ANNEXE A Résultats de la détection des complexes QRS

L'algorithme de détection des complexes QRS a été testé sur l'ensemble de la base MIT. Cette base est constituée de 48 enregistrements de 30 mm chacun. Un avantage de cette base est qu'elle couvre un grand nombre de pathologies, ce qui permet de valider la détection des ondes R pour un grand nombre de cas.

Pour juger de la qualité d'un algorithme de ce type, trois grandeurs sont habituellement mesurées :

- NTA (Nombre Total Analysé), qui est le nombre de battements analysés par l'algorithme,
- FP, qui est le nombre de « faux positifs » : c'est le nombre d'ondes R qui ont été détectées par l'algorithme alors qu'elles ne font pas partie de cette catégorie : ces erreurs peuvent correspondre à des emplacements repéré par l'algorithme alors qu'il n'y avait aucune onde caractéristique, ou encore à une onde rerpérée comme R alors qu'il s'agit d'une autre onde caractéristique.
- FN : faux négatifs : c'est le nombre d'ondes étiquetées R que l'algorithme n'a pas détectées.

Le tableau 1 présente les valeurs de ces trois grandeurs obtenues lors des analyses des 48 enregistrements.

Chaque analyse a été effectuée en deux étapes ; on présente ici les résultats en fin de chacune des étapes.

La première étape correspond à l'analyse de la totalité des 30 minutes de chaque enregistrement : les annotations issues de l'algorithme sont comparées à celles des fichiers de référence, et les toutes les erreurs sont comptabilisées ; cette étape est dite « sans analyse du bruit ».

Par opposition, la seconde étape est l'analyse « avec analyse du bruit » : dans ce cas, on ne compte pas, dans les erreurs, les annotations issues de zones trop bruitées. Ces zones sont identifiées de manière automatique pendant l'analyse. C'est cette dernière analyse qui est utilisée dans la suite, car il est primordial de repérer les zones trop bruitées pour ne pas les analyser.

		Sans Analyse du bruit			Avec analyse du bruit			
Enregistrement	**NT**	**NTA**	**FP**	**FN**	**NTA**	**FP**	**FN**	**%Réussite**
MIT_100	2272	2265	0	2	2265	0	0	100
MIT_101	1865	1857	3	2	1847	0	0	100
MIT_102	2187	2180	0	2	2180	0	0	100
MIT_103	2084	2077	0	2	2077	0	0	100
MIT_104	2229	2222	4	2	2221	4	0	99,82
MIT_105	2572	2561	29	5	2475	16	3	99,23
MIT_106	2027	2022	7	0	2022	7	0	99,65
MIT_107	2137	2128	0	3	2128	0	1	99,95
MIT_108	1763	1755	6	2	1714	6	0	99,65
MIT_109	2532	2525	0	2	2525	0	0	100
MIT_111	2124	2116	0	2	2114	0	0	100
MIT_112	2539	2532	0	2	2532	0	0	100
MIT_113	1794	1785	0	2	1784	0	0	100
MIT_114	1879	1871	0	2	1871	0	0	100
MIT_115	1953	1946	1	2	1940	1	0	99,95
MIT_116	2412	2406	0	1	2404	0	0	100
MIT_117	1535	1528	0	2	1524	0	0	100
MIT_118	2278	2270	0	2	2270	0	0	100
MIT_119	1987	1981	0	1	1981	0	0	100
MIT_121	1863	1856	0	2	1849	0	0	100
MIT_122	2476	2469	0	2	2467	0	0	100
MIT_123	1518	1511	0	2	1511	0	0	100
MIT_124	1619	1612	0	2	1612	0	0	100
MIT_200	2601	2588	8	5	2574	8	3	99,57
MIT_201	1963	1935	0	23	1935	0	21	98,91
MIT_202	2136	2129	1	2	2129	1	1	99,91
MIT_203	2980	2964	77	10	2922	76	8	97,13
MIT_205	2656	2649	0	2	2634	0	0	100
MIT_207	1860	1851	403	2	1818	0	0	100
MIT_208	2955	2925	8	15	2910	7	13	99,31
MIT_209	3004	2993	0	2	2991	0	2	99,93
MIT_210	2650	2636	3	7	2628	1	5	99,77
MIT_212	2748	2741	0	2	2696	0	0	100
MIT_213	3250	3243	0	2	3241	0	0	100
MIT_214	2261	2254	1	2	2250	1	0	99,96
MIT_215	3363	3356	0	2	3355	0	0	100
MIT_217	2208	2200	1	2	2199	1	0	99,95
MIT_219	2154	2147	0	1	2133	0	0	100
MIT_220	2048	2041	0	2	2041	0	0	100
MIT_221	2427	2411	11	3	2410	11	1	99,5
MIT_222	2483	2475	8	3	2467	8	1	99,64
MIT_223	2605	2598	0	2	2592	0	0	100
MIT_228	2053	2045	0	3	2045	0	1	99,95
MIT_230	2256	2249	0	2	2249	0	0	100
MIT_231	1571	1561	0	5	1561	0	3	99,81
MIT_232	1743	1735	7	2	1735	7	0	99,6
MIT_233	3079	3071	0	2	3071	0	0	100
MIT_234	2753	2745	0	2	2739	0	0	100
TOTAL	**109452**	**109017**	**578**	**151**	**108638**	**155**	**63**	
Erreur (%)				**0.67 %**			**0.2 %**	

Tableau 1 : Résultats de l'analyse sur la base MIT

I Avec analyse du bruit / Sans analyse du bruit

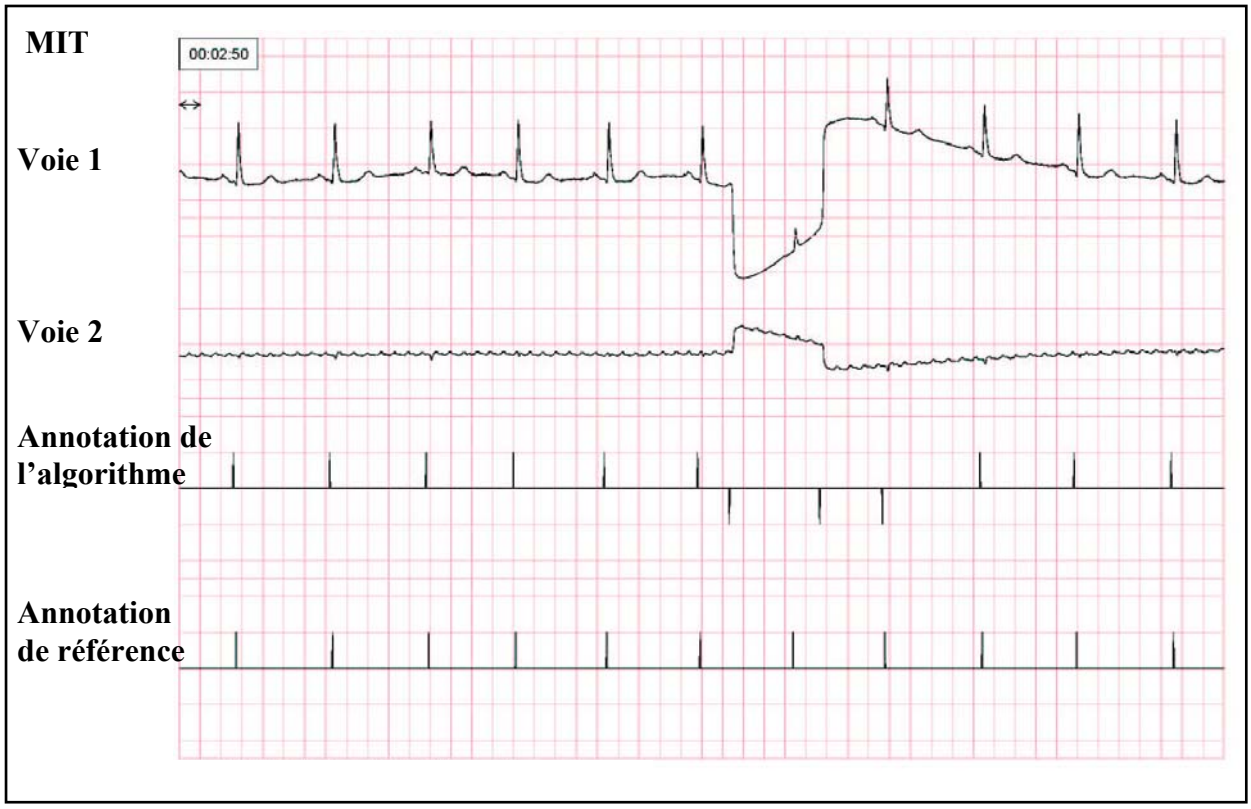


Figure 1 : Pendant cet enregistrement, il y a eu un bruit simultané sur les deux pistes. L'algorithme a alors repéré un complexe en trop. Cependant, celui-ci est rejeté (marque orientée vers le bas). On remarque que les deux complexes QRS suivants, bien que correctement repérés, sont également rejetés : les deux pistes sont en effet encore trop bruitées.

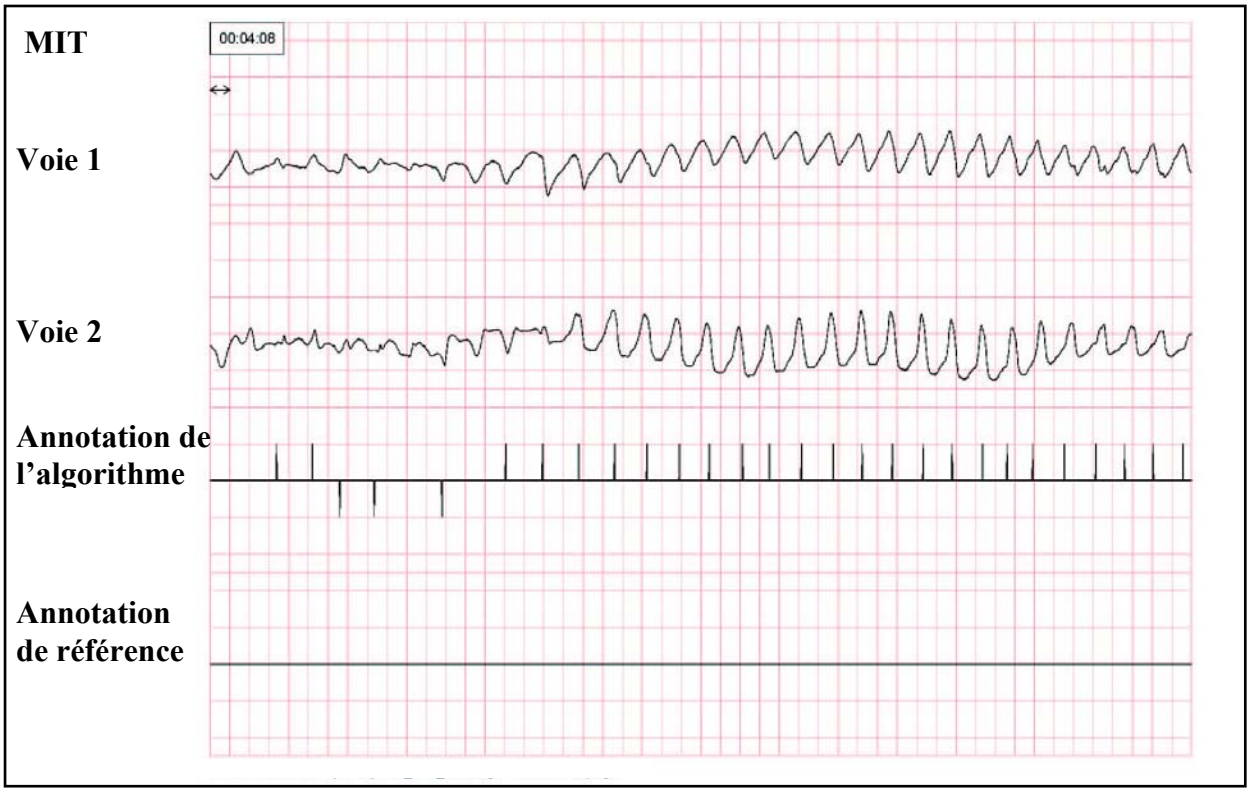


Figure 2 : Lors de l'enregistrement MIT_207, le patient présente 5 périodes de tachycardie ventriculaire. Pendant ces périodes, les annotations sont absentes de la base car les ondes ne peuvent être identifiées comme des QRS. En revanche, notre algorithme repère cette période, ce qui explique le nombre très élevé de faux positifs pour cet enregistrement.

II Faux positifs (FP)

Figure 3 : L'enregistrement est ici très bruité en moyenne fréquence : les pics de bruit ressemblent beaucoup aux QRS voisins. L'algorithme n'a donc pas sélectionné cette zone comme zone trop bruitée, et a introduit des faux positifs (FP).

III Faux négatifs (FN)

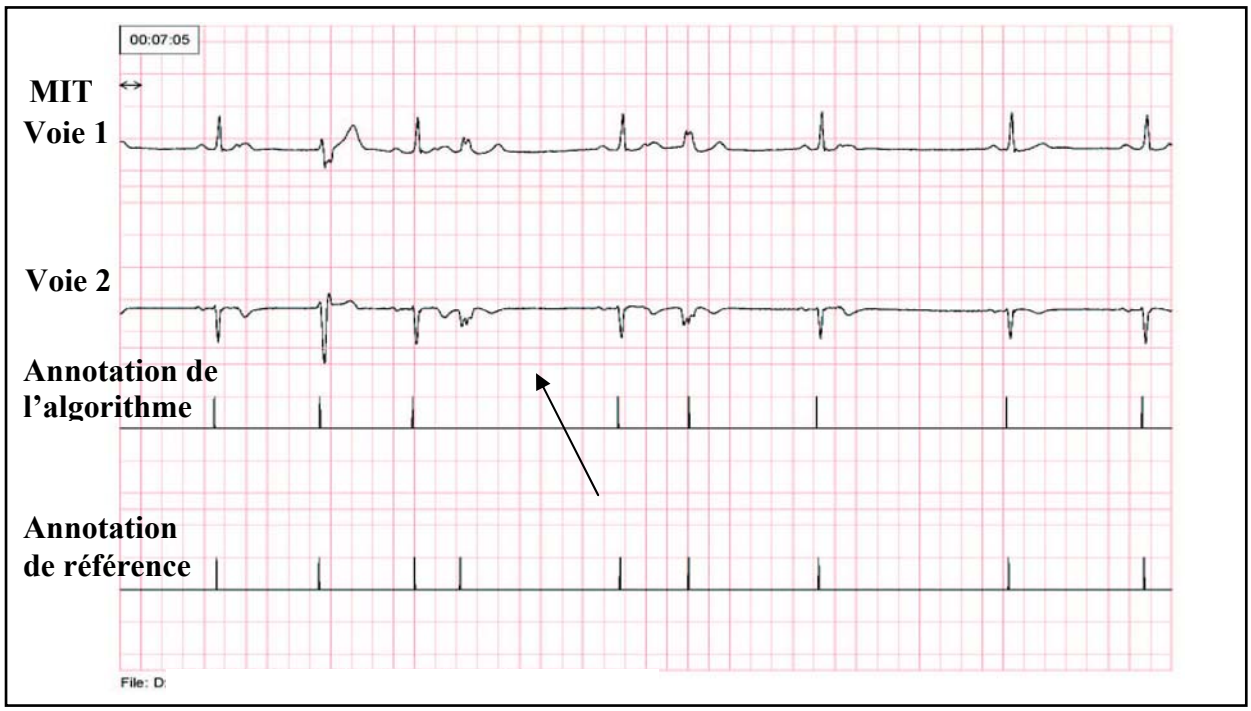


Figure 4 : L'extrasystole n'a pas été détectée. Sa forme est en effet trop étalée, et sa projection trop petite pour qu'elle soit repérée par l'algorithme. C'est un faux négatif (FN).

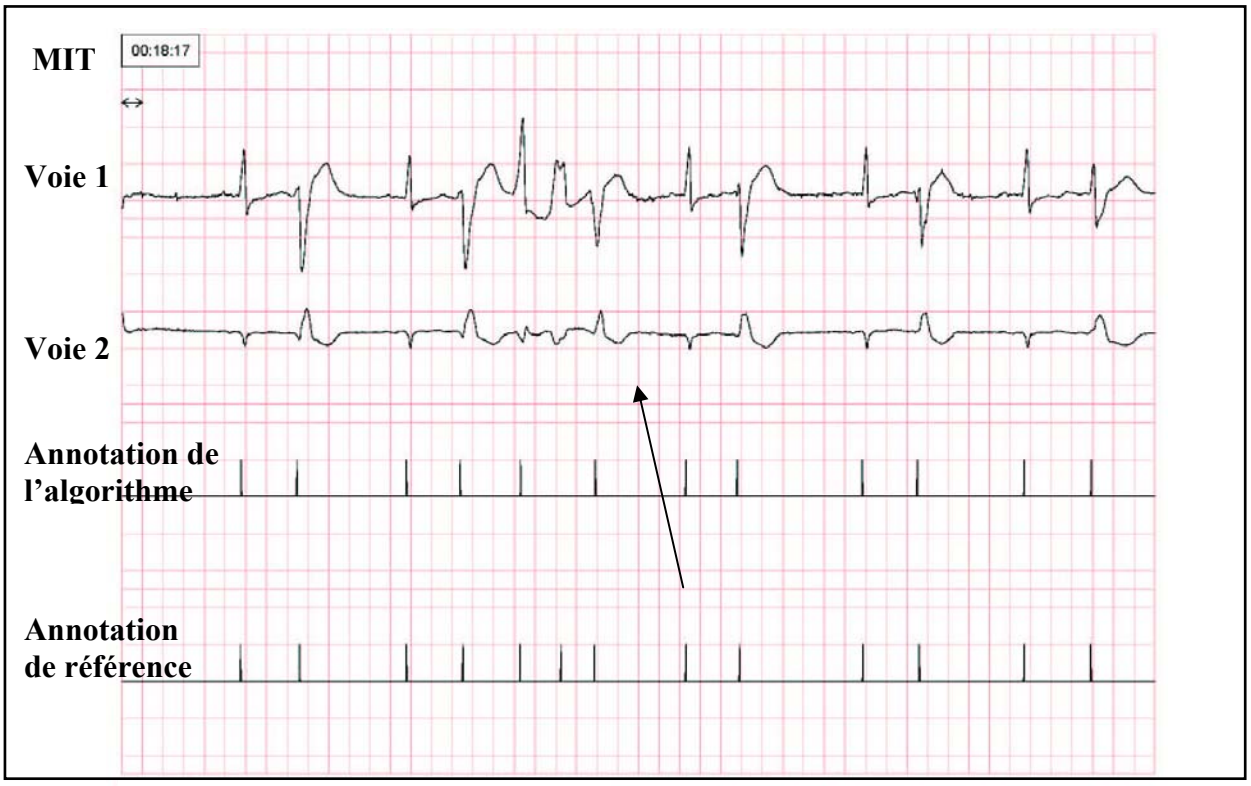


Figure 5 : Ici encore la deuxième ESV n'a pas été détectée. C'est une erreur difficile à corriger. En effet, la première ESV, de grande amplitude, « masque » cette dernière.

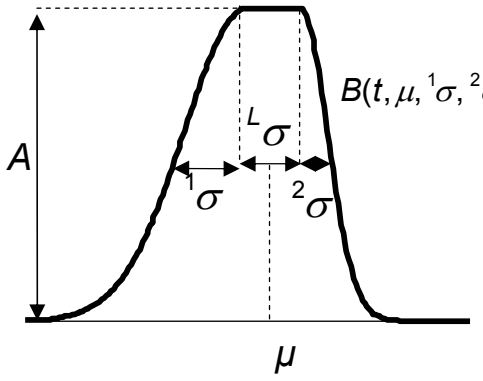
.

ANNEXE B Algorithmes d'optimisation sous contraintes

Adapter les paramètres des fonctions bosses B au signal représentant le battement cardiaque E revient à minimiser la fonction de coût suivante :

$$J(\mu, {}^1\sigma, {}^2\sigma, {}^L\sigma, A) = \frac{1}{N_p}\sum_{k=1}^{N_p}\left({}^{app}E^1(k) - B(k, \mu, {}^1\sigma, {}^2\sigma, {}^L\sigma, A)\right)^2$$

où $\mu, {}^1\sigma, {}^2\sigma, {}^L\sigma$ et A sont les paramètres de la bosse B, et ${}^{app}E^1$ est le signal représentant le battement dans le voisinage de la bosse comme défini au chapitre 7.



$$B(t, \mu, {}^1\sigma, {}^2\sigma, {}^L\sigma, A) = \begin{cases} A\exp\left(-\frac{1}{2}\left(\frac{t-(\mu-{}^L\sigma/2)}{{}^1\sigma}\right)^2\right) & \text{si } t \le \mu - {}^L\sigma/2 \\ A & \text{si } \mu - {}^L\sigma/2 < t < \mu + {}^L\sigma/2 \\ A\exp\left(-\frac{1}{2}\left(\frac{t-(\mu+{}^L\sigma/2)}{{}^2\sigma}\right)^2\right) & \text{si } t \ge \mu + {}^L\sigma/2 \end{cases}$$

La fonction J étant non linéaire en les paramètres, cette optimisation est effectuée par des algorithmes d'optimisation non linéaire multidimensionnelle. Deux algorithmes sont successivement appliqués pour chaque adaptation: le premier est un algorithme du 1er ordre, au cours duquel la direction d'optimisation en un point est colinéaire au gradient de J en ce point. Le second est d'ordre 2 ; la direction de descente est colinéaire au *produit du gradient par l'inverse du Hessien* [Minoux, 1983].

Par construction, les paramètres de la bosse sont soumis aux contraintes suivantes :

$${}^1\sigma > 0,\ {}^2\sigma > 0 \text{ et } {}^L\sigma \ge 0$$

Les apprentissages sont donc réalisés *sous contraintes*.

Le premier algorithme est dit du *gradient projeté* [Minoux, 1983] ; nous lui avons apporté une modification pour accélérer sa convergence. Comme nous le montrons dans la suite, cette amélioration a été rendue possible grâce au caractère particulier des contraintes.

Le deuxième algorithme est celui de Broyden, Fletcher, Goldfarb et Shanno (BFGS) [Broyden, 1970] [Minoux, 1983] que nous avons adapté à l'apprentissage sous contraintes.

I Algorithme du gradient projeté

L'algorithme est bien présenté dans [Minoux, 1983] ; les notations que nous utilisons ici sont les mêmes que celles de cette référence.

Le vecteur $\boldsymbol{x}$ à optimiser est un vecteur de dimension 5, $\boldsymbol{x} = \left[{}^{1}\sigma, {}^{2}\sigma, {}^{L}\sigma, \mu, A \right]^{T}$. Le problème s'écrit donc sous la forme :

$$\begin{cases} \text{Minimiser } J(\boldsymbol{x}) \\ \text{sous les contraintes : } \boldsymbol{A} \cdot \boldsymbol{x} \le \boldsymbol{B} \end{cases}$$

où la matrice $\boldsymbol{A}$ et le vecteur $\boldsymbol{B}$ expriment les contraintes. Ils s'écrivent dans notre cas de la manière suivante : $\boldsymbol{A} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$ et $\boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

On appelle $I^{0}(\boldsymbol{x})$ l'ensemble des indices des lignes de $\boldsymbol{A}$ tels que *la contrainte est saturée* au point $\boldsymbol{x}$, ce qui se traduit en notant $\boldsymbol{a}_i$ la i$^{\text{ème}}$ ligne de $\boldsymbol{A}$ par :

$I^{0}(\boldsymbol{x}) = \{ i \ / \ \boldsymbol{a}_i \cdot \boldsymbol{x} = \boldsymbol{b}_i \ \}$.

Soit $\boldsymbol{A^{0}}$ la sous-matrice issue de $\boldsymbol{A}$ construite avec les lignes d'indices I^{0}.

L'algorithme classique du gradient projeté consiste à calculer la direction $\boldsymbol{d}$ d'optimisation en projetant le gradient de $J(\boldsymbol{x})$, $\nabla J(\boldsymbol{x})$, sur l'intersection des hyperplans définis par chacune des contraintes saturées en $\boldsymbol{x}$, ce qui s'écrit de la manière suivante :

$$\boldsymbol{d} = -(\boldsymbol{Id} - \boldsymbol{A^{0T}}[\boldsymbol{A^{0}}\boldsymbol{A^{0T}}]^{-1}\boldsymbol{A^{0}})\nabla J(\boldsymbol{x})$$

où $\boldsymbol{Id}$ est la matrice identité.

On remarque tout d'abord que, avec les paramètres choisis ici, la matrice $[\boldsymbol{A^0 A^{0T}}]^{-1}$ est toujours *égale à l'identité*, ce qui évite de calculer une inversion de matrice.
On remarque également que les contraintes étant portées par des axes de la base des 5 paramètres, le calcul de $(\boldsymbol{Id} - \boldsymbol{A^{0T} A^0})\nabla J(\boldsymbol{x})$ revient simplement à annuler les composantes d'indice I^0 du vecteur $\nabla J(\boldsymbol{x})$. Autrement dit, au point $\boldsymbol{x}$, la recherche des contraintes saturées (i.e. de I^0) nous permet directement d'obtenir la direction de descente par annulation des composantes de $\nabla J(\boldsymbol{x})$ d'indice I^0.

Notons cependant un inconvénient de la méthode classique du gradient projeté : lorsque que le gradient $\nabla J(\boldsymbol{x})$ nous conduit hors de l'une des contraintes saturées, il est *quand même projeté sur l'hyperplan défini par cette contrainte saturée*[I].
Pour résoudre ce problème de descente systématique le long des contraintes saturées, l'ensemble I^0 utilisé ici est l'ensemble constitué des indices des contraintes saturées au point $\boldsymbol{x}$ dont *le gradient conduit dans le domaine non acceptable* :

$$I^0(\boldsymbol{x}) = \left\{ i \ / \ \boldsymbol{a_i} \cdot \boldsymbol{x} = \boldsymbol{b_i} \ \text{ et } \ \boldsymbol{a_i} \cdot \nabla J(\boldsymbol{x}) > 0 \right\}.$$

Ainsi, seules les composantes du gradient nous amenant dans la mauvaise direction sont projetées sur les contraintes.

Une fois la direction $\boldsymbol{d}$ trouvée, on recherche le pas maximal autorisé $\mu_{\max}$: pour chaque contrainte i, on calcule le pas maximal $\mu^i_{\max}$ qui nous amène sur la contrainte i :

$$\boldsymbol{x'} = \boldsymbol{x} + \mu^i_{\max} \boldsymbol{d},$$

or si $\boldsymbol{x'}$ est sur la contrainte i, on a :

$$0 = \boldsymbol{a_i} \cdot \boldsymbol{x'} = \boldsymbol{a_i} \cdot \boldsymbol{x} + \mu^i_{\max} \boldsymbol{a_i} \cdot \boldsymbol{d}$$

ce qui donne accès à $\mu^i_{\max}$ si $\boldsymbol{a_i} \cdot \boldsymbol{d}$ est non nul. Si ce produit scalaire est nul, la direction de descente est alors parallèle à la contrainte i, et donc $\mu^i_{\max} = \infty$.

[I] L'intérêt de ce type de projection systématique dans le cas classique, est que l'ensemble I^0 ne diffère au plus que d'un élément d'une itération à l'autre de l'algorithme, et ainsi, l'inversion de la matrice $[A^0 A^{0T}]$ peut se faire à partir de l'inverse de cette même matrice calculée à l'itération précédent [Minoux, 1983]. Ici, cette matrice étant toujours l'identité, cette astuce de calcul n'est pas utile.

On définit alors

$$\mu_{\max} = \min_i \mu_{\max}^i$$

Le nouveau point $\boldsymbol{x'}$ est alors obtenu par minimisation unidimensionnelle en μ de la fonction $J(\boldsymbol{x} + \mu\,\boldsymbol{d})$ sur l'intervalle $[0, \mu_{\max}\ \ \mu_{\max}]$. Cette optimisation est réalisée par une dichotomie rapide car son exactitude n'est pas indispensable [Minoux, 1983].

L'algorithme est donc le suivant :

(1) À l'itération $k = 0$ on est en $\boldsymbol{x}^0$

(2) À l'itération courante k on est en $\boldsymbol{x}^k$; on détermine pour ce point l'ensemble $I^0(\boldsymbol{x}^k)$

(3) On calcule la direction de descente d telle que :

$\boldsymbol{d}_i = \nabla J_i(\boldsymbol{x}^k)$ si $i \notin I^0(\boldsymbol{x}^k)$, $\boldsymbol{d}_i = 0$ sinon.

Si $\boldsymbol{d} = 0$ (la direction de descente est nulle) aller en (6)

(4) On détermine le pas maximal autorisé $\mu_{\max}$

(5) Puis on détermine $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mu\,\boldsymbol{d}$ tel que $J(\boldsymbol{x}^{k+1}) < J(\boldsymbol{x}^k)$ par une recherche de μ sur l'intervalle $[0, \mu_{\max}]$.

Si $k < K_{\max} = 100$ et si $|\,\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\,| > 10^{-10}$ aller en (2) sinon aller en (6).

(6) $\boldsymbol{x}^k$ est satisfaisant.

Après cette première optimisation du 1er ordre, on effectue une seconde optimisation avec un algorithme du second ordre, qui est plus efficace pour la convergence finale.

II Algorithme de BFGS projeté

Cet algorithme est proposé dans [Minoux, 1983]. Il consiste à introduire dans le calcul de la direction d'optimisation *une approximation de l'inverse* $\boldsymbol{H}$ *du Hessien* de J. Cet algorithme est beaucoup plus rapide que le précédent : en effet, dans celui-ci, les variables sont modifiées proportionnellement au gradient de la fonction de coût, donc ces modifications tendent vers zéro lorsque l'on s'approche d'un minimum, ou lorsque l'on se trouve sur un plateau, de la fonction de coût.

Ainsi la seule différence avec l'algorithme précédent réside dans le calcul de la direction $\boldsymbol{d}$. À l'itération k ; celle-ci s'écrit :

$$\boldsymbol{d}_i = \boldsymbol{H}^K \ \nabla J_i(\boldsymbol{x}^k) \quad \text{si } i \notin I^0(\boldsymbol{x}^k), \qquad d_i = 0 \text{ sinon.}$$

où $\boldsymbol{H}^K$ est l'approximation de l'inverse du hessien à l'itération k donnée par la formule suivante :

$$\boldsymbol{H}^k = \boldsymbol{H}^{k-1} + [1 + \frac{\boldsymbol{\gamma}_k^T \boldsymbol{H}^k \boldsymbol{\gamma}_k}{\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k}] \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k} - \frac{\boldsymbol{\delta}_k \boldsymbol{\gamma}_k^T \boldsymbol{H}^k + \boldsymbol{H}^k \boldsymbol{\gamma}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k}$$

avec $\boldsymbol{\gamma}_k = \nabla J(\boldsymbol{x}^{k+1}) - \nabla J(\boldsymbol{x}^k)$ et $\boldsymbol{\delta}_k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$

Comme précisé dans [Minoux, 1983], cette approximation nécessite des réinitialisations périodiques avec la matrice identité.

III Conclusion

Les algorithmes proposés ici sont des algorithmes très classiques en optimisation non linéaire. Nous les avons cependant adaptés afin de réduire la quantité de calculs et d'accélérer la vitesse de convergence, ce qui a été rendu possible par des choix judicieux des paramètres de la fonction bosse.

ANNEXE C Hypothèse de similitude du signal S^i et E^i

Pour l'algorithme de GOFR avec des gaussiennes ou des bosses (chapitre 6.II.4 et III.2), nous considérons deux espaces différents : celui des variables, appelé également *représentation temporelle*, et celui des observations, appelé *représentation vectorielle*. Ainsi, le signal à modéliser à l'itération i est l'erreur entre le signal original et le modèle construit avec les i-1 régresseurs par l'algorithme de GOFR ; ce signal est noté E^i.

La sélection de la bosse suivante (m_i), qui permet de poursuivre la modélisation, se fait dans l'espace orthogonal aux régresseurs déjà sélectionnés d'indice (m_1. m_{i-1}). Dans cet espace, le signal à modéliser est S^i qui est le signal représentant l'ECG qui a été orthogonalisé à chaque itération $j<i$.
En théorie, S^i devrait être calculé à partir de E^i, puisque S^i est la partie de E^i qui se trouvant dans l'orthogonal des régresseurs d'indice (m_1. m_{i-1}).

En pratique, nous ne recalculons pas S^i de cette manière. Nous nous contentons de calculer S^i à partir de S^{i-1} par orthogonalisation par rapport au régresseur ajusté $B^*_{m_i}$ (Chapitre 6, équation 27).
On peut évaluer la qualité de cette approximation de S^i par rapport à son estimation exacte et montrer qu'elle est satisfaisante dans tous les cas: en effet, la plus grande erreur quadratique moyenne entre S^i et E^i , qui s'observe à la fin de l'algorithme de sélection-apprentissage des bosses, c'est-à-dire pour i=6, est un ordre de grandeur inférieur à l'erreur quadratique moyenne de modélisation. Ceci est illustré par les Figures 1 et 2.

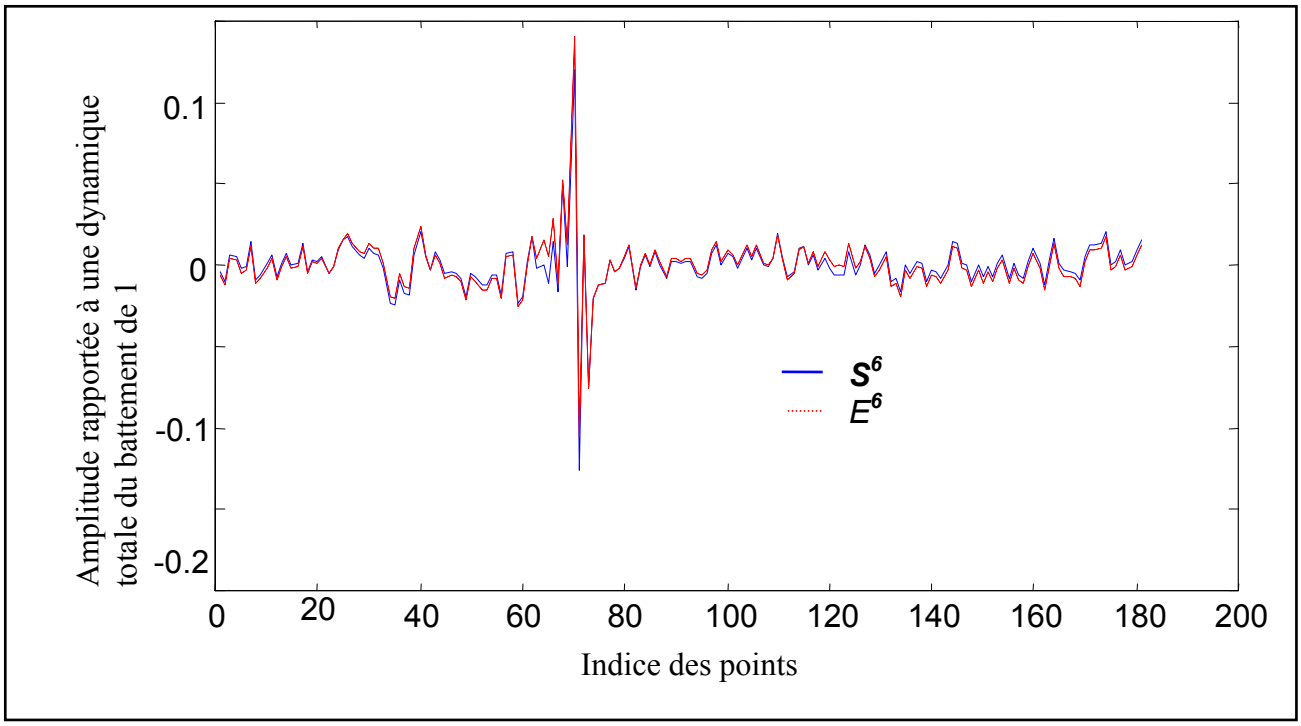


Figure 1 : Tracé des signaux S^6 et E^6 après l'apprentissage de 6 bosses. Les deux signaux sont quasiment superposés ; ils ont une différence quadratique moyenne de l'ordre de 1.10^{-5}. L'erreur de modélisation pour ce battement est de 3.10^{-4}.

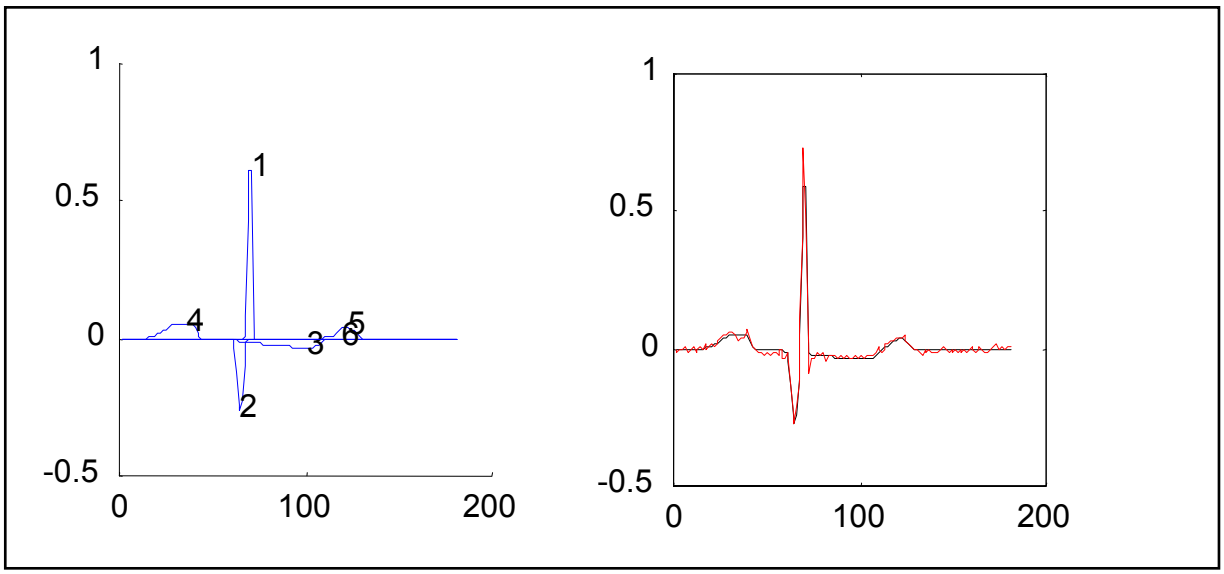


Figure 2 : Modèle en bosses d'un battement normal. L'erreur quadratique moyenne de modélisation est de 3.10^{-4}.

ANNEXE D Résultats de la classification non supervisée

On présente ici les résultats de l'algorithme d'agrégation exposé au chapitre 7. Il a été testé sur les enregistrements de 30mn de la base MIT.
Pour juger de la qualité deux critères sont retenus :

- *l'homogénéité des familles* : on souhaite rassembler les battements de même origine,
- le *nombre de familles* total.

La meilleure classification suivant ces critères est celle qui fournit finalement :

- une famille pour les battements normaux,
- une famille par type d'extrasystole ventriculaire,
- et éventuellement quelques familles supplémentaires pour des battements de formes atypiques (anormalement larges par exemple).

Pour rappel, chaque enregistrement est traité par paquets de 1200 battements (~20mm), c'est-à-dire que tous les 1200 battements, les familles sont réinitialisées ; en conséquence, le cardinal maximum d'une famille est de 1200.

Les résultats sont présentés dans le tableau 1 :

- *Nt* : nombre total de battements analysés,
- *Nnc* : nombre de battements non classés : les deux voies d'enregistrements sont jugées trop bruitées pour que le battement puisse être valablement analysé.
- *Ns* : nombre de paquets de 1200 battements,
- *N1v* : nombre de familles « 1 voie » créées : une des deux voies est trop bruitée pour être analysée, la mise en famille ne s'est donc effectuée qu'à partir de la voie disponible.
- *N2v* : nombre de famille « 2 voies » créées,
- *NE1v* : nombre d'erreurs de classification pour les battements classés sur 1 voie : correspond au nombre de battements d'origine sinusale classés dans une famille ventriculaire ou inversement.
- *NE2v* : nombre d'erreurs de classification pour les battements classés sur 2 voies,

- *Err* : correspondant au pourcentage des erreurs cumulées de classification 1 voie et 2 voies :

$$Err = \frac{(NE1v + NE2v)}{Nt - Nnc}.100$$

- *Errc :* taux d'erreur corrigé. Les labels de la base MIT sont plus riches que la simple distinction *N/V*. Par exemple, on trouve le label *F* qui correspond à des battements de fusion entre un battement normal et un battement ventriculaire. Ces battements sont parfois de formes identiques aux battements normaux, et parfois de formes identiques aux battements ventriculaires (Figure 1); l'algorithme ne fait pas cette distinction ici, car les paramètres qu'il utilise pour la classification ne sont pas pertinents pour permettre une telle séparation. De même, le label *J* correspond aux battements ayant pour origine une décharge du nœud auriculo-ventriculaire (AV) (c.f. Chapitre 2.II.2.1) ; excepté l'absence de l'onde P, la forme de la dépolarisation ventriculaire est identique à une dépolarisation normale, d'origine sinusale : il est donc normal que l'algorithme de classification rassemble de tels battements avec les battements normaux. Ainsi pour chaque battement mal classé étiqueté dans la base par F, J, e, j, a [I], on a vérifié le classement et ainsi calculé le taux corrigé *Errc*.

[I] Les principaux labels de la base MIT sont les suivants : *N* : battement normal ; *A* : battement auriculaire prématuré ; *a* : battement auriculaire prématuré isolé ; *J* : battement prématuré jonctionnel ; *S* : extrasystole supraventriculaire ; *V* : extrasystole ventriculaire ; *F* : battement de fusion entre une extrasystole ventriculaire et battement normal ; *j* : battement d'échappement jonctionnel.

Nom	*Nt*	*Nnc*	*Ns*	*N1v*	*NE1v*	*N2v*	*NE2v*	*Err*	*Errc*
MIT_100	2268	0	2	1	0	7	0	0%	0%
MIT_101	1863	30	2	4	0	0	0	0%	0%
MIT_103	2080	10	2	3	0	18	0	0%	0%
MIT_105	2593	244	3	47	4	10	0	0,17%	0,17%
MIT_106	2032	45	2	45	5	52	0	0,25%	0,25%
MIT_107	2131	3	2	21	3	41	0	0,14%	0,14%
MIT_108	1764	178	2	30	3	40	0	0,19%	0,06%
MIT_109	2528	2	3	9	1	24	0	0,04%	0,04%
MIT_111	2119	11	2	13	0	17	0	0%	0%
MIT_112	2535	15	3	10	0	10	0	0%	0%
MIT_113	1788	42	2	9	1	6	1	0,11%	0,11%
MIT_114	1874	4	2	4	0	19	3	0,16%	0,05%
MIT_115	1950	11	2	4	0	4	0	0%	0%
MIT_116	2409	25	3	7	0	10	0	0%	0%
MIT_117	1531	9	2	1	0	16	0	0%	0%
MIT_118	2273	1	2	6	0	20	0	0%	0%
MIT_119	1984	2	2	7	0	6	0	0%	0%
MIT_121	1859	39	2	6	0	12	0	0%	0%
MIT_122	2472	4	3	3	0	4	0	0%	0%
MIT_123	1514	0	2	1	0	17	0	0%	0%
MIT_124	1615	16	2	6	0	19	3	0,19%	0,06%
MIT_200	2599	71	3	40	9	25	0	0,36%	0,36%
MIT_201	1938	6	2	19	1	23	2	0,16%	0,16%
MIT_202	2133	2	2	11	0	16	0	0%	0%
MIT_203	3044	541	3	129	79	69	18	3,88%	3,2%
MIT_205	2652	40	3	15	3	8	0	0,11%	0%
MIT_207	2257	446	2	40	8	33	0	0,44%	0,28%
MIT_208	2936	105	3	37	104	24	20	4,38%	1,94%
MIT_209	2998	39	3	11	1	13	0	0,03%	0,03%
MIT_210	2642	85	3	43	6	46	3	0,35%	0,35%
MIT_212	2744	23	3	15	0	15	0	0%	0%
MIT_213	3246	21	3	18	25	31	201	7,01%	1,09%
MIT_214	2258	21	2	20	2	39	0	0,09%	0,04%
MIT_215	3359	18	3	17	1	29	0	0,03%	0,03%
MIT_219	2150	99	2	10	5	33	36	2%	2%
MIT_220	2044	0	2	2	0	3	0	0%	0%
MIT_221	2425	17	3	16	0	20	0	0%	0%
MIT_222	2486	145	3	16	0	15	0	0%	0%
MIT_223	2601	31	3	18	37	41	23	2,33%	2,22%
MIT_228	2048	3	2	22	0	28	0	0%	0%
MIT_230	2252	0	2	4	0	49	0	0%	0%
MIT_231	1564	0	2	3	0	6	0	0%	0%
MIT_232	1782	4	2	15	0	31	0	0%	0%
MIT_233	3074	57	3	35	3	49	5	0,27%	0,03%
MIT_234	2748	20	3	6	0	12	0	0%	0%
TOTAL	**103162**	**2485**	**-**	**799**	**301**	**1010**	**315**	**0,61%**	**0,31 %**

Tableau 1 : Résultat de la mise en famille sur la base MIT.

I Battement de fusion (label F)

Comme mentionné plus haut, un certain nombre de battements de la base MIT sont étiquetés *F*. Ce sont des battements de fusion, c'est-à-dire des battements qui « mélangent » une extrasystole ventriculaire et une dépolarisation supraventriculaire. Pour le taux d'erreur *Err*, ces battements sont systématiquement étiquetés comme erreurs de classification dès lors qu'ils sont mélangés avec des battements étiquetés *N* ou *V*. Pour calculer le taux d'erreur corrigé, nous avons regardé une à une ces erreurs pour valider ou non la réponse de l'algorithme.

La Figure 1 montre deux battements étiquetés *F* de l'enregistrement 213. D'après le tableau 1, cet enregistrement comporte 201 erreurs de classification pour les familles « 2 voies ». En fait, on constate que les résultats de la classification sont en réalité corrects par rapport à la classification attendue.

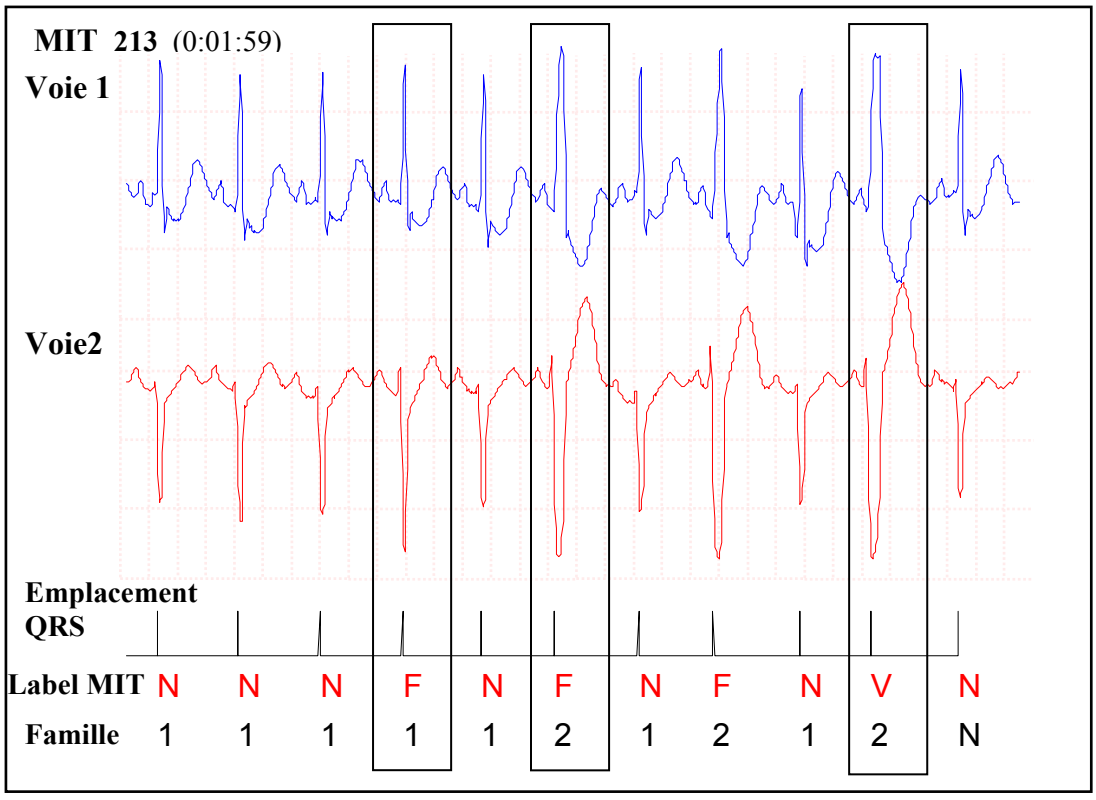


Figure 1 : Les battements étiquetés F sont des battements de fusion, c'est-à-dire « mélangeant » un battement d'origine sinusal avec une dépolarisation ventriculaire. Pour le taux d'erreur Err, ces battements sont comptabilisés comme mal classés, cependant ici, il est correct d'associer le premier battement F avec la famille 1 (battements normaux) et le deuxième avec la famille 2 (extrasystoles ventriculaires). Le taux d'erreurs corrigé Errc tient compte de telles corrections.

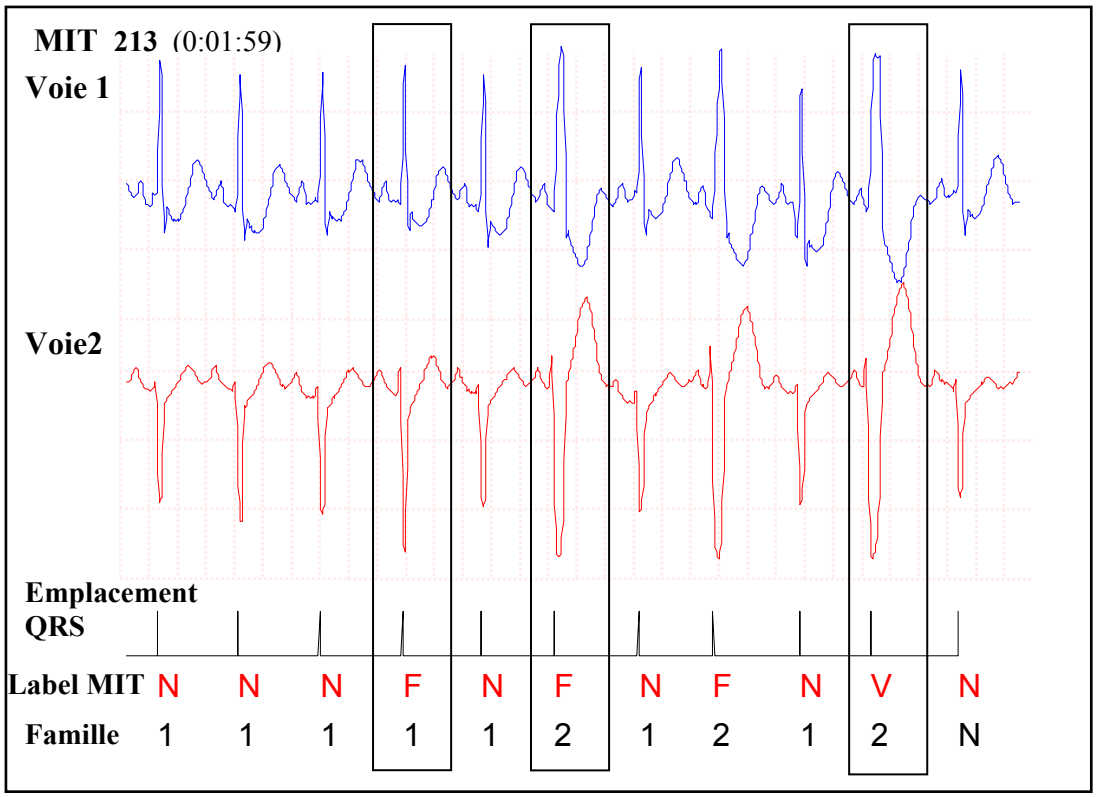


Figure 1 : Les battements étiquetés F sont des battements de fusion, c'est-à-dire « mélangeant » un battement d'origine sinusal avec une dépolarisation ventriculaire. Pour le taux d'erreur Err, ces battements sont comptabilisés comme mal classés, cependant ici, il est correct d'associer le premier battement F avec la famille 1 (battements normaux) et le deuxième avec la famille 2 (extrasystoles ventriculaires). Le taux d'erreurs corrigé Errc tient compte de telles corrections.

II Erreurs de classification

II.1 Analyse sur une voie unique

Lors de l'analyse, certaines voies d'enregistrement peuvent être écartées pendant quelques battements. L'analyse, pendant ce temps, continue sur les voies restantes.

Lorsque, sur la base MIT, une voie est trop bruitée, il ne reste donc plus qu'une seule voie pour effectuer la classification, et la distinction *N/V* sur une seule voie s'avère parfois difficile. De nombreuses erreurs de classification proviennent précisément de classifications réalisées sur 1 voie (*NE1v*) (Figure 2 et 3).

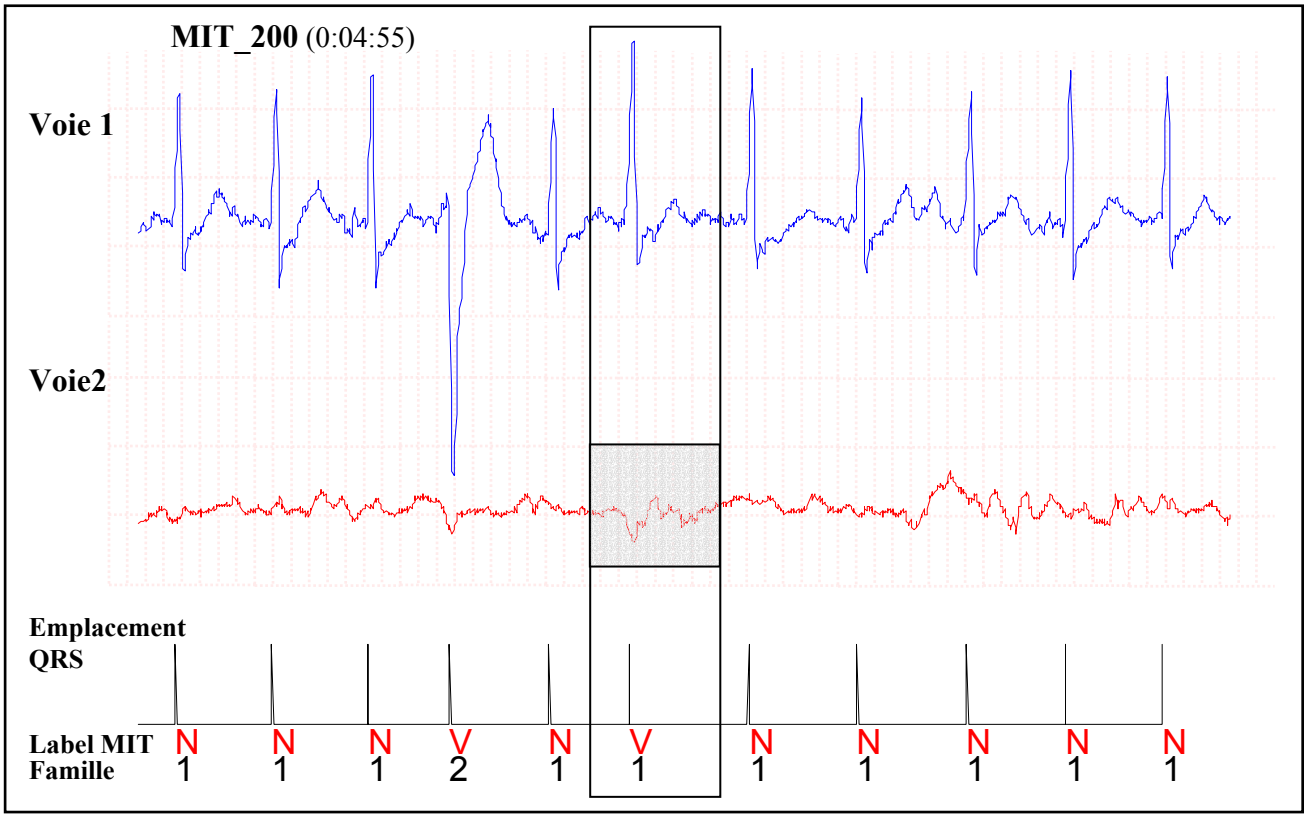


Figure 2 : La voie 2 étant très bruitée, elle ne participe pas à l'analyse. L'erreur de classification est difficilement évitable sur cet exemple.

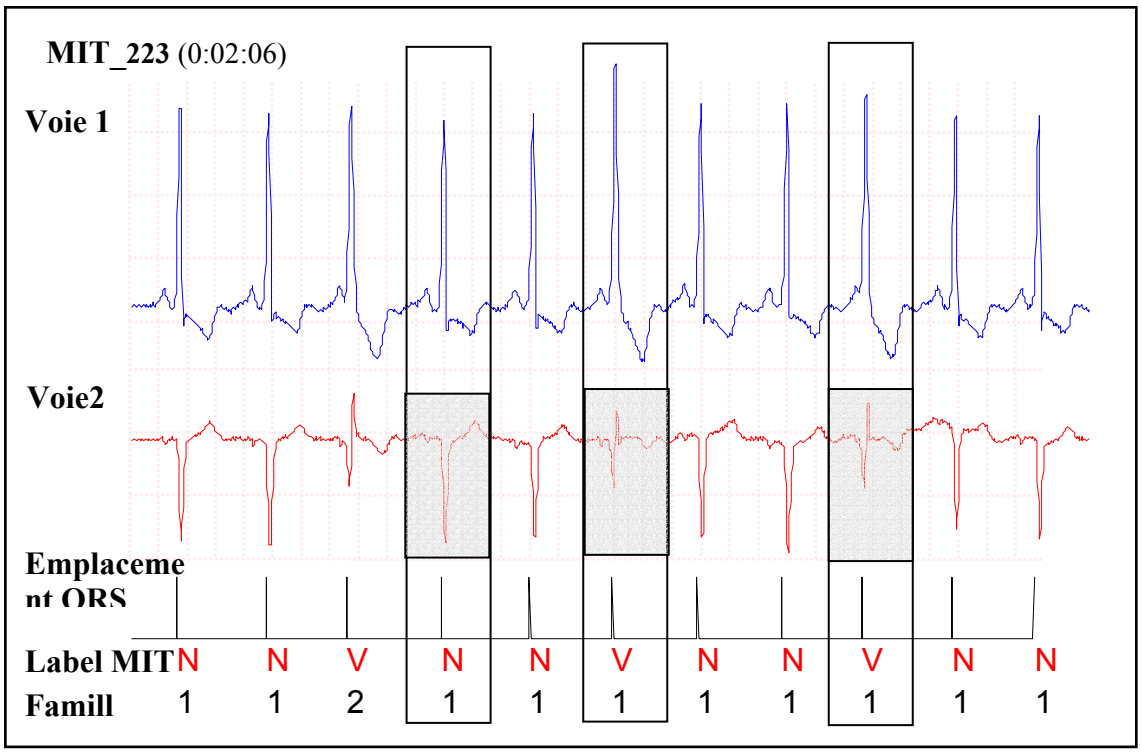


Figure 3 : La voie 2 est, ici encore, exclue de l'analyse à cause de larges variations de la ligne de base non représentées ici. Au regard de la seule voie 1, il est difficile d'effectuer correctement la distinction N/V.

II.2 Erreur en 2 voies

Nous présentons ici un type d'erreur heureusement peu fréquent : sa contribution au taux d'erreur n'est pas très élevée (environ 40 erreurs au total sur l'ensemble de la base MIT). Cette erreur correspond à un battement mal classé à cause du calcul des distances : celui-ci révèle que ce battement est plus proche d'une famille d'un label différent que d'une famille dont le label lui conviendrait mieux. Bien que rare, une erreur de ce type est définitive, car elle ne pourra être corrigée dans la suite de l'algorithme tel que nous l'avons construit. Au regard de l'ecg Figure 4, on comprend la difficulté à bien classer le deuxième battement encadré, celui-ci étant très proche sur la voie 1 du battement de la famille 2, les intervalles RR sont également identiques, et l'angle de l'axe ACP quasiment dans la même direction. Seule la voie 2 permet ici de distinguer ces deux familles. Une méthode à terme pour résoudre ce type d'erreur pourrait être de repérer pour chacune des familles les éléments qui la distingue réellement des autres pour lui donner plus d'importance dans le critère de décision.

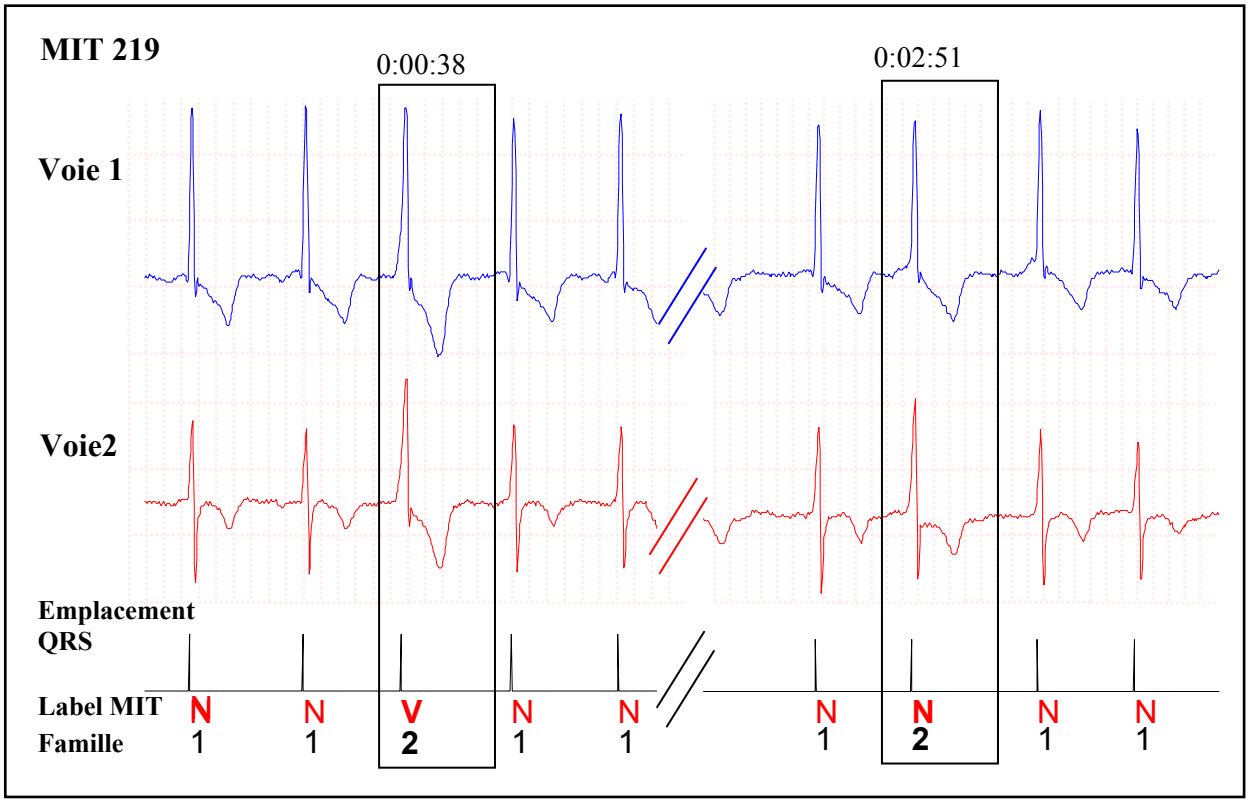


Figure 4 : Une battement normal a été associé à une famille de battements ventriculaires. La distance calculée entre ce battement et la famille 2 est en effet plus faible que la distance entre ce battement et la famille 1.

ANNEXE E Réseau de neurones pour la localisation des bosses modélisant l'onde R

Les familles de battements sont représentées par un prototype qui a été modélisé en bosses par l'algorithme présenté au chapitre 6. Afin d'associer à chacune de ces familles un label *N* ou *V* en fonction du type de battements qu'elle représente, il est indispensable de repérer l'onde R pour analyser sa forme.

Le réseau de neurones que nous présentons ici effectue cette tâche. Il fournit la probabilité qu'une bosse B de paramètres $\boldsymbol{x} = \left[{}^{1}\sigma, {}^{2}\sigma, {}^{L}\sigma, \mu, A \right]^{T}$ modélise une onde R. En appelant C_R la classe des bosses modélisant une onde R, le réseau calcule $P_{RN}(C_R \mid \boldsymbol{x})$ qui est une estimation de $P(C_R \mid \boldsymbol{x})$.

I Architecture du réseau

Le réseau est un perceptron multicouche possédant une couche de neurones cachés dont le nombre *Ncc* est à déterminer. Les entrées du réseau sont au nombre de 6 : les 5 paramètres de la bosse (vecteur $\boldsymbol{x}$) et un biais. La sortie est un neurone unique (Figure 1).

Les fonctions d'activation des neurones cachés et du neurone de sortie sont des sigmoïdes.

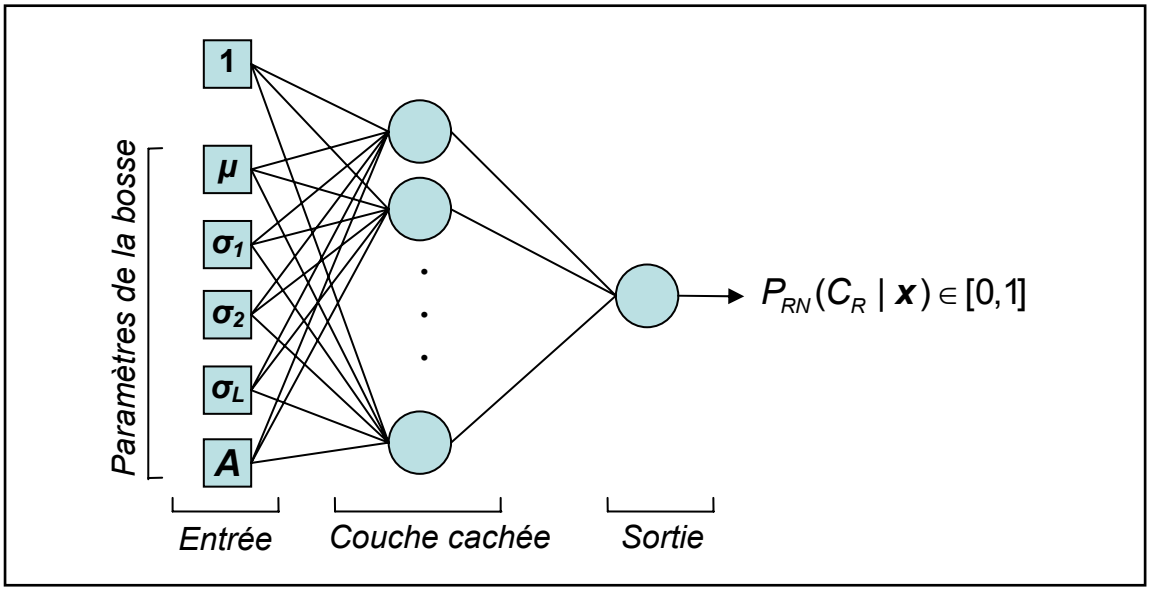


Figure 1 : Architecture du réseau de neurones utilisé pour le repérage des bosses qui modélisent les ondes R. Il permet de calculer $P_{RN}(C_R \mid \boldsymbol{x})$, qui constitue une estimation de la probabilité pour qu'une bosse caractérisée par les paramètres $\boldsymbol{x}$ soit une bosse qui modélise l'onde R. Le nombre de neurones cachés est un paramètre du réseau à déterminer.

II Base d'apprentissage et base de test

La base MIT a permis la création de deux bases de données : une pour l'apprentissage, l'autre pour le test.
Chaque enregistrement de la base a été analysé : nous avons utilisé notre algorithme de mise en famille présenté au chapitre 7 pour rassembler les battements identiques. Nous disposons donc, pour chaque enregistrement, d'un certain nombre de famille ; chacune d'elle est modélisée en bosses.
Pour chacune de ces familles, nous avons annoté les bosses modélisant l'onde R. La base MIT comprend 9037 bosses, dont 1920 modélisent une onde R.

La basse d'apprentissage est donc constituée de la moitié des 1920 bosses modélisant une onde R, soit 960 et d'autant de bosses ne modélisant pas une onde R. Le total *d'exemples* de la base d'apprentissage est donc de Np = 1920.
Le reste des bosses est utilisé pour le test : soit 960 bosses modélisant une onde R et 7117 bosses ne modélisant pas une onde R.

III Déroulement de l'apprentissage

L'apprentissage est réalisé par la minimisation d'une fonction de coût choisie ici comme l'erreur quadratique moyenne[L] sur la sortie : chaque bosse k modélisant une onde R se voit attribuer la valeur de sortie désirée $L(k)=1$, les autres bosses ayant la valeur désirée $L(k)=0$.

La fonction de coût s'écrit de la manière suivante :

$$J(\theta) = \frac{1}{N_P} \sum_{k=1}^{N_p} \left(L(k) - P_{RN}(C_R \mid \boldsymbol{x}^k) \right)^2$$

[L] Il existe d'autres fonctions de coût possible pour les problèmes de classification comme l'entropie croisée par exemple [Dreyfus, 2002], [Bishop, 1995]. Cette dernière a été également testée, mais, suite aux expériences effectuées, les résultats s'avèrent dans notre cas moins intéressants. En outre un avantage d'utiliser l'erreur quadratique moyenne comme fonction de coût est de permettre l'utilisation de l'algorithme de Levenberg-Marquardt, particulièrement efficace pour l'apprentissage [Levenberg, 1944]

où θ est le vecteur des *paramètres du réseau* (les poids entre les unités qui le constituent), $\boldsymbol{x}^k$ le vecteur des *paramètres de la bosse* d'indice k, et N_p le nombre total de bosses de l'ensemble d'apprentissage, soit $N_p = 1920$.

La minimisation de la fonction J est réalisée par ajustement des paramètres θ du réseau par l'algorithme de Levenberg-Marquardt [Levenberg, 1944], [Dreyfus, 2002].

IV Résultats de l'apprentissage

Afin de déterminer le nombre optimal de neurones cachés (Ncc) nous allons effectuer des apprentissages pour différentes architectures : pour chacune des architectures, nous effectuons 30 initialisations différentes des paramètres, ce qui conduit à 30 modèles par architecture.
Une base de validation nous permet de choisir parmi l'ensemble de ces architectures celle qui est la plus adaptée à notre problème.

IV.1 Critère de mesure de la qualité de l'apprentissage

La qualité de l'apprentissage est mesurée par le nombre de bosses de la base *bien classées*, *mal classées* et *non classées,* d'après un critère fondé sur la valeur de la probabilité en sortie du réseau.

- Si $P_{RN}(C_R \mid \boldsymbol{x}) > 0{,}8$ alors la bosse de paramètre x est classé comme appartenant à C_R.
- Si $0.2 < P_{RN}(C_R \mid \boldsymbol{x}) < 0{,}8$ alors la bosse n'est pas classée.
- Si $P_{RN}(C_R \mid \boldsymbol{x}) < 0{,}2$ alors la bosse est classée comme n'appartenant pas à C_R.

Les résultats pour ce type de décision sont présentés dans le paragraphe suivant (Figure 2 – Critère 1). Cependant, nous avons deux hypothèses supplémentaires qui peuvent compléter notre arbre de décision: d'une part, on suppose que chaque prototype possède au moins une bosse qui modélise l'onde R^{LI} ; d'autre part on suppose que deux bosses au plus, parmi les six qui modélisent le battement, modélisent l'onde R (dans le cas d'un complexe biphasique ou

[LI] Ce qui correspond à une réalité physiologique dans tous les cas, sauf situation pathologique rarissime évoquée en note II au chapitre 8

d'un bloc de branche par exemple). Ainsi, on peut établir un deuxième critère (critère 2) de décision, fondé sur une étude comparative des probabilités des six bosses modélisant le battement :

- Si, parmi les six bosses, au moins deux ont une probabilité supérieure à 0,6 de modéliser une onde R, alors les deux plus probables sont affectées à la classe C_R ; les autres sont classées comme n'appartenant pas à C_R.
- Si une seule bosse présente une probabilité $P_{RN}(C_R \mid \boldsymbol{x})$ supérieure à 0,6 alors elle seule est affectée à C_R.
- Si aucune bosse ne se voit attribuer de probabilité supérieure à 0,6, mais que plusieurs bosses possèdent une probabilité supérieure à 0,1 alors les deux plus probables sont affectées à la classe C_R ; les autres sont classées comme n'appartenant pas à C_R.
- Dans tous les autres cas, la bosse qui a la probabilité la plus élevée est affectée à C_R.

C'est ce second critère de décision qui a été finalement retenu (Figure 2 – Critère2) car l'introduction de connaissance à ce niveau conduit à de meilleurs résultats.

IV.2 Résultats sur la base d'apprentissage

Pour un nombre de neurones cachés *Ncc* fixé, nous avons réalisé 30 apprentissages successifs avec des initialisations différentes des paramètres du réseau.
La Figure 2 ci-dessous présente les résultats sur la base d'apprentissage de la meilleure initialisation des poids du réseau pour *Ncc* compris entre 1 et 15[LII].

[LII] Pour Ncc=15 neurones cachés, le réseau est constitué de 105 paramètres ajustables, ce qui est acceptable compte tenu de la taille de la base d'apprentissage : 1920 exemples [Dreyfus, 2002].

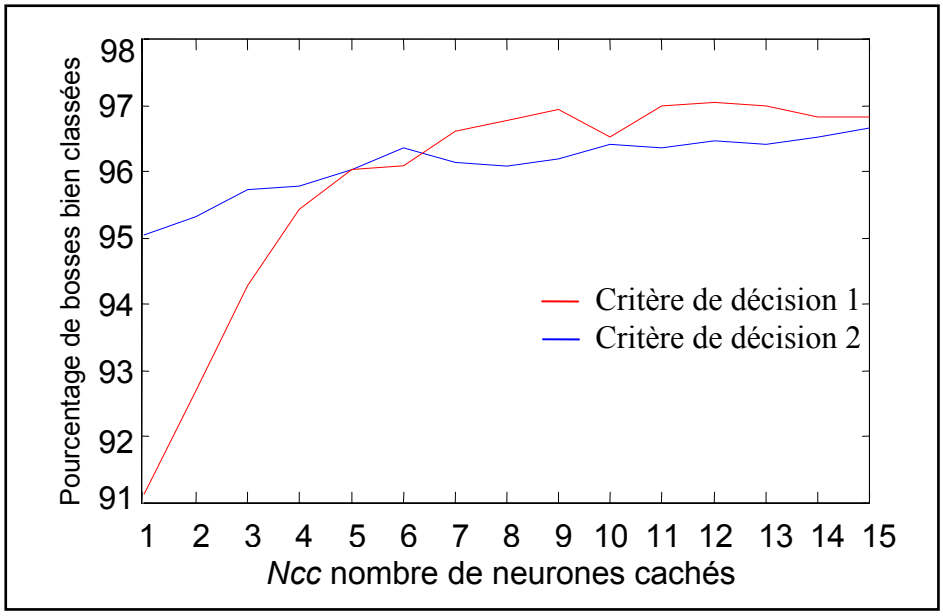


Figure 2 : Pour chaque architecture (Ncc fixé entre 1 et 15), 30 initialisations et autant d'apprentissages ont été réalisés. Le meilleur d'entre eux pour chaque Ncc est présenté ici. La qualité a été jugée suivant les deux critères de décision présentés ci-dessus.

Nous sélectionnons ici le meilleur réseau pour *Ncc* fixé parmi les 30 initialisations des poids. On constate que plus le nombre de neurones cachés est important, plus le nombre de d'exemples bien classés (quel que soit le critère de décision) est grand car le réseau, grâce au nombre de paramètres croissant se spécialise sur la base d'apprentissage.

Pour éviter ce *surajustement*, nous allons choisir le *Ncc* optimal à partir de la base de validation.

IV.3 Résultats sur la base de validation

La Figure ci-dessous représente les résultats obtenus par les réseaux sélectionnés précédemment sur la base d'apprentissage.

Le meilleur résultat est obtenu pour *Ncc* = 4, qui correspond à 97.3% de bosses bien classées.

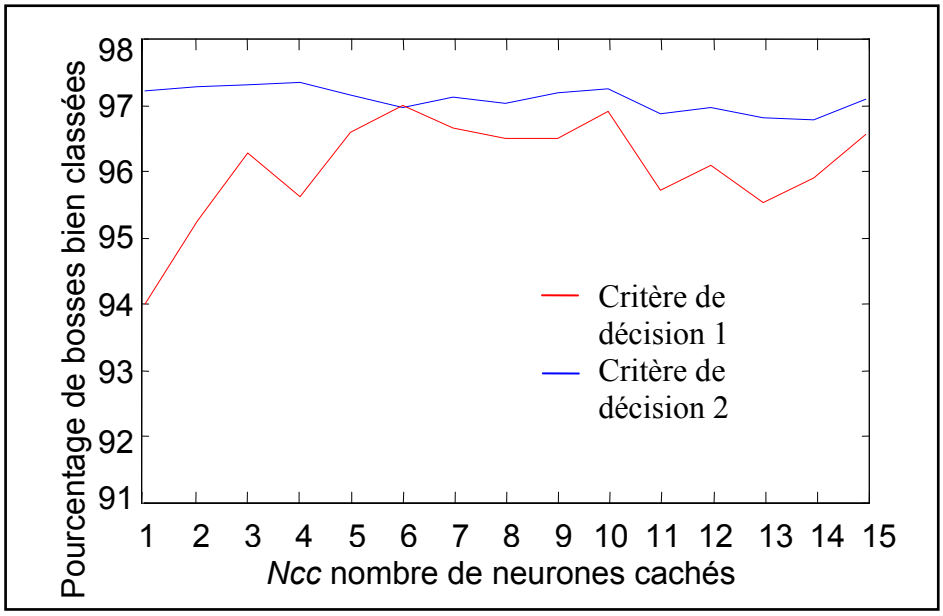


Figure 3 : Résultat des meilleurs réseaux (sélectionnés précédemment) sur la base de validation en fonction du nombre de neurones cachés. La meilleure architecture, selon le critère de décision 2, est celle qui possède 4 neurones cachés.

ANNEXE F Étiquetage des battements N / V

Cette annexe présente en détail l'arbre de décision qui permet l'association à chacune des familles regroupant les battements d'un label parmi les labels suivants : *N*, *V*, *L*, et *?*.

- *N* : pour les battements dont l'origine est une dépolarisation supraventriculaire,
- *V* : pour les extrasystoles ventriculaires,
- *L* : pour les complexes QRS larges qui ne sont classés ni comme *N*, ni comme *V*,
- *?* : pour les familles qui n'ont pu être associées à aucun des labels précédents.

I Arbre de décision

L'arbre de décision traduit la connaissance experte sur les différents paramètres du représentant d'une famille, paramètres qui sont ici au nombre de 5 :

1) la largeur de l'onde R (L_R), qui est définie en fonction des paramètres de la bosse (cf. Chapitre 8 I.1.3 rassemblement des bosses)
2) une valeur discrète (M_{RR}) définie ci-dessous, qui rend compte d'une éventuelle rupture locale de rythme,
3) une valeur discrète (M_A), également définie ci-dessous, qui rend compte de différences d'amplitudes
4) une valeur relative ($L_{RN} = L_R / L_{Réf}$) de la largeur de l'onde R, qui correspond à sa largeur rapportée à la largeur de l'onde R de la famille de référence des battements normaux,
5) Une valeur discrète M_C qui traduit la corrélation entre ce battement et le battement normal le plus récent.

I.1 Mesure de la position RR de la famille (M_{RR})

Lors de la création des familles, nous avons mémorisé, pour chacune d'elles, un certain nombre de paramètres, et notamment les intervalles *RR* du représentant avec le battement qui le suit (*RRs*) et le battement qui le précède (*RRp*) ; on mémorise aussi le rythme *RR* moyen au moment de la création de la famille par ce battement qui en devient le représentant.Un paramètre utile pour la discrimination *N/V* des battements est la possible rupture locale du rythme lors de l'apparition du battement : les extrasystoles ventriculaires arrivent souvent de manière prématurée par rapport au rythme normal, et sont souvent suivies d'un repos compensateur.

Afin de tenir compte de ces 3 paramètres, nous calculons, pour chaque famille, un nouveau paramètre noté M_{RR} qui possède une valeur entière comprise entre -1 et 3.

Si les intervalles *RR*, *RRs* et *RRp* ont des valeurs très peu différentes (différence relative <15%) on associe à M_{RR} la valeur -1. Inversement dans le cas d'une extrasystole prématurée avec repos compensatoire, la valeur associée à M_{RR} est 3 (Figure 1) ; les valeurs intermédiaires correspondent à des degrés moindres de différences.

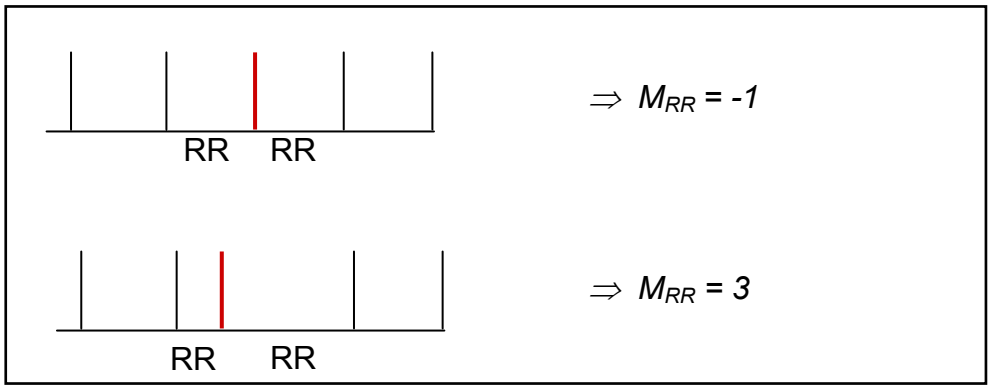


Figure 1 : Mesure de la régularité de la position RR par un nombre entre -1 et 3. Les familles de battements qui ont des intervalles RR irréguliers ont une valeur de M_{RR} élevée, alors que celles qui représentent des battements réguliers ont une valeur faible.

I.2 Mesure des différences d'amplitudes (M_A)

L'amplitude du QRS de part et d'autre de la ligne de base sur chacune des pistes valides de l'enregistrement est également un critère utile pour la séparation *N/V* lorsqu'on le rapporte à l'amplitude de la famille de référence.

Ainsi, nous calculons les différences d'amplitude par un paramètre M_A prenant une valeur entière dans l'intervalle [0,4].

M_A prend la valeur 0 lorsque, sur l'ensemble des voies valides, les différences d'amplitude avec la famille de référence sont négligeables. Inversement, on attribue à M_A la valeur 4 lorsque les amplitudes sont très différentes (Figure 2).

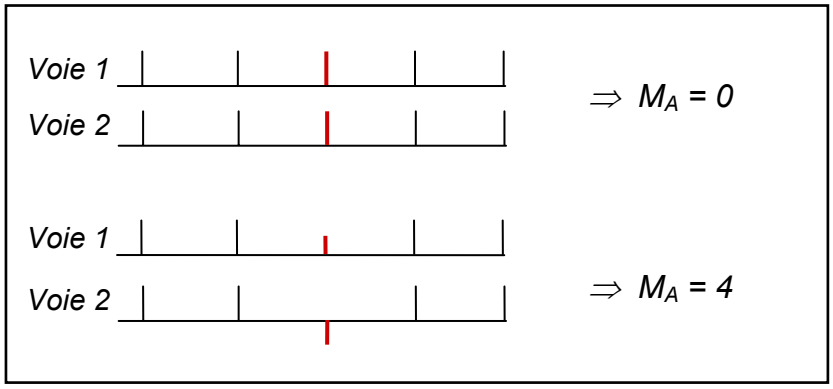


Figure 2 : Une comparaison des amplitudes entre le battement représentant d'une famille et celui de la famille de référence permet de calculer le coefficient M_A, dont la valeur est d'autant plus élevée que le nombre de différences d'amplitudes sur les voies valides sont importantes.

I.3 Mesure discrète de la corrélation

Pour certaines familles, il est difficile de se prononcer directement sur le label (cf. I.4 Décision). Pour prendre la décision, on s'aide d'un calcul de corrélation entre le battement représentant la famille et le battement étiqueté N le plus récent. En fonction du résultat de ce calcul, on attribue au paramètre M_C une valeur entière entre 0 et 3. 0 étant associé aux corrélations supérieures à 0,8 et 3 à celles inférieures à 0.2.

Cette valeur est particulièrement importante car elle permet de s'adapter au contexte du moment.

I.4 Décision

L'algorithme d'attribution des labels se déroule essentiellement en 2 étapes. La première peut être qualifié de *décision absolue* (Figure 3) : une première décision est prise, fondée sur les seuls paramètres absolus, c'est-à-dire sur les paramètres qui ne dépendent pas du patient, ici

L_R et M_{RR}[I]. Par exemple, une famille de battements réguliers, dont la largeur de l'onde R est inférieure à 80 ms [Houghton, 1997] sera automatiquement étiquetée *N*, ou une famille dont le battement représentatif est prématuré, suivi d'un repos compensatoire et dont la largeur de l'onde R est supérieure à 80 ms sera étiquetée *V*.

La deuxième étape, ou *décision relative*, s'applique aux familles qui n'ont pas reçu de label à la première étape (Figure 4). Ici, pour chaque famille non étiquetée, on étudie, outre L_R et M_{RR}, des paramètres relatifs au patient : M_A, L_{RN}, et M_C . Nous avons créé des classes en fonction de la valeur de chacun de ces paramètres à partir de la connaissance médicale dont nous disposons. L'appartenance à ces classes n'est pas exclusive, ainsi on examine les classes une à une, et dès que les paramètres d'une famille correspondent à une classe, le label est associé à la famille.

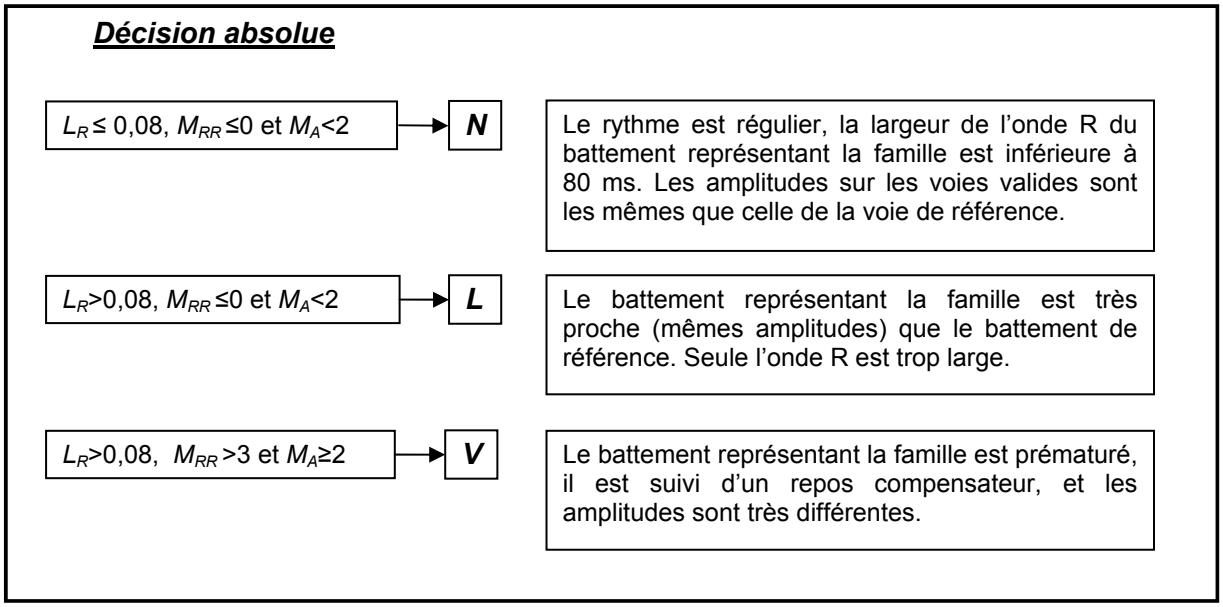


Figure 3 : La décision absolue associe un label aux familles par application de règles sur les critères non ramenés au patient.

[I] Une légère contrainte sur le paramètre M_A est également introduite à ce niveau pour ne pas étiqueter *N* des ESV interpolées dont les projections restent peu larges.

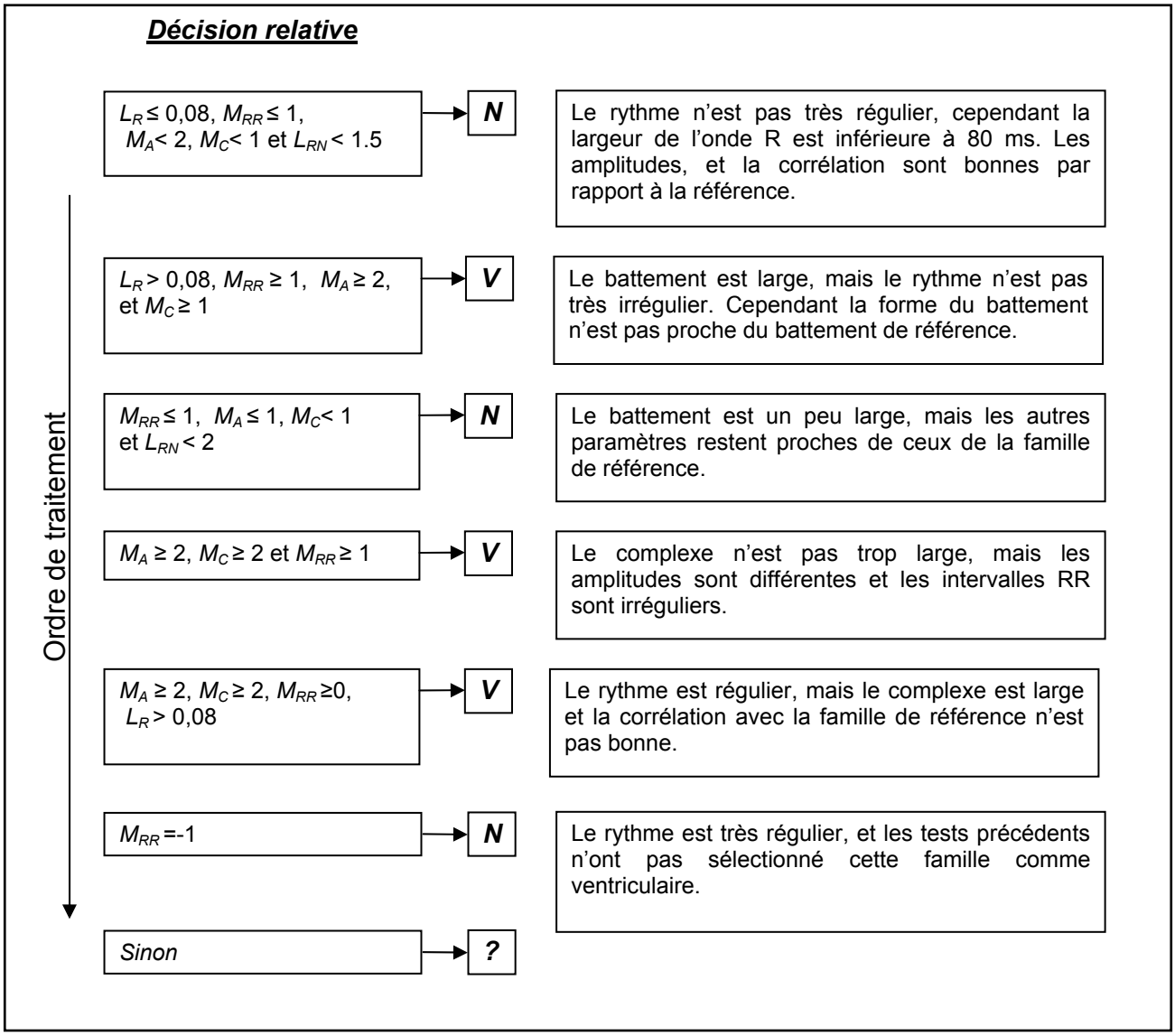


Figure 4 : la décision relative ajoute aux paramètres de décision précédents des paramètres relatifs, ramenés aux patients. Chaque critère est vérifié un à un ; dès qu'un label est trouvé, on passe à la famille suivante.

II Résultats sur les bases MIT et AHA

Les résultats sur les enregistrements de la base MIT sont décrits dans le tableau 1, ceux sur la base AHA dans le tableau 2.

Les grandeurs suivantes sont présentées pour chaque enregistrement :

- *NT* est le nombre de battements présentés à l'algorithme,

- *NTe* est le nombre de battements que l'algorithme a étiquetés, la différence entre *NT* et *NTe* correspond donc au nombre de battements que l'algorithme n'a pas étiquetés car ils ont été exclus soit pendant les phases de détection des QRS, soit lors de l'analyse en composantes principales à cause du bruit trop élevé,
- *Ne* est le nombre d'erreurs d'étiquetage, où un battement a été étiqueté *N* alors qu'il est d'origine ventriculaire, ou inversement. Un détail de ces erreurs est donné par les grandeurs qui suivent,
- *NVP* : correspond aux battements normaux vrais positifs pour le label *N*, c'est-à-dire les battements dont le label de référence dans la base MIT est *N* ou *A*, et également étiqueté *N* par l'algorithme,
- *VVP* correspond aux battements étiquetés *V* dans la base MIT, et également étiqueté *V* par l'algorithme,
- *NFN* est le nombre de faux négatifs pour la famille *N* : battements étiquetés *N* dans la base, qui n'ont pas été étiquetés *N* par l'algorithme,
- *NFP* est le nombre de faux positifs pour la famille *V* : battements étiquetés *N* par l'algorithme alors qu'ils n'ont pas ce label dans la base,
- *VFN* et *VFP* sont les grandeurs équivalentes pour les familles ventriculaires.

Il est important de noter que les erreurs sont doublement comptabilisées à ce niveau : par exemple, lorsque qu'un battement qui a comme label de référence le label *N* est étiqueté *V* par l'algorithme, cette erreur est comptabilisée à la fois par *NFN* car on n'a pas étiqueté ce battement *N* alors qu'il a ce label, et par *VFP* car on a étiqueté ce battement *V* alors qu'il n'a pas ce label. Les taux de réussite sont donc calculés à partir des colonnes *NTe* et *Ne*.

Les deux colonnes suivantes comptabilisent les battements étiquetés *L* et *?*. La dernière colonne indique pour information le nombre de battements de la base annotés avec le label *F* correspondant aux battements de fusion. Ces battements ont été classés avec le label *N, V, L* ou *?*, mais comme le label exact dépend des cas, ils ne sont comptabilisés ni dans le calcul d'erreur, ni dans le calcul de réussite (cf. Annexe D Figure 1).

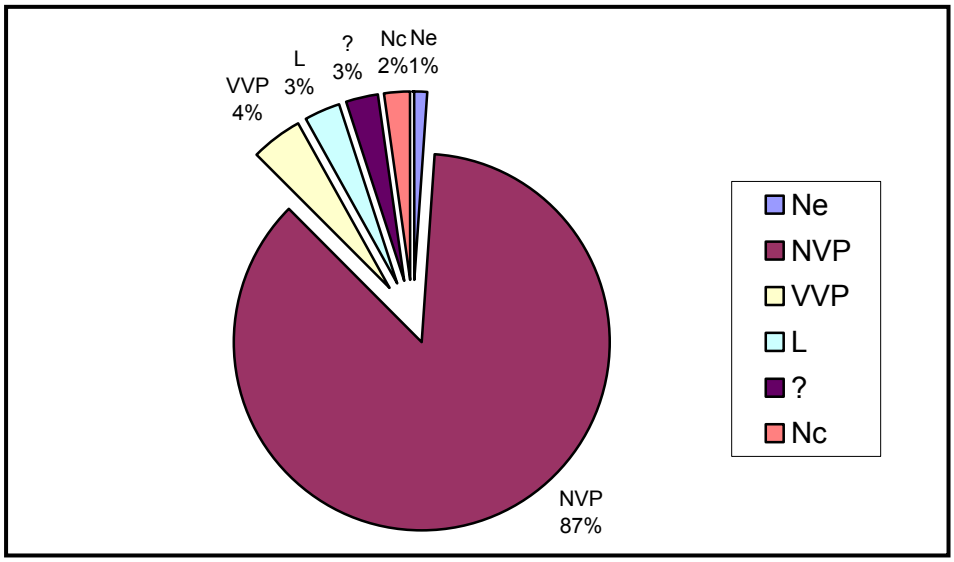


Figure 5 : Synthèse des résultats sur la base MIT pour cent battements. Le taux de réussite pour les labels N et V est de 91%, 3% des battements sont étiquetés avec le label L, 3% avec le label ?, 2% ne sont pas étiquetés et 1% sont mal étiquetés.

NOM	NT	NTe	Ne		NVP	NFN	NFP	VVP	VFN	VFP		L	?	F
MIT_100	2258	2256	0		2255	0	0	1	0	0		0	0	0
MIT_101	1850	1821	3		1819	0	0	0	0	0		0	2	0
MIT_103	2070	2058	0		2047	0	0	0	0	0		0	11	0
MIT_105	2554	2333	53		2240	10	15	2	14	12		15	52	0
MIT_106	2015	1972	27		1463	0	22	377	20	0		10	102	0
MIT_108	1748	1569	9		1124	2	3	11	1	2		245	186	2
MIT_109	2518	2512	1		2460	0	1	16	1	0		15	20	2
MIT_111	2109	2096	0		1832	0	0	0	0	0		251	13	0
MIT_112	2525	2508	0		2507	0	0	0	0	0		0	1	0
MIT_113	1778	1734	4		1728	4	0	0	0	4		0	2	0
MIT_114	1864	1852	0		1808	0	0	40	0	0		0	4	4
MIT_115	1939	1924	1		1923	0	1	0	0	0		0	1	0
MIT_116	2399	2372	0		2263	0	0	95	0	0		0	14	0
MIT_117	1521	1510	0		1510	0	0	0	0	0		0	0	0
MIT_118	2263	2260	15		1244	0	15	0	15	0		1000	1	0
MIT_119	1974	1970	0		1526	0	0	440	0	0		0	4	0
MIT_121	1849	1808	0		1805	0	0	1	0	0		0	2	0
MIT_122	2462	2456	0		2456	0	0	0	0	0		0	0	0
MIT_123	1504	1502	3		1498	0	3	0	3	0		1	0	0
MIT_124	1605	1548	4		1498	0	4	41	4	0		0	5	5
MIT_200	2581	2510	49		1705	3	41	568	38	4		7	189	1
MIT_201	1928	1908	22		1635	22	0	144	0	22		0	107	2
MIT_202	2122	2117	6		2071	4	2	18	1	4		0	23	1
MIT_203	2957	2460	217		1788	102	49	242	38	106		24	266	1
MIT_205	2642	2590	3		2545	0	3	35	3	0		0	7	11
MIT_207	1844	1645	439		1226	2	37	38	34	29		271	74	0

MIT_208	2918	2453	12		1468	1	3	458	3	2	3	520	370
MIT_209	2986	2945	2		2940	0	1	1	0	0	0	4	0
MIT_210	2629	2535	24		2349	11	10	142	10	12	7	16	9
MIT_212	2734	2668	0		2268	0	0	0	0	0	397	3	0
MIT_213	3236	2851	60		2601	0	60	53	60	0	1	136	362
MIT_214	2247	2223	5		1917	0	5	195	4	0	65	42	1
MIT_215	3349	3329	2		3173	0	2	111	2	0	0	43	1
MIT_219	2140	2038	54		1968	0	54	7	54	0	4	5	1
MIT_220	2034	2032	0		2032	0	0	0	0	0	0	0	0
MIT_221	2404	2392	11		1997	0	3	388	0	0	2	5	0
MIT_222	2468	2125	8		2101	0	0	0	0	1	0	24	0
MIT_223	2591	2544	77		1847	4	73	33	73	4	82	505	14
MIT_228	2038	2033	5		1674	0	5	179	5	0	0	175	0
MIT_230	2242	2240	0		2028	0	0	0	0	0	161	51	0
MIT_231	1554	1552	0		1546	0	0	2	0	0	4	0	0
MIT_232	1729	1722	46		1164	3	34	0	0	5	382	173	0
MIT_233	3064	2995	12		2191	0	12	735	12	0	3	54	11
MIT_234	2738	2666	1		2662	1	0	3	0	1	0	0	0
TOTAL:	**99980**	**96634**	**1175**		**85902**	**169**	**458**	**4376**	**395**	**208**	**2950**	**2842**	**798**

Tableau 2 : Résultats de l'étiquetage N/V sur la base MIT (les enregistrements 102, 104, 107 et 217 sont des enregistrements avec stimulateur cardiaque ; l'analyse de ce type de tracé n'étant pas l'objet de la présente étude, ils ont été retirés des résultats)

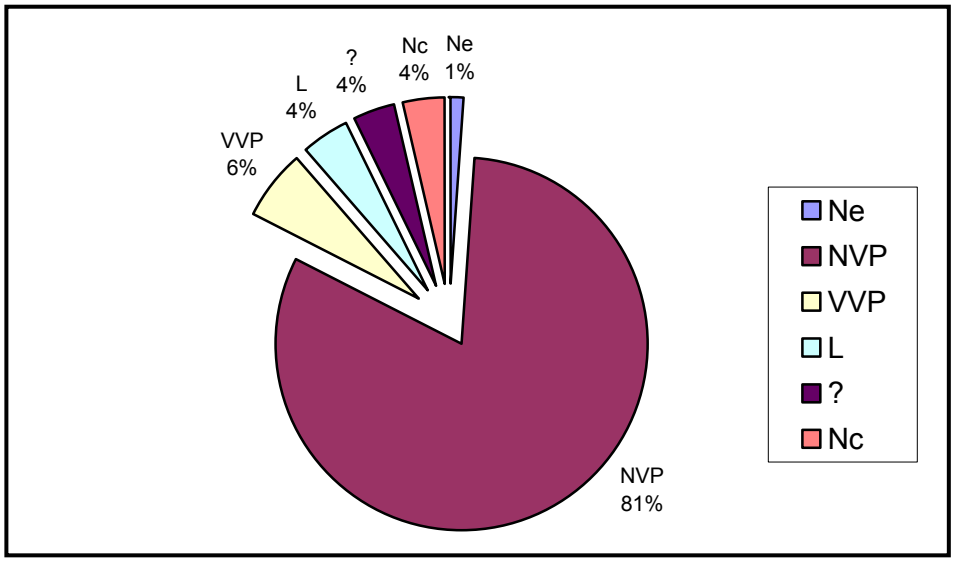


Figure 6 : Synthèse des résultats sur la base AHA pour cent battements. Le taux de réussite pour les labels N et V est de 87%, 4% des battements sont étiquetés avec le label L, 4% avec le label ?, 4% ne sont pas étiquetés et 1% sont mal étiquetés.

NOM	NT	NTe	Ne	NVP	NFN	NFP	VVP	VFN	VFP	L	?	F
1001	1609	1607	0	1607	0	0	0	0	0	0	0	0
1002	2583	2562	0	1933	0	0	0	0	0	550	79	0
1003	2170	2165	2	2165	0	0	0	0	0	0	0	0
1004	2961	2913	0	2900	0	0	0	0	0	13	0	0
1005	2541	2539	115	2276	115	0	0	0	115	98	50	0
1006	2110	2108	0	1862	0	0	0	0	0	246	0	0
1007	1523	1521	0	1521	0	0	0	0	0	0	0	0
1008	2435	2430	0	2397	0	0	0	0	0	33	0	0
1009	3568	3268	1	2356	0	0	0	0	0	792	120	0
1010	1983	1890	22	1416	2	1	0	0	2	421	51	0
2001	2863	2794	0	2516	0	0	70	0	0	0	208	0
2002	2232	312	9	267	4	5	14	5	4	7	15	419
2003	2401	2249	0	2226	0	0	8	0	0	12	3	0
2004	3499	2973	12	2574	0	11	16	11	0	365	7	0
2005	1607	1595	7	1532	0	1	28	0	0	0	35	0
2006	1600	1598	0	1330	0	0	213	0	0	0	55	0
2007	3275	3164	34	2703	30	4	270	4	30	44	113	22
2008	2841	1787	1	1654	1	0	43	0	1	0	89	2
2009	2403	2366	2	1679	0	2	138	2	0	521	26	0
2010	2524	2455	0	2074	0	0	79	0	0	72	230	0
3001	2163	2118	8	2073	0	6	13	6	0	15	11	17
3002	2930	2928	0	2863	0	0	34	0	0	0	31	0
3003	1938	1888	8	1850	3	5	17	0	3	0	18	0
3004	1865	1848	2	1513	0	2	52	2	0	53	228	0
3005	1771	1768	1	1390	0	1	11	1	0	17	349	0
3006	3233	3219	3	2975	0	3	105	3	0	2	134	2
3007	2311	2309	8	2283	0	8	18	8	0	0	0	0
3008	2410	2408	92	2247	0	92	8	92	0	0	61	0
3009	2570	2565	38	2464	4	34	25	34	4	11	27	0
3010	2455	2357	2	2217	1	1	55	1	1	61	22	0
4001	1919	1910	0	1473	0	0	407	0	0	0	30	0
4002	2364	2341	17	2227	0	16	81	16	0	0	17	1
4003	2569	2567	320	2095	0	320	152	320	0	0	0	0
4004	2245	2241	4	2131	0	3	9	3	0	1	97	1
4005	1428	1423	0	1287	0	0	131	0	0	0	5	0
4006	1886	1868	11	1566	0	11	4	11	0	195	92	16
4007	3507	3445	186	2492	0	185	71	185	0	0	697	50
4008	1865	1863	2	1838	0	0	22	0	1	0	3	0
4009	2359	2283	0	1478	0	0	792	0	0	1	12	0
4010	2894	2878	184	2155	43	141	505	141	43	0	34	12
5001	2164	2079	95	1881	4	91	9	91	4	1	93	0
5002	2333	2325	0	2162	0	0	157	0	0	1	5	0
5003	2360	2246	1	2238	0	1	0	1	0	5	2	0
5004	2275	2251	0	1433	0	0	355	0	0	6	457	0
5005	1793	1791	8	1478	0	8	293	8	0	0	12	0
5006	2055	1422	3	1350	0	0	25	0	0	30	17	0
5007	2908	2819	0	2775	0	0	44	0	0	0	0	0
5008	1833	1811	0	1686	0	0	34	0	0	5	86	0
5009	2151	2149	0	2138	0	0	11	0	0	0	0	0
5010	2004	1976	53	442	45	8	340	8	45	1014	127	0

6001	2486	1462	6		1429	1	5	25	5	1		0	2	0
6002	1923	1920	1		1681	1	0	217	0	1		0	21	1
6003	2686	2673	5		2440	0	5	101	5	0		0	127	1
6004	2238	2232	23		2078	19	4	34	4	19		0	97	0
6005	2248	2216	4		2044	4	0	148	0	4		0	20	0
6006	2770	2673	4		2306	0	0	304	0	1		3	60	51
6007	2032	2028	0		1566	0	0	448	0	0		0	14	1
6008	2346	2340	0		2188	0	0	49	0	0		0	103	0
6009	2487	2461	25		1643	0	25	187	25	0		12	594	18
6010	3276	3184	3		2664	2	1	321	1	2		71	125	21
7001	3150	3103	8		1792	0	8	610	8	0		236	457	9
7002	2115	2113	134		1116	133	1	209	1	133		469	185	0
7003	2519	2510	8		1845	6	2	165	2	6		491	1	6
7004	1913	1889	0		1489	0	0	20	0	0		380	0	0
7005	2411	2285	3		2073	0	3	106	3	0		94	9	0
7006	3095	3093	4		1217	0	4	1853	4	0		2	17	0
7007	2327	2316	6		2217	5	1	82	1	5		5	6	2
7008	1548	1544	0		1507	0	0	36	0	0		0	1	2
7009	2883	2379	382		1043	0	382	439	382	0		36	479	1
TOTAL:	**163739**	**155813**	**1867**		**131526**	**423**	**1401**	**10013**	**1394**	**425**		**6391**	**6066**	**655**

Tableau 3 : Résultats sur la base AHA. La série des enregistrements 800x n'a pas été analysée car il s'agit uniquement de fibrillations ventriculaires pour lesquelles le tracé ne présente aucune forme précise.

Le plus important taux d'erreur sur la base AHA est réalisé pour l'enregistrement 4003 (NFP = 320). Il s'agit ici d'une famille qui n'a été analysée que sur une voie, la distinction N/V ne se faisant clairement que sur l'autre (identique au cas de l'enregistrement MIT_223 Figure 3 Annexe D).

Nous comparons ici les résultats obtenus par notre algorithme avec ceux trouvés dans la littérature.

I Détection des QRS

La sensibilité sur la base MIT de notre algorithme pour la détection des QRS est (cf. Annexe A pour le détail) [I]:

$Se = 99{,}85\%$ et $+P = 99{,}94\%$

[Pan, 1985], article de référence pour la détection des QRS, propose les résultats suivants :

$Se = 99{,}76\%$ et $+P = 99{,}56\%$

La solution que nous avons proposée ici est largement inspirée de celle proposée par les auteurs, seule la dernière partie de l'algorithme a été optimisée (Chapitre 1.I.1.6 Seuillage adaptatif). En revanche la nouveauté par rapport à cet article est l'analyse multipiste qui n'est pas effectuée par Pan et Tompkins, et le repérage automatique des zones bruitées (Chapitre 4.II Estimation du bruit).

[Bahoura, 1997] propose également un algorithme de détection des QRS à partir d'ondelettes. Les résultats sur la base MIT sont les suivants :

$Se = 99{,}88\%$ et $+P = 99{,}85\%$

La sensibilité de l'algorithme est légèrement meilleure, la prédiction positive légèrement moins bonne. Notre algorithme et celui présenté dans cet article obtiennent donc des résultats très voisins. Cependant, notre solution permet indifféremment l'analyse sur 2 ou 3 voies, qui

[I] Pour rappel des équations 1 et 2 en introduction, $Se = \frac{VP}{VP + FN}.100$ est la sensibilité ; et $+P = \frac{VP}{VP + FP}.100$ la prédiction positive, où VP est le nombre de vrais positifs, FP le nombre de faux positifs, et FN le nombre de faux négatifs.

n'est pas proposée par [Bahoura, 1997]. L'intérêt principal est la prise de décision très efficace sur trois voies (cf. Chapitre 3.II Analyse multipiste).

Les résultats sur la base MIT de l'algorithme Synescope de Ela Medical sont proches des nôtres.

Se = 99,56% et $+P$ = 99,86%

Leur analyse reste cependant limitée à la prise en de 2 voies d'enregistrement au maximum.

II Étiquetage N / V

Peu d'articles récents proposent des résultats pour l'étiquetage de l'origine des battements (N / V). Nous avons donc comparé nos résultats avec un article de 1990 [Coast, 1990] et le programme SynScope de Ela Medical.

Notre algorithme obtient les résultats suivants (cf. Annexe F pour le détail):

- Base MIT, battements d'origine sinusale (N):

  Se = 99,80% et $+P$ = 99,46%

- Base MIT, battements d'origine ventriculaire (V) :

  Se = 91,70% et $+P$ = 95,46%

- Base AHA, battements d'origine sinusale (N):

  Se = 99,68% et $+P$ = 98.95%

- Base AHA, battements d'origine ventriculaire (V) :

  Se = 87.78% et $+P$ = 95.93%

Les résultats proposés par [Coast, 1990] sur une partie de la base AHA (6 enregistrements) sont :

- battements d'origine sinusale (N):

  Se = 99,82% et $+P$ = 98,90%

- battements d'origine ventriculaire (*V*) :

$Se = 97.25\%$ et $+P = 85,67\%$

Calculés sur ce même sous-ensemble d'enregistrements, nos résultats sont :

- battements d'origine sinusale (*N*):

$Se = 98,89\%$ et $+P = 99,86\%$

- battements d'origine ventriculaire (*V*) :

$Se = 97,62\%$ et $+P = 83,90\%$

Les résultats sont quasiment identiques. La sensibilité pour les battements normaux de l'algorithme proposé par [Coast, 1990] est supérieure à la nôtre ; en revanche, notre prédiction positive est meilleure pour ces mêmes battements, et inversement pour les battements d'origine ventriculaire.

Cependant, le point faible de l'algorithme proposé par [Coast, 1990] est qu'il nécessite un regard expert à chaque enregistrement pour réaliser un apprentissage, c'est-à-dire que pour chaque enregistrement à analyser, il faut au préalable demander à un expert de sélectionner trois battements normaux et trois battements ventriculaires pour effectuer de manière ad hoc l'adaptation des paramètres, ce qui n'est pas la méthodologie envisagée dans notre programme.

L'algorithme Synescope de ELA Medical correspond, quant à lui, à la même méthodologie que la nôtre. Les résultats sur la base MIT sont les suivants :

- battements d'origine sinusale (*N*):

$Se = 99,01\%$ et $+P = 98,10\%$

- battements d'origine ventriculaire (*V*) :

$Se = 76,47\%$ et $+P = 82,7\%$

Ces résultats sont donc inférieurs au nôtre sur cette base. Le traitement que nous réalisons est en effet plus complexe que celui effectué par ce logiciel, il est également plus long.

L'absence de base de référence concernant la localisation des ondes P et T ne nous a pas permis de faire des comparaisons pour cette partie de l'algorithme, la qualité ne peut donc être effectué que par l'analyse des résultats par un expert.

Journal of Machine Learning Research 3 (2003) 1399-1414 Submitted 5/02; Published 3/03

Ranking a Random Feature for Variable and Feature Selection

Hervé Stoppiglia HERVE_STOPPIGLIA@PECHINEY.COM
Informatique Caisse des Dépôts et Consignations
113, rue Jean Marin Naudin
F – 92220 Bagneux, France

Gérard Dreyfus GERARD.DREYFUS@ESPCI.FR
Rémi Dubois REMI.DUBOIS@ESPCI.FR
Yacine Oussar YACINE.OUSSAR@ESPCI.FR
ESPCI, Laboratoire d'Électronique
10, rue Vauquelin
F – 75005 Paris, France

Editors: Isabelle Guyon and Andrè Elisseeff

Abstract

We describe a feature selection method that can be applied directly to models that are linear with respect to their parameters, and indirectly to others. It is independent of the target machine. It is closely related to classical statistical hypothesis tests, but it is more intuitive, hence more suitable for use by engineers who are not statistics experts. Furthermore, some assumptions of classical tests are relaxed. The method has been used successfully in a number of applications that are briefly described.

Keywords: Model selection, variable selection, feature selection, kernel, classification, neural networks, leave-one-out, Gram-Schmidt orthogonalization, statistical tests, information filtering

1. Introduction

The present paper addresses (i) the problem of variable selection for polynomials, and (ii) the problem of selecting explicitly computed kernels such as radial basis functions or wavelets. It is thus essentially a filter method, although it can be used indirectly for selecting a learning machine, e.g. for selecting the inputs and the hidden neurons of neural networks.

Assume that a database is available, including measurements of a set of candidate variables, from which a set of features are computed (for linear machines, the variables are identical to the features). The latter can be ranked in order of decreasing relevance to the output; only the most relevant features, i.e., the top features of the list, should be selected; the question that we address here is that of setting the boundary between the "top" and the "bottom" features, i.e., those which should be selected and those which should be discarded, given the available experimental data.

The following intuitive method, whose close relation to statistical tests will be proved in Section 4, is discussed in the present paper: append to the set of candidate features a "probe" feature, which is a random variable; if the amount of available data were infinite, this feature should be ranked last, or should be ranked as low as other irrelevant features, if any. Since the amount of available data is finite, the probe feature will appear somewhere in the ranked feature list; all features that

are ranked below the probe should be discarded. Actually, since the probe is a random variable, its rank in the list is a random variable too. Therefore, the decision of keeping or discarding a given feature is based on the probability that this feature be ranked higher or lower than the probe. In the spirit of classical hypothesis tests, the designer of the model will choose a risk of selecting a feature although it is less relevant than a random one (or a risk of discarding a feature although it is more relevant than a random one), and will base its decision on that risk.

The first part of the paper recalls the Gram-Schmidt orthogonalization procedure, whereby the candidate features are ranked in order of decreasing relevance to the measured process output, or concept. Section 3 describes the use of the probe feature, the computation of the probability distribution function of its rank in the list, and its use for feature selection. The relation between the present procedure and Fisher's test is subsequently derived. In Section 4, the extension to models that are nonlinear with respect to their parameters is described. Section 5 discusses several applications of the method, both academic and industrial. Finally, we discuss the limitations of the method and show that it is potentially useful in a larger framework.

2. Feature Ranking

A general, lucid discussion of the feature ranking and feature selection problems can be found in the paper of Guyon et al. (2002). The present section is devoted to recalling briefly the use of the Gram-Schmidt orthogonalization procedure for ranking the variables of a model that is linear with respect to its parameters; it was first described by Chen et al. (1989); it was first used in the machine learning context for RBF networks by Chen et al. (1991), for neural networks by Urbani et al. (1993), and for wavelet networks by Oussar (1998), Oussar and Dreyfus (2000); variants of the method were developed recently under the name of Matching Pursuit (see for instance Vincent and Bengio, 2001).

We consider a model with Q candidate features; a data set containing N input-output pairs (measurements of the output of the process to be modelled, and of the candidate features - or of the candidate variables for linear models) is available. We denote by $\boldsymbol{x}^i = [x_1^i, x_2^i, ..., x_N^i]^T$ the vector of values of feature i, or of input i. We denote by $\boldsymbol{y}_p$ the N-vector of the measured values of the output of the process to be modelled. We consider the (N, Q) matrix $X = [\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^Q]$. The model can be written as $\boldsymbol{y} = X\theta$, where θ is the vector of the parameters of the model.

The first iteration of the procedure consists in finding the feature vector that best explains the concept, i.e., which has the smallest angle with the process output vector in the N-dimensional space of observations. To this end, the following quantities are computed

$$\cos^2\left(\boldsymbol{x}^k, \boldsymbol{y}_p\right) = \frac{\left(\boldsymbol{x}^k . \boldsymbol{y}_p\right)^2}{\left\|\boldsymbol{x}^k\right\|^2 \left\|\boldsymbol{y}_p\right\|^2}, \quad k = 1 \text{ to } Q \tag{1}$$

and the vector $\boldsymbol{x}^k$ for which this quantity is largest is selected. In order to discard the part of the concept that is explained by the first selected vector, all remaining candidate inputs, and the output vector, are projected onto the null subspace (of dimension N-1) of the selected feature. In that subspace, the projected input vector that best explains the projected output is selected, and the Q-2 remaining feature vectors are projected onto the null subspace of the first two ranked vectors. The procedure (termed "classical Gram-Schmidt" algorithm) terminates when all Q input vectors are

ranked, or when a stopping criterion is met; the main point of the present paper is the description of a new stopping criterion for that procedure.

In addition, the procedure computes the parameters of the model that are optimum in the least squares sense, so that a model is built while the feature selection procedure is performed. It provides valuable information:

- if the resulting linear-in-its-parameters model does not perform well, the validity of the result of the selection procedure should be questioned; this point will be developed below;
- if the resulting linear-in-its-parameters model performs well, the selected features, or variables derived from the selected features, may be fed to a nonlinear-in-its-parameters model, which may be more parsimonious, hence provide better generalization.

The performance of the linear-in-its-parameters model can be assessed efficiently by making use of the analytic expression of the leave-one-out error computed by using the Sherman-Morrisson-Woodbury theorem (Myers, 1990), which was extended to nonlinear-in-their-parameters models by Monari (1999), Monari and Dreyfus (2000, 2002).

For improved numerical stability, it is recommended to use a slightly different procedure, termed "modified Gram-Schmidt" (Bjoerck, 1967). Full algorithmic descriptions of both the classical Gram-Schmidt and the modified Gram-Schmidt algorithms are available in the paper of Chen et al. (1989).

Given a set of Q candidate features, there are 2^Q possible models. The above procedure allows us to consider only Q models for selection: the model with the feature ranked first, the model with the first two features, etc. The price paid for that complexity reduction is the fact that there is no guarantee that the best model is among the Q models generated by the procedure. It can be shown that the procedure is almost optimal (de Lagarde, 1983).

3. Feature Selection

The main point of the present paper is the presentation of a stopping criterion, which exempts the model designer from ranking all parameters.

Assume that a "probe" feature, which is simply a realization of a random variable, is ranked, just as all other candidate features, by the procedure described in the previous section. It would be natural to discard all features that are ranked below the realization of the probe. However, the rank of the probe feature is actually a random variable, whose cumulative distribution function can be computed exactly as shown below. Once the cumulative distribution function is available, one has to choose an acceptable value of the risk that a random variable might explain the concept more efficiently than one of the selected features, i.e., the risk that a feature might be kept although, given the available data, it might be less relevant than the probe.

Therefore, at each step of the Gram-Schmidt orthogonalization, the procedure is the following:

- after orthogonalization (by classical of modified Gram-Schmidt), pick the projected candidate feature (not selected at previous steps) that has the smallest angle with the projected output,
- compute the value of the cumulative distribution function as described in the next section,
- if that value is smaller than the risk, keep the feature and perform the next step of Gram-Schmidt orthogonalization

- if that value is larger than the risk, discard the feature under consideration and terminate the procedure.

The choice of the risk is problem-dependent: if data is sparse, the model should be as parsimonious as possible, hence a low value of the risk should be chosen, in order to make sure that only relevant inputs are present (but some features with low relevance might be missed); conversely, if data is abundant, a higher risk may be acceptable (but some irrelevant features might be kept).

3.1 Computation of the Cumulative Distribution Function of the Rank of the Probe

We proceed to prove that the cumulative distribution function of the squared cosine of the angle between a given vector and a random vector can be computed exactly, and that the cumulative distribution function of the rank can be derived from that result.

The first step is the computation of the probability distribution function of the squared cosine of the angle φ between a fixed vector and a vector whose components are normally distributed, in a space of dimension ν. It can be expressed as:

$$f_\nu(x) = \frac{\Gamma\left(\frac{\nu}{2}\right)}{\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{\nu-1}{2}\right)} \frac{(1-x)^{\frac{\nu-3}{2}}}{\sqrt{x}} \tag{2}$$

where $\Gamma(.)$ is the gamma function, with $x = \cos^2\varphi$, $\nu \geq 2$ and $0 \leq x \leq 1$. $f_\nu(x)$ is a beta-function with $a = 1/2$ and $b = (\nu\text{-}1)/2$ (see for instance Mood et al., 1974).

The cumulative distribution function $F_\nu(\cos^2\varphi)$ is obtained by integration of relation (2). It can be computed exactly as indicated in Appendix A. From the cumulative distribution function, the probability that the angle between a probe and a fixed vector be smaller than a given angle φ is easily derived as

$$P_\nu(cos^2\varphi) = 1 - F_\nu(cos^2\varphi) \tag{3}$$

for $\nu \geq 2$.

Finally, the cumulative distribution function of the rank of a probe can be derived as follows. At iteration n, n candidate features have been ranked, and a new feature is chosen among the $Q-n$ remaining ones. We denote by φ_n the angle (in a space of dimension $\nu = N-n$) between the selected projected feature and the projected output, and by Π_n the probability that the angle between a realization of the probe and the projected output be smaller than φ_n: $\Pi_n = P_{N-n}(cos^2\varphi_n)$. We denote by G_{n-1} the probability that a realization of the probe be more relevant than one of the n-1 candidate features selected at the n-1 previous steps of the Gram-Schmidt procedure. The probability that a realization of the probe be less relevant than one of the n-1 previous features is equal to 1 - G_{n-1}. Therefore, the probability that a realization of the probe be more relevant than the n-1 previous features but less relevant than the n-th feature is equal to

$$P_{N-n}(cos^2\varphi_n)(1-G_{n-1})$$

Hence, the probability that a realization of the probe be more significant than one of the n features selected after iteration n is given by

$$G_n = G_{n-1} + P_{N-n}(\cos^2\varphi_n)(1-G_{n-1}) \tag{4}$$

with $G_0 = 0$.

As a first illustration, we consider (de Lagarde, 1983) a data set of 15 observations generated by the following simulated process:

$$\boldsymbol{y}_p = X\theta + \omega \tag{5}$$

where $\boldsymbol{y}_p$ and ω are 15-dimensional vectors, θ is a 10-dimensional vector, X is a (15, 10) matrix. The data generating process has actually 5 relevant features ($\{x^1$ to $x^5\}$) only, chosen from a normal distribution: $\theta_i \neq 0$ for i = 1 to 5, $\theta_i = 0$ for i = 6 to 10. The components of vector ω are Gaussian distributed with zero mean and variance 2 10^{-2}. The input vectors $\boldsymbol{x}^j$ (j = 1 to 10) are also chosen from normal distributions.

Figure 1 shows the computed cumulative distribution function of the rank of a realization of the probe. If a model with 5 features is selected, the probability that a random feature might explain the output better than one of the 5 features chosen is lower than 10%. As expected, the five selected features are the features with non-zero parameters of the data generating process.

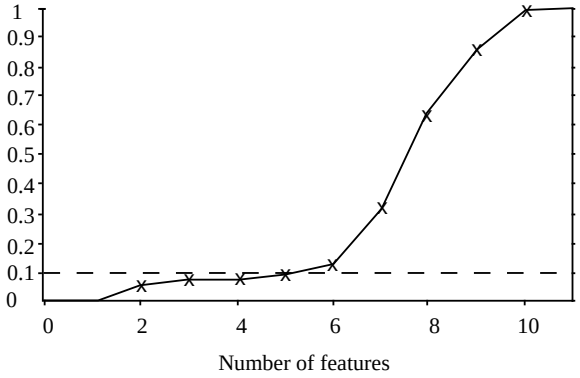


Figure 1: Computed cumulative distribution function of the rank of the probe feature, as a function of the number of selected features. The 5 relevant inputs of the generating procedure are selected if a risk of 10% is chosen.

3.2 Summary

In the present section, we summarize the feature selection procedure for a linear-in-its-parameters model.

First, one should choose a risk r of selecting a feature that is less relevant than a random feature.

At step n of the orthogonalization algorithm ($n < Q$):

- choose the n-th candidate feature in the ranked list,
- compute $\cos^2\varphi_n$ from relation (1), $F_\nu(\cos^2\varphi_n)$ from Appendix A, $P_{N-n}(\cos^2\varphi_n)$ from relation (3), G_n from relation (4),
- if $G_n > r$, select the n-th feature and proceed to step n+1; otherwise, terminate the procedure.

4. Relation to Fisher's Test

Fisher's test is a classical statistical (frequentist) approach to the selection of models that are linear with respect to their parameters. It relies on the assumption that the model is *complete*, i.e., that the regression function belongs to the family of functions within which the model is searched for. If one (or more) input is irrelevant, the corresponding parameter(s) of the model should be equal to zero. Therefore, the hypothesis that is tested is the fact that one or more parameters are equal to zero.

Fisher's test compares a sub-model to the complete model. Other tests, such as the Likelihood Ratio Test (Goodwin and Payne, 1977) and the Logarithm Determinant Ratio Test (Leontaritis and Billings, 1987) compare models that are not thus related. It is proved that these tests are asymptotically equivalent to Fisher's test (Soederstroem, 1977).

In principle, the complete model (with Q parameters) should be compared, using Fisher's test, to all 2^Q sub-models. Using feature ranking with the Gram-Schmidt method as explained above, the number of comparisons can be reduced to Q.

It is shown in Appendix B that the random variable that is used by Fisher's test to discriminate between the null hypothesis and the alternative one can be derived from the probe feature method. The latter thus appears as an alternative to Fisher's test, which (i) gives the model designer a clear explanation as to why features should be discarded (given the available data) and (ii) does not rely on the assumption that the complete model actually contains the regression.

5. Application to the Selection of Models that are Nonlinear with Respect to their Parameters

Since this procedure applies only to models that are linear with respect to their parameters, it is not directly applicable to the selection of the inputs of nonlinear-in-their-parameters models: multilayer perceptrons, radial basis function networks, wavelet networks, etc. This drawback can be circumvented by noting that a variable which is irrelevant is irrelevant irrespective of the model, provided that the latter can learn the task; therefore, the variables can be first selected with a model linear with respect to its parameters (a polynomial model for instance), and subsequently used as inputs to a neural net, thereby taking advantage of the parsimony of the latter. In the next section, we describe an example where the relevant variables in a XOR classification problem are discovered among many irrelevant variables, by selecting the inputs of a polynomial model of degree 2 that solves the problem.

Therefore, the procedure is as follows:

- perform feature selection on a model that is linear with respect to its parameters, e.g. a polynomial; check that the model gives reasonable results on the training set (if there is no point in checking its generalization ability), or assess the generalization performance by computing the leave-one-out error as mentioned above;
- if the linear-in-its-parameters model can learn the task, use the variables that appear in the selected monomials as inputs to a nonlinear-in-its-parameters model;
- if the polynomial model cannot learn the task, increase the degree of the polynomial.

The main limitation of the procedure is the increase of the complexity of the polynomial with the number of features to be ranked. It might become intractable if thousands of features were to be ranked.

Furthermore, the procedure can be applied to the estimation of the number of hidden neurons. Consider a feedforward neural network, with a single layer of hidden neurons and a linear output neuron, whose inputs are known. We address the problem of estimating the minimum number of hidden neurons required to perform a nonlinear regression on the available data. The application of the method described above is straightforward since the output of the hidden layer may be considered as the input of a linear model. The method can be applied either destructively, starting with a large model and using the selection method to discard useless neurons, or constructively, adding neurons until a hidden neuron is considered to be less relevant than a "probe" hidden neuron. In the first case, one trains a large neural network, performs the above procedure by using as candidate features the outputs of the hidden neurons, and discards the neurons that are less relevant than the probe, with the chosen risk. The new architecture is retrained fully, and the procedure is iterated until no irrelevant neuron is detected. In the second case, one starts from a minimal network and increments the number of hidden neurons until an irrelevant neuron is detected.

In the same spirit, the procedure can be used for selecting RBF or wavelet networks: a library of RBF's (resp. wavelets), with fixed centers and widths (resp. translations and dilations) is created, and the procedure is applied to select the most relevant kernels. The surviving kernels are subsequently trained (i.e., their centers or translations, and widths or dilations, are adjusted). If necessary, the process can be iterated for further selection (Chen et al., 1989, Oussar and Dreyfus, 2000).

6. Numerical Illustrations and Applications

This section is devoted to the presentation of academic problems and of several applications in which the method proved successful.

6.1 Academic Problems

Before proceeding to industrial and financial applications, we illustrate the method's use on three problems of academic interest.

6.1.1 VARIABLE SELECTION IN A SYNTHETIC CLASSIFICATION PROBLEM

In http://www.clopinet.com/isabelle/Projects/NIPS2001/#dataset, a database is generated as follows: a linear discriminant function is chosen with random parameters in 2-dimensional space, random examples are generated from that separator, and a given percentage of the outputs are flipped. Additional features are generated randomly, with given percentages of independent features, dependent features, and repeated features. In our experiments, 100 such databases were generated. For each database, 800 examples were generated for training and 400 for testing; 238 additional features were generated. 10 % of the outputs were flipped randomly.

One "true" feature was among the top two ranked features 100 times, both "true" features were selected 74 times; the selected features were used as inputs of linear separators with sigmoid nonlinearity, which were trained by minimizing the usual least squares cost function with the Levenberg-Marquardt algorithm; for comparison, similar linear separators were trained with the "true" features, whenever the latter were not selected. The mean misclassification rate on the training sets was 10.4% (standard deviation 1.1%) with the selected features, whereas it was 10.1% (standard deviation 0.7%) with the "true" features. A t-test performed with 0.5% significance accepts the hypothesis that the difference between the means is smaller than or equal to 0.125 %, which is the

smallest misclassification rate that can be detected (1 error out of 800). Thus, the performances of the true features and of the selected ones are not distinguishable with that level of significance, thereby proving that the probe feature method selects either the "true" features, or features that are essentially as good as the "true" ones for the problem under consideration, with the machine that was implemented. The classification rates on the test sets are not significantly different from those on the training sets (a t-test with 5% significance supports the null hypothesis that the mean misclassification rates are equal). For comparison, the misclassification rate when two features are chosen randomly is about 45%; it is about 30% when one "true" feature and one randomly chosen feature are used.

If a 1% risk is used,[1] the first three features of the list are selected. The resulting misclassification rates are not significantly different from the above results.

If 100 examples only are present in the training set, both "true" features are found in only 37 cases out of 100; however, the misclassification rates of classifiers trained with both true features did not differ significantly by more than 1% from the misclassification rates of classifiers trained with the two selected features.

6.1.2 VARIABLE SELECTION FOR THE XOR PROBLEM

In the same spirit, we build a database for classification with two classes of 50 examples each, drawn from identical Gaussian distributions whose centers are in XOR positions in 2-dimensional space. 50 additional candidate variables are generated from a uniform distribution in [-2, +2]. If feature selection is attempted with a linear model, the above procedure fails to give a satisfactory model, as expected, so that the result of the selection is not valid; the relevant inputs are ranked quite low. If variable selection is attempted with a quadratic model (leading to 1,326 different features, with 52 independent features including 2 relevant features), the random probe procedure selects the relevant variables, and no other, with 1% risk. If the regression is performed with the selected variables, the valid discriminant function $f = x_1 x_2$ is found, where x_1 and x_2 are the relevant variables.

In the present example, there is no point in trying to find a better solution by feeding the selected variables $x_1$and x_2 to a nonlinear-in-its-parameters model: since the problem is 2-dimensional, a neural network would not provide a more parsimonious solution.

6.1.3 SELECTION OF INPUTS AND HIDDEN NEURONS IN A NEURAL NETWORK

A training set of 2,000 examples, and a test set of 2,000 examples, are generated by a neural network with 10 inputs, 5 hidden neurons with sigmoid activation function, and a linear output neuron. Its weights are drawn from a gaussian distribution (0, 0.1). The inputs are drawn from a Gaussian distribution with zero mean, whose standard deviation is computed so as to convey a given variance to the potential of the hidden neurons: the larger the variance of the potential, the more severe the nonlinearity. A zero-mean Gaussian noise is added to the output. Inputs are first selected as described in Section 3, and hidden neurons are selected as described in Section 5. Training is performed with the BFGS optimization algorithm, using gradient values computed by backpropagation. Table 1 shows the results obtained for 2 different standard deviations of the potential of the hidden neurons, and 5 different noise variances ranging from 10^{-10} to 1. In all cases, the selection method, starting from a candidate architecture with 20 candidate inputs and 10 hidden neurons, retrieves the correct

1. Experiments performed with the NeuroOne software package by Netral S.A.

Standard deviation of the potential	Standard deviation of the noise	Number of inputs of the final model	Number of hidden neurons	Root mean square training error	Root mean square test error
3	$1.\ 10^{-10}$	10	5	$9.6\ 10^{-11}$	$1.1\ 10^{-10}$
3	$1.\ 10^{-1}$	10	5	$9.8\ 10^{-2}$	$1.1\ 10^{-1}$
3	1	10	4	1.04	1.1
5	$1.\ 10^{-10}$	10	5	$9.7\ 10^{-11}$	$1.1\ 10^{-10}$
5	$1.\ 10^{-1}$	10	5	$1.0\ 10^{-1}$	$1.1\ 10^{-1}$
5	1	10	4	1.02	1.1

Table 1: Feature selection for the neural network problem, varying noise and number of hidden units.

architecture, except for high noise levels where a lower complexity is appropriate for explaining the measured output.

6.2 Industrial and Financial Applications

In this section, we describe briefly a number of real applications in which this method proved powerful, and was readily understood by field experts who were not familiar with statistical methods such as hypothesis testing.

The prediction of chemical properties of molecules (or QSAR – Quantitative Structure-Activity Relations), viewed as an aid to drug discovery, is a notoriously difficult problem (see for instance Hansch and Leo, 1995), because data is sparse, and because the candidate features are numerous. Both neural networks (see for instance Bodor et al., 1994) and support vector machines (Breneman et al., 2002) have been used extensively. The variable selection method presented here (together with an efficient machine selection method) allowed the prediction of the partition coefficient of a large number of molecules with previously unequalled accuracy on the same data sets (Duprat et al., 1998).

Spot welding is the most widely used welding process in the car industry. Two steel sheets are welded together by passing a current of a few kiloamperes between two electrodes pressed against the metal surfaces, typically for a hundred milliseconds. The heating thus produced melts a roughly cylindrical region of the metal sheets. After cooling, the diameter of the melted zone characterizes the effectiveness of the process; therefore, the spot diameter is a crucial element in the safety of a vehicle. At present, no fast, non-destructive method exists for measuring the spot diameter, so that there is no way of assessing the quality of the weld immediately after welding. Modelling the dynamics of the welding process from first principles is a difficult task, which cannot be performed in real time. These considerations led to considering black-box modelling for designing a "virtual sensor" of the spot diameter from electrical and mechanical measurements performed during welding. The main concerns for the modelling task were the choice of the model inputs, and the limited amount of examples available initially in the database. Variable selection (Monari, 1999) was performed both as described in the present paper, and by more classical methods (stepwise

backward regression and statistical tests based on performance comparisons), with identical results. Our method is computationally less expensive than methods based of performance comparisons since performance comparisons between models with different inputs require (i) training several models with different initial values of the parameters (for nonlinear-in-the-parameters models), (ii) selecting the model with the smallest leave-one-out or cross-validation score for each set of candidate inputs, (iii) performing the test. The variables selected by the random probe method with a polynomial model of degree 3, were subsequently used as inputs to neural networks. The feature set was validated by the process experts. The selection of the prediction machine itself was performed on the basis on the computed leverages of the example, as described by Monari and Dreyfus (2002).

Still in the area of nondestructive testing, but in a completely different application, the feature selection method described here was implemented for the classification of electromagnetic signatures provided by eddy current sensors mounted a few millimeters above the rails, under carriages of the Paris subway. The purpose of the application is the automatic detection of rail defects. Fourier analysis yields 100 candidate features, while the number of examples was limited to 140, for a 4-class problem. The 4-class problem was split into 2-class subproblems, and feature selection was performed independently for each problem; the number of variables was thus reduced to less than 10 for each classifier (Oukhellou et al., 1998).

The present selection method was originally developed for two target applications in finance: the financial analysis of companies for investment purposes, and the financial analysis of town budgets. In the first case, experts suggested 45 financial ratios that were deemed relevant. The probe feature method reduced the number of features to 7, leading to a model that was more efficient and more clearly understandable than the previous ones; it has been in constant use for the last five years. In the second case, the modelling was a 5-class classification problem, which was split into 10 pairwise classification problems; variable selection was performed separately for each classifier. Using a 5% risk, the largest pairwise classifier had 10 variables. The classifier was applied to all 36,000 French towns for financial assessment. Both applications are described in detail by Stoppiglia (1997)

Finally, the present method proved particularly successful for information filtering. The purpose of information filtering is to find information that is relevant to a given topic (defined in a short sentence) in a wide corpus of texts. This can be formalized as a simple 2-class classification problem (a document is either relevant or irrelevant), but the selection of the variables of the classifier (related to the frequency of occurrence of words in the text to be classified) is difficult, since the vocabulary is virtually infinite. Furthermore, since isolated words tend to be ambiguous, the context must be considered, thereby making the structure of the classifier even more complex (see for instance Jing and Tzoukermann, 1999). Therefore, feature selection is crucial. Detailed comparisons between the present method, mutual information, statistical tests, and a selection method that is specific to automatic language processing, can be found in the study of Stricker (2000). The method presented here was used both to find the specific vocabulary of the topics and to find the relevant context of each word of the specific vocabulary. Experiments performed on very large corpuses (Reuters and Agence France-Presse corpuses, and other corpuses mentioned below) and large numbers of topics, showed that the specific vocabulary of a topic can be reduced to 25 words on the average, with an average of 3 context words per word of the specific vocabulary. Linear classifiers trained with regularization were found to be suitable after variable selection. Detailed descriptions of applications of the present selection method to information filtering can be found in the papers of Stricker et al. (1999), Wolinski et al. (2000).

Task	Number of examples	Number of candidate features	Number of selected features
QSAR (regression)	321	74	8
Spot welding (regression)	310	15	4
Eddy current signals (classification, 4 classes)	100	140	< 10
Financial analysis (classification, 5 classes)	250	45	7
Information filtering (classification, 2 classes)	1000 (typical)	400 (typical)	25 (typical)

Table 2: Summary of industrial and financial application results.

Table 2 summarizes results obtained in the above applications. For the last one, typical values are given, because thousands of different classifiers were designed in order to deal with the databases that were investigated.

7. Discussion and Conclusion

The probe feature method, as described in the present paper, contains two distinct ingredients: a method for ranking features (classical or modified Gram-Schmidt orthogonalization) and a method for selecting ranked features (the introduction of a probe feature among candidate features). Although they are presented together here, they deserve separate discussions.

The ranking of features through orthogonalization for linear-in-their-parameters models is by no means new. It has many interesting features. First, it is fast. Second, it takes into account the mutual information between features: if two features are almost collinear in observation space, the fact that one of them is selected will tend to drive the other to a much lower rank in the list. It has the additional advantage of allowing an incremental construction of the model, so that training can be terminated, without using all features, as soon as a satisfaction criterion is met; if the linear-in-the-parameters machine thus trained is expected to be satisfactory, the generalization ability of the machine, as estimated by a cross-validation or leave-one-out score, can be used as a satisfaction criterion. Conversely, if the features are intended for subsequent use as inputs of a different machine, it is only necessary to make sure that the linear-in-its-parameters machine can learn the task; in the affirmative, the selected variables or features thus selected can be used as inputs to a different machine that is not necessarily linear in its parameters. On the negative side, the method is based on the minimization of a squared error loss, which is not always the most appropriate for classification, even though it gives very good results, as shown above; its extension to other loss functions (such as cross-entropy for classification) is an open problem.

The idea of appending a random probe feature to the set of candidate features and ranking it among the others is central in the present paper. It is a powerful stopping criterion for Gram-Schmidt orthogonalization or any of its variants, because the cumulative distribution function of the rank of the probe can be computed analytically as proved above, so that one does not have to actually rank realizations of the probe. However, as shown by Stoppiglia (1997), it can be used in a different way: instead of *computing* the cumulative distribution function of the probe feature

analytically, one can *estimate* it by generating a number of realizations of the probe feature, and ranking them among the others by whatever ranking method is preferred, thereby generating a corresponding number of realizations of the random rank of the probe, and allowing an estimation of its cumulative distribution function. This makes the probe method potentially of more general use, e.g. for selection methods that are based on weight elimination in the spirit of OBD (see for instance Reed, 1993); since weights are related to individual examples in SVM's, the method might also be useful for example selection (Guyon et al., 2002). In addition, the assumption of normality of the probe can be relaxed since it is necessary only for the analytical computation of the cumulative distribution function.

The selection method that is described in the present paper is intuitive and easily understandable, even by engineers who are not familiar with hypothesis testing; this is an attractive feature for researchers who endeavor to make machine learning techniques popular in industry. However, the method is not yet another heuristics for model selection, since it is firmly based on statistics. Furthermore, in contrast to Fisher's test - to which the probe technique is closely related - the assumption that the complete model actually contains the regression is not required. In contrast to the approach described by Weston et al. (2001) the probe feature method does not aim directly at improving the learning machine itself. It can only be conjectured that the withdrawal of irrelevant variables or features will help the machine perform better. The method proved powerful, in several contexts involving a large number of candidate features, and compared favorably, in terms of computation times, with classical tests.

Appendix A. Computation of the Cumulative Distribution Function

The cumulative distribution function is given by:

$$P_\nu(x) = \int_0^x \frac{\Gamma\left(\frac{\nu}{2}\right)}{\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{\nu-1}{2}\right)} \frac{(1-u)^{(\nu-3)/2}}{\sqrt{u}} du$$

with $\nu \geq 2$ and $x = \cos^2\theta$.

If ν is even, then

$$P_\nu(x) = \frac{2}{\pi}\left[\sin^{-1}\sqrt{x} + \sqrt{x(1-x)}\Phi_{\nu/2-2}(x)\right]$$

where $\Phi_{\nu/2-2}$ is a polynomial of degree $\nu/2 - 2$,

$$\Phi_{\nu/2-2}(x) = 1 + \sum_{k=1}^{\nu/2-2} 2^k \frac{k!}{(2k+1)!!}(1-x)^k \quad \text{for} \quad \nu \geq 6$$

$\Phi_0(x) = 1$ for $\nu = 4$,
$\Phi_{-1}(x) = 0$ for $\nu = 2$.

If ν is odd, then

$$P_\nu(x) = \sqrt{x}\Psi_{(\nu-3)/2}(x)$$

where $\Psi_{(\nu-3)/2}$ is a polynomial of degree $(\nu-3)/2$,

$$\Psi_{(\nu-3)/2}(x) = 1 + \sum_{k=1}^{(\nu-3)/2} \frac{1}{2^k}\frac{(2k-1)!!}{k!}(1-x)^k \quad \text{for} \quad \nu \geq 5$$

$\Psi_0(x) = 1$ for $\nu = 3$

Appendix B. Relation of the Probe Feature Method to Fisher's Test

Fisher's test is a classical statistical (frequentist) approach to the selection of models that are linear with respect to their parameters. It is assumed that the process can be described by equation:

$$\boldsymbol{y}_p = X\theta + \omega$$

where ω is Gaussian distributed $(0, \sigma^2)$. Since $E(\omega) = 0$, it is assumed that the regression function belongs to the family of linear equations

$$\boldsymbol{y} = X\theta \tag{6}$$

within which the model is searched for (the model is said to be *complete*).

If one (or more) input is irrelevant, the corresponding parameter of the model should be equal to zero. Therefore, the hypothesis that is tested is the fact that one or more parameters are equal to zero. Assume that it is desired to test the validity of the complete model against that of a sub-model with q parameters equal to zero. The following quantities are defined

$\boldsymbol{y}_Q = X\theta_{LS}$ where θ_{LS} is the parameter vector obtained by least-squares fitting of the complete model (Q parameters) to the available data,

$\boldsymbol{y}_{Q-q} = X\theta^q_{LS}$ where θ^q_{LS} is the parameter vector obtained by least squares fitting of the complete model, under the constraint that q parameters out of $Q$are equal to zero

The considered hypotheses are

H_0: the q parameters are equal to zero,

H_1: the q parameters are not equal to zero.

If H_0 (the null hypothesis) is true, the random variable

$$R = \frac{N-Q-1}{q} \frac{\left\|\boldsymbol{y}_p - \boldsymbol{y}_{Q-q}\right\|^2 - \left\|\boldsymbol{y}_p - \boldsymbol{y}_Q\right\|^2}{\left\|\boldsymbol{y}_p - \boldsymbol{y}_Q\right\|^2} = \frac{N-Q-1}{q} \frac{\left\|\boldsymbol{y}_Q - \boldsymbol{y}_{Q-q}\right\|^2}{\left\|\boldsymbol{y}_p - \boldsymbol{y}_Q\right\|^2} \tag{7}$$

has a Fisher-Snedecor distribution with q and $(N-Q-1)$ degrees of freedom. If, with a given risk, the test leads to rejecting the null hypothesis, the sub-model with q parameters equal to zero is rejected.

Fisher's test compares a sub-model to the complete model. Other tests, such as the Likelihood Ratio Test (Goodwin and Payne, 1977) and the Logarithm Determinant Ratio Test (Leontaritis and Billings, 1987) compare models that are not thus related. It is proved in (Soederstroem, 1977) that these tests are asymptotically equivalent to Fisher's test.

In principle, the complete model (with Q parameters) should be compared, using Fisher's test, to all 2^Q sub-models. Using feature ranking with the Gram-Schmidt method as explained above, the number of comparisons can be reduced to Q.

Relation between the probe feature method and Fisher's test

In the previous section, it was proved that, at iteration n of the procedure, $\cos^2\varphi_n$ obeys a Beta distribution with a = 1/2 and b = ($N-n$-1)/2 (relation 2 with $\nu = N-n$). If a random variable X is distributed with a Beta law, then $\frac{b}{a}\frac{X}{1-X}$ obeys a Fisher law with 2a and 2b degrees of freedom. Therefore, the random variable

$$(N-n-1)\frac{\cos^2\varphi_n}{1-\cos^2\varphi_n} = \frac{N-n-1}{\tan^2\varphi_n}$$

obeys a Fisher law with 1 and $N-n$-1 degrees of freedom.

At iteration n of the procedure, a model with n-1 parameters is available. Assume that we want to perform Fisher's test to compare the n-parameter model obtained by adding the next parameter in the ranked list to the model with n-1 parameters (assuming that the complete model contains the regression). From relation (7), the random variable

$$R = \frac{N-n-1}{1} \frac{\left\|\boldsymbol{y}_p^{n-1} - \boldsymbol{y}_{n-1}\right\|^2 - \left\|\boldsymbol{y}_p^{n-1} - \boldsymbol{y}_n\right\|^2}{\left\|\boldsymbol{y}_p^{n-1} - \boldsymbol{y}_{n-1}\right\|^2} \tag{8}$$

should be a Fisher variable with 1 and $N-n$-1 degrees of freedom, where $\boldsymbol{y}_p^{n-1}$ is the projection of the output considered at iteration n, $\boldsymbol{y}_n$ and $\boldsymbol{y}_{n-1}$ are the outputs of the models with n and n-1 variables respectively. At iteration n, all vectors of interest are in a space of dimension $N-n$+1, and the least-squares solution of the model with n-1 parameters lies in the null space of that space, so that $\boldsymbol{y}_{n-1} = 0$. Moreover, $\boldsymbol{y}_n$ is the projection of $\boldsymbol{y}_p^{n-1}$ onto the direction of the selected feature, so that the angle between those vectors is φ_n.

Therefore,

$$\left\|\boldsymbol{y}_p^{n-1} - \boldsymbol{y}_n\right\|^2 = \left\|\boldsymbol{y}_p^{n-1}\right\|^2 \sin^2 \varphi_n$$

and

$$R = \frac{N-n-1}{\tan^2 \varphi_n}$$

Hence, the random variable that is used to discriminate between the null hypothesis and the alternative one can be derived from the probe feature method. The latter thus appears as an alternative to Fisher's test, which (i) gives the model designer a clear explanation as to why features should be discarded (given the available data) and (ii) does not rely on the assumption that the complete model actually contains the regression.

References

A. Bjoerck. Solving linear least squares problems by gram-schmidt orthogonalization. *Nordisk Tidshrift for Informationsbehadlung*, 7:1–21, 1967.

N. Bodor, M. J. Huang, and A. Harget. Neural network studies. 3. prediction of partition coefficients. *J. Mol. Struct. (Theochem.)*, 309:259–266, 1994.

C. Breneman, K. Bennett, M. Embrechts, S. Cramer, M. Song, and J. Bi. Descriptor generation, selection and model building in quantitative structure-property analysis. In J. Crawse, editor, *Experimental Design for Combinatorial and High Throughput Materials Development*. Wiley (to be publised), 2002.

S. Chen, S.A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50:1873–1896, 1989.

S. Chen, F. Cowan, and P. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2:302–309, 1991.

J. de Lagarde. *Initiation à l'analyse des données*. Dunod, Paris, 1983.

A. Duprat, T. Huynh, and G. Dreyfus. Towards a principled methodology for neural network design and performance evaluation in qsar; application to the prediction of logp. *J. Chem. Inf. Comp. Sci.*, 38:586–594, 1998.

G. C. Goodwin and R.L. Payne. Dynamic system identification: Experiment design and data analysis. *Mathematics in Science and Engineering, Academic Press*, 136, 1977.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

C. Hansch and A. Leo. Exploring qsar, fundamentals and applications in chemistry and biology. *American Chemical Society*, 1995.

H. Jing and E. Tzoukermann. Information retrieval based on context distance and morphology. In *Proceedings of the 22nd Annual International Conference on Research and Development in Information Retrieval (SIGIR '99)*, pages 90–96, 1999.

I. J. Leontaritis and S. A. Billings. Model selection and validation methods for non-linear systems. *International Journal of Control*, 45:311–341, 1987.

G. Monari. *Sélection de modèles non linéaires par leave-one-out. Etude théorique et application au procédé de soudage par points*. PhD thesis, Université Pierre et Marie Curie, 1999.

G. Monari and G. Dreyfus. Withdrawing an example from the training set: an analytic estimation of its effect on a nonlinear parameterised model. *Neurocomputing*, 35:195–201, 2000.

G. Monari and G. Dreyfus. Local overfitting control via leverages. *Neural Computation*, 14(6): 1481–1506, 2002.

A. Mood, F. A. Graybill, and D. C. Boes. *Introduction to the Theory of Statistics*. MacGraw-Hill International, 1974.

R. H. Myers. *Classical and Modern Regression with Applications*. Duxbury Press, 1990.

L. Oukhellou, P. Aknin, H. Stoppiglia, and G. Dreyfus. A new decision criterion for feature selection: Application to the classification of non destructive testing signatures. In *European SIgnal Processing COnference (EUSIPCO'98)*, 1998.

Y. Oussar. *Réseaux d'ondelettes et réseaux de neurones pour la modélisation statique et dynamique de processus*. PhD thesis, Université Pierre et Marie Curie, 1998.

Y. Oussar and G. Dreyfus. Initialization by selection for wavelet network training. *Neurocomputing*, 34:131–143, 2000.

R. Reed. Pruning algorithms – a survey. *IEEE Transactions on Neural Networks*, 4:740–747, 1993.

T. Soederstroem. On model structure testing in system identification. *International Journal of Control*, 26:1–18, 1977.

H. Stoppiglia. *Méthodes Statistiques de Sélection de Modèles Neuronaux ; Applications Financières et Bancaires*. PhD thesis, Université Pierre et Marie Curie, Paris, 1997.

M. Stricker. *Réseaux de neurones pour le traitement automatique du langage : conception et réalisation de filtres d'informations*. PhD thesis, Université Pierre et Marie Curie, Paris, 2000.

M. Stricker, F. Vichot, G. Dreyfus, and F. Wolinski. Two-step feature selection and neural network classification for the trec-8 routing. In *Proceedings of the Eighth Text Retrieval Conference*, 1999.

D. Urbani, P. Roussel-Ragot, L. Personnaz, and G. Dreyfus. The selection of neural models of non-linear dynamical systems by statistical tests. In J. Vlontzos, J.Hwang, and E. Wilson, editors, *Neural Networks for Signal Processing IV*, pages 229–237, 1993.

P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning*, 48:165–187, 2001.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In *Neural Information Processing Systems 14*, 2001.

F. Wolinski, F. Vichot, and M. Stricker. Using learning-based filters to detect rule-based filtering obsolescence. In *Proceedings RIAO'2000*, 2000.

Résumé

L'enregistrement Holter (enregistrement électrocardiographique de 24 heures) est un examen très fréquemment utilisé en cardiologie. Parmi les 100 000 battements enregistrés, seul un petit nombre d'entre eux peut traduire la présence d'une pathologie sous-jacente ; l'analyse automatique est donc indispensable.
Les outils actuels fonctionnent sur le principe d'un système expert, robuste, mais peu adaptatif et essentiellement limité à la détection et la classification des signaux de dépolarisation ventriculaire. Une analyse plus détaillée des signaux cardiaques permet une bien meilleure détection de nombreuses pathologies, en particulier grâce à l'extraction des signaux d'origine auriculaire et des ondes de repolarisation.

Nous proposons dans cette thèse une méthode de décomposition mathématique originale des battements cardiaques sur une base de fonctions appelées « bosses ». Contrairement aux régresseurs classiques utilisés en modélisation (ondelettes, RBF,…), les bosses sont des fonctions prévues pour modéliser chaque onde caractéristique du battement cardiaque (les ondes P, Q, R, S et T).
Chaque battement de l'enregistrement est ainsi décomposé en bosses ; puis les labels médicaux P, Q, R, S et T leur sont attribués par des classifieurs (réseaux de neurones).
Disposant alors de l'emplacement et de la forme des toutes les ondes caractéristiques pour l'ensemble de l'ECG, nous pouvons désormais repérer automatiquement des anomalies comme l'inversion de l'onde P, jusqu'alors non détectées par les algorithmes sur les enregistrements de longues durées.

Cette approche a été testée sur de nombreuses bases de données et a montré toute son efficacité par rapport aux méthodes actuelles dem détection d'anomalies.

Abstract

The Holter recording technique (24-hour electrocardiogram) is an important tool for non-invasive electrocardiology. A Holter record features at least 100,000 heart beats, but a heart anomaly can be expressed by only a few of them. Therefore, a fully automated analysis is desirable as a computer-aided diagnosis tool.

In this thesis, we propose a mathematical decomposition of each individual heart beat using a dedicated family of regressors (called "bumps"). Each bump has five adjustable parameters. Unlike conventional regressors (wavelet, RBF,...), bumps are designed to fit the usual cardiac "waves" as defined by cardiologists (P, Q, R S and T waves) ; the shape and position of these waves are the basis of the experts' diagnostics. Since each wave is fitted by a single bump (possibly two), not only the number of parameters needed to model the relevant information is parsimonious, but the decomposition meets the intelligibility requirements of automated medical diagnostics tools.
Once all the cardiac waves of a heart beat are decomposed into bumps, classifiers (neural nets) assign a "medical" label to each bump. Having the position and the shape of all the characteristic waves, we can automatically detect anomalies such as P wave inversion, whose automatic detection was not feasible previously.

This approach was tested on several international databases, showing the efficiency of our method and it is showed that our approach outperforms existing ones in several respects.