proceedings of the 23rd European Colloquium on Information Retrieval Research (ECIR)*, 2001.

N. Slonim and Y. Weiss. Maximum likelihood and the information bottleneck. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 335–342, 2003.

C. Stachniss. Libhungarian, hungarian method, 2004.

M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Proceedings of the Knowledge Discovery and Data Mining Workshop on Text Mining*, 2000.

F. J. Theis, K. Stadlthanner, and T. Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *Proceedings of European Signal Processing Conference (EUSIPCO)*, 2005.

N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

TREC. Text REtrieval Conference, 1992–2006. URL `http://trec.nist.gov/`.

A. Vinokourov. Why nonnegative matrix factorization works well for text information retrieval. URL `http://citeseer.ist.psu.edu/458322.html`. Preprint, 2002.

A. Vinokourov and M. Girolami. A probabilistic framework for the hierarchic organisation and classific ation of document collections. *Journal of Intelligent Information Systems*, 18(2/3) :153–172, 2002.

R. Vinot. *Classification automatique de textes dans des catégories non thématiques.* Thèse de doctorat en informatique, École Nationale Supérieure des Télécommunications, 2004.

R. Vinot and F. Yvon. Improving Rocchio with weakly supervised clustering. In *Proceedings of the 14th uropean Conference on Machine Learning (ECML)*, pages 456–467, 2003.

E. M. Voorhees. The cluster hypothesis revisited. In *Proceedings of the 8th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 188–196, 1985.

H. M. Wallach. Topic modeling : Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 977–984, 2006.

Y. Weiss. Segmentation using eigenvectors : A unifying view. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 975–982, 1999.

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 267–273, 2003.

Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd ACM international Conference on Research and Development of Information Retrieval (SIGIR)*, pages 42–49, 1999.

C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 743–748, 2004.

D. Zhou, J. Li, and H. Zha. A new mallows distance based metric for comparing clusterings. In *Proceedings of the 22nd international conference on Machine learning (ICML)*, pages 1028–1035, 2005.

G. K. Zipf. *Human Behavior and the Principle of Least-Effort.* Addison-Wesley, 1949.

Index