Apprentissage de représentations temps-fréquence adaptées à la classification de signaux par optimisation semi-définie positive

Maxime SANGNIER[1 2], Jérôme GAUTHIER[1], Alain RAKOTOMAMONJY[2]

[1]CEA, LIST
91191 Gif-sur-Yvette CEDEX, France

[2]Université de Rouen, LITIS EA 4108
76800 Saint-Etienne du Rouvray, France

maxime.sangnier@cea.fr, jerome.gauthier@cea.fr, alain.rakoto@univ-rouen.fr

Résumé – Le travail présenté ici considère le problème d'extraction automatique de descripteurs pour la classification linéaire de signaux. Ce problème est formalisé, dans le cadre SVM, de manière variationnelle comme l'apprentissage régularisé d'un banc de filtres et résolu par programmation semi-définie positive. Des expériences sur données simulées et réelles montrent l'intérêt de cette stratégie par rapport à une représentation figée en terme de classification et par rapport à une résolution par descente de gradient en terme de temps de calculs.

Abstract – This paper addresses the problem of automatic feature extraction for signal linear classification. It is formally written, in the SVM framework, as a variational problem of learning a filter bank with a regularization, and solved through positive semidefinite optimization. Experimental results on synthetic and real data bring out the advantages of this strategy regarding both the classification accuracy (compared to a fixed time-frequency transform) and the training time (compared to a gradient descent).

1 Introduction

Depuis les années 1990, la nécessité de trouver des descripteurs discriminants (*i.e.* efficients pour une tâche de reconnaissance de formes) a ouvert la collaboration entre les communautés de Traitement du Signal et d'Apprentissage Automatique. Il existe actuellement une grande variété de descripteurs discriminants, dépendants du domaine d'application : perceptions physiques (sonie, timbre), moments statistiques (matrice de covariance), caractérisation spectrale (transformée de Fourier) et représentations temps-fréquence (décomposition en ondelettes). Cependant, ces descripteurs sont souvent choisis arbitrairement et sans lien apparent avec le classifieur, pénalisant ainsi potentiellement les performances globales de reconnaissance.

En reconnaissance automatique de signaux, les représentations temps-fréquence ont été sujettes à un vif intérêt. En effet, elles semblent particulièrement adaptées à la classification de signaux non-nécessairement stationnaires, du fait de leur capacité à extraire des informations de localisation temporelle et fréquentielle. De telles représentations ont ainsi été apprises avec un critère de discrimination, souvent lié aux méthodes à vastes marges (machines à vecteurs de support – ou SVM –, alignement de noyaux, *etc.*). On peut grossièrement dessiner trois grandes familles de méthodes : l'apprentissage de formes d'ondelettes spécifiques [1–3] (décomposition atomique de type analyse), et plus récemment, de dictionnaires [4–6] (décomposition atomique de type synthèse) et de transformations de la classe de Cohen [7, 8] (distribution d'énergie).

Les nombreux travaux portés sur les décompositions atomiques montrent l'intérêt de celles-ci pour la problématique dont il est question ici, et expliquent le choix des bancs de filtres comme modèle de représentation. Ces derniers ont été intensivement étudiés dans le domaine de la compression et du débruitage [9] et gagnent aujourd'hui à l'être dans celui de la reconnaissance automatique, comme le montre [10].

Dans ce travail, nous formalisons d'abord le problème sous la forme de l'apprentissage conjoint d'un banc de filtres et d'un classifieur linéaire (SVM) et appliquons ensuite une méthode d'optimisation originale par rapport à celles utilisées dans les travaux précédemment cités (méthodes pseudo-exhaustives [1], génétiques [2], gloutonnes [4, 7]). Remarquons que les problèmes variationnels (quelques soient les travaux) étant non-convexes, les stratégies de résolution ont une importance notable. Nous confronterons finalement notre algorithme à une décomposition classique et à une technique d'optimisation usuelle par descente de gradient.

2 Formulation du problème

On modélise la représentation temps-fréquence ϕ par un banc de M filtres à réponses impulsionnelles finies $\{\boldsymbol{h_k}\}_{1\leq k\leq M}$ et de facteurs de décimation $\{N_k\}_{1\leq k\leq M}$. Tout signal $\boldsymbol{x}$ de $\mathbb{R}^n$ est donc représenté dans le plan temps-fréquence par $\phi(\boldsymbol{x}) \overset{\text{def}}{=} \left\{((\boldsymbol{h_k}\star\boldsymbol{x})[N_k l+1])_{0\leq l<q_k}\right\}_{1\leq k\leq M}$, où pour tout k de $\mathbb{N}_M$

(entiers compris entre 1 et M), $q_k \stackrel{\text{def}}{=} \left\lfloor \frac{n}{N_k} \right\rfloor$. On désire classer linéairement ce signal dans le plan temps-fréquence, *i.e.* trouver $\boldsymbol{\beta}$ (assimilé à un vecteur) dans $\prod_{k=1}^{M} \mathbb{R}^{q_k}$ et b dans $\mathbb{R}$ tels que le test $\langle \boldsymbol{\beta} | \phi(\boldsymbol{x}) \rangle + b \lessgtr 0$ indique la classe de $\boldsymbol{x}$. Pour ce faire, on considère un ensemble d'apprentissage $\{(\boldsymbol{x_i}, y_i)\}_{1 \leq i \leq T}$ constitué de signaux $\boldsymbol{x_i}$ étiquetés par y_i (pris dans $\{-1, 1\}$) et on se place dans le cadre d'une machine à vecteurs de support (SVM). On cherche ainsi à minimiser le risque structurel non pas uniquement par rapport au classifieur (problème SVM classique)

$$\begin{array}{l} \underset{\boldsymbol{\beta},b,\boldsymbol{\xi}}{\texttt{minimiser}} \ \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{T} \xi_i \\ \texttt{sc} \ \left\{ \begin{array}{l} \forall i \in \mathbb{N}_T, \ y_i(\langle \boldsymbol{\beta} | \phi(\boldsymbol{x_i}) \rangle + b) \geq 1 - \xi_i \\ \boldsymbol{\xi} \succcurlyeq 0, \end{array} \right. \end{array} \tag{1}$$

mais aussi par rapport à la représentation temps-fréquence ϕ, aboutissant ainsi au problème suivant :

$$\begin{array}{l} \underset{\phi,\boldsymbol{\beta},b,\boldsymbol{\xi}}{\texttt{minimiser}} \ \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{T} \xi_i \\ \texttt{sc} \ \left\{ \begin{array}{l} \forall i \in \mathbb{N}_T, \ y_i(\langle \boldsymbol{\beta} | \phi(\boldsymbol{x_i}) \rangle + b) \geq 1 - \xi_i \\ \boldsymbol{\xi} \succcurlyeq 0 \\ \|\phi\|_2 \leq 1. \end{array} \right. \end{array} \tag{2}$$

Dans la formulation qui précède, la norme 2 sur ϕ correspond à la norme 2 sur les réponses impulsionnelles des filtres. Cette contrainte convexe a pour but d'éviter des instabilités des filtres conduisant à un phénomène de sur-apprentissage.

Pour montrer que le problème (2) admet une solution, considérons le problème suivant :

$$\underset{\substack{\phi \\ \phi \in \mathcal{B}}}{\texttt{minimiser}} \left(J_{\text{SVM}}(\phi) \stackrel{\text{def}}{=} \min_{\substack{\boldsymbol{\beta},b,\boldsymbol{\xi} \\ (\boldsymbol{\beta},\boldsymbol{\xi}) \in \mathcal{C}(\phi,b)}} J(\boldsymbol{\beta},\boldsymbol{\xi}) \right) \tag{3}$$

où $J(\boldsymbol{\beta},\boldsymbol{\xi})$ est la valeur de la fonction objectif de (1) et (2), $\mathcal{C}(\phi,b)$ est l'ensemble des contraintes de (1) et $\mathcal{B}$ représente la boule unité de la norme 2. Sous certaines conditions (que l'objectif de (1) soit strictement convexe, ce qui est toujours possible en pratique) le sous-problème de (3) admet un unique minimum (c'est un problème d'apprentissage SVM) et le théorème 4.1 de [11] assure ainsi que J_{SVM} est une fonction continue. Toute fonction continue étant bornée sur la boule unité et atteignant ses bornes, on en déduit que le problème (3) admet pour solution un certain ϕ^* pour lequel le sous-problème (1) est résolu par un certain $(\boldsymbol{\beta^*}, b^*, \boldsymbol{\xi^*})$. Ainsi,

$$\begin{array}{l} \exists \phi^* \in \mathcal{B}, \ \exists b^* \in \mathbb{R}, \ \exists (\boldsymbol{\beta}^*, \boldsymbol{\xi}^*) \in \mathcal{C}(\phi^*, b^*), \\ \begin{array}{rcl} J(\boldsymbol{\beta}^*, \boldsymbol{\xi}^*) & = & \min\limits_{\substack{\phi \\ \phi \in \mathcal{B}}} \left(\min\limits_{\substack{\boldsymbol{\beta},b,\boldsymbol{\xi} \\ (\boldsymbol{\beta},\boldsymbol{\xi}) \in \mathcal{C}(\phi,b)}} J(\boldsymbol{\beta},\boldsymbol{\xi}) \right) \\ & \leq & \min\limits_{\substack{\boldsymbol{\beta},b,\boldsymbol{\xi} \\ (\boldsymbol{\beta},\boldsymbol{\xi}) \in \mathcal{C}(\phi,b)}} J(\boldsymbol{\beta},\boldsymbol{\xi}), \ \forall \phi \in \mathcal{B} \\ & \leq & J(\boldsymbol{\beta},\boldsymbol{\xi}), \ \forall \phi \in \mathcal{B}, \forall b, \forall (\boldsymbol{\beta},\boldsymbol{\xi}) \in \mathcal{C}(\phi,b). \end{array} \end{array}$$

3 Stratégies d'optimisation

3.1 Relâchement semi-défini

Le problème (2) est un programme quadratique (non-nécessairement convexe) à contraintes quadratiques. Pour s'en persuader, on introduit les variables d'amortissement $\boldsymbol{\delta}$ (de $\mathbb{R}^T$) et τ (de $\mathbb{R}$) permettant de transformer les contraintes d'inégalité 1 et 3 de (2) en contraintes d'égalité puis on regroupe les variables conduisant aux termes quadratiques sous un seul vecteur $\boldsymbol{\gamma} \stackrel{\text{def}}{=} \text{Vec}(\boldsymbol{\beta}, \boldsymbol{h_1}, \ldots, \boldsymbol{h_M})$ (où Vec est l'opérateur de vectorisation) et celles menant aux termes linéaires sous $\boldsymbol{\zeta} \stackrel{\text{def}}{=} \text{Vec}(b, \boldsymbol{\xi}, \boldsymbol{\delta}, \tau)$. En notant ${}^t\cdot$ l'opérateur de transposition et en définissant correctement les matrices $\boldsymbol{A}$, $\boldsymbol{A_i}$ ($i \in \mathbb{N}_T$) et $\boldsymbol{D}$, le problème (2) est strictement équivalent au problème (4).

$$\begin{array}{l} \underset{\boldsymbol{\gamma},\boldsymbol{\zeta}}{\texttt{minimiser}} \ \frac{1}{2} \, {}^t\boldsymbol{\gamma}\boldsymbol{A}\boldsymbol{\gamma} + C\sum_{i=2}^{T+1} \zeta_i \\ \texttt{sc} \ \left\{ \begin{array}{l} \forall i \in \mathbb{N}_T \left\{ \begin{array}{l} {}^t\boldsymbol{\gamma}\boldsymbol{A_i}\boldsymbol{\gamma} + \zeta_{i+1} - \zeta_{T+i+1} = 1 \\ \zeta_{i+1} \geq 0, \ \zeta_{T+i+1} \geq 0 \end{array} \right. \\ {}^t\boldsymbol{\gamma}\boldsymbol{D}\boldsymbol{\gamma} + \zeta_{2(T+1)} = 1 \\ \zeta_{2(T+1)} \geq 0 \end{array} \right. \end{array} \tag{4}$$

En remarquant enfin que ${}^t\boldsymbol{\gamma}\boldsymbol{A}\boldsymbol{\gamma} = \text{Tr}(\boldsymbol{A}\boldsymbol{\gamma}\,{}^t\boldsymbol{\gamma})$ et en posant $\boldsymbol{X}_+ \stackrel{\text{def}}{=} \boldsymbol{\gamma}\,{}^t\boldsymbol{\gamma}$ ($(\cdot)_+$ signifiant que la matrice est semi-définie positive), on obtient l'équivalence entre les problèmes (2) et (5).

$$\begin{array}{l} \underset{\boldsymbol{X}_+,\boldsymbol{\zeta}}{\texttt{minimiser}} \ \frac{1}{2}\,\text{Tr}(\boldsymbol{A}\boldsymbol{X}_+) + C\sum_{i=2}^{T+1} \zeta_i \\ \texttt{sc} \ \left\{ \begin{array}{l} \forall i \in \mathbb{N}_T \left\{ \begin{array}{l} \text{Tr}(\boldsymbol{A_i}\boldsymbol{X}_+) + \zeta_{i+1} - \zeta_{T+i+1} = 1 \\ \zeta_{i+1} \geq 0, \ \zeta_{T+i+1} \geq 0 \end{array} \right. \\ \text{Tr}(\boldsymbol{D}\boldsymbol{X}_+) + \zeta_{2(T+1)} = 1 \\ \zeta_{2(T+1)} \geq 0 \\ \boldsymbol{X}_+ \succcurlyeq 0, \ \text{Rang}(\boldsymbol{X}_+) = 1 \end{array} \right. \end{array} \tag{5}$$

La spécificité du problème (5) réside dans la localisation de la difficulté : la non-convexité du programme (2) a été entièrement reléguée dans la contrainte de rang. En supprimant cette contrainte, on réalise un relâchement semi-défini. En effet, en définissant correctement $\boldsymbol{\delta^*}$, τ^* et $\boldsymbol{\gamma^*}$ à partir de ϕ^*, $\boldsymbol{\beta^*}$, b^* et $\boldsymbol{\xi^*}$, $\boldsymbol{\zeta^*} \stackrel{\text{def}}{=} \text{Vec}(b^*, \boldsymbol{\xi^*}, \boldsymbol{\delta^*}, \tau^*)$ et $\boldsymbol{X}_+^* \stackrel{\text{def}}{=} \boldsymbol{\gamma^*}\,{}^t\boldsymbol{\gamma^*}$ forment une solution admissible de (5). De plus, en notant $S(\boldsymbol{X}_+, \boldsymbol{\zeta})$ l'objectif de (5) et $(\boldsymbol{X}_+^\dagger, \boldsymbol{\zeta}^\dagger)$ un minimiseur, on obtient :

$$S(\boldsymbol{X}_+^\dagger, \boldsymbol{\zeta}^\dagger) \leq S(\boldsymbol{X}_+^*, \boldsymbol{\zeta^*}) = J(\boldsymbol{\beta}^*, \boldsymbol{\xi}^*).$$

Le nouveau problème issu de (5) est un programme semi-défini (*i.e.* linéaire à contraintes linéaires sur le cône des matrices semi-définies positives) dont la résolution peut être donnée par un solveur comme SeDuMi [12]. Remarquons que la formulation de notre problème sous la forme d'un programme semi-défini est envisageable en pratique car les matrices $\boldsymbol{A}$, $\boldsymbol{A_i}$ ($i \in \mathbb{N}_T$) et $\boldsymbol{D}$ sont creuses.

En outre, la résolution du programme semi-défini ne donne pas directement une solution admissible de (2) ; deux manières sont envisageables afin d'en obtenir une [13] : considérer le vecteur propre de $\boldsymbol{X}_+^\dagger$ associé à la plus grande valeur propre ou choisir $\boldsymbol{\gamma}$ comme réalisation d'une variable aléatoire suivant une loi multinormale de moyenne nulle et de matrice de covariance $\boldsymbol{X}_+^\dagger$. Dans les deux cas, il sera nécessaire de projeter ultérieurement le vecteur candidat sur l'ensemble des contraintes.

3.2 Gradient projeté

La résolution du problème (2) par relâchement semi-défini est comparée à la résolution du problème (3) par descente de gradient projeté initialisé aléatoirement. On oppose ainsi deux approches : *optimisation puis génération aléatoire* (relâchement) et *génération aléatoire puis optimisation* (gradient). Dans cette stratégie de résolution, la convergence est assurée par la règle de Armijo. Le sous-problème de (3) est résolu par un solveur SVM tandis que le gradient est calculé de manière analytique grâce au théorème 4.1 de [11] :

$$\forall \phi,\ \nabla J_{\text{SVM}}(\phi) = -\frac{1}{2}\, {}^t\boldsymbol{\alpha}\boldsymbol{Y}_+ \nabla\left(\boldsymbol{K}_+\right)(\phi)\boldsymbol{Y}_+\boldsymbol{\alpha},$$

où $\boldsymbol{K}_+(\phi) \overset{\text{def}}{=} \left(\langle \phi(x_i)|\phi(x_j)\rangle\right)_{1\leq i,j\leq N}$ et $\boldsymbol{\alpha}$ sont respectivement le noyau et le vecteur dual optimal du problème SVM, et $\boldsymbol{Y}_+$ est la matrice diagonale des étiquettes.

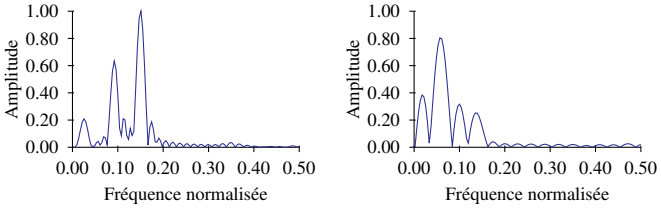
4 Résultats expérimentaux

Les deux stratégies de résolution présentées ci-avant sont comparées entre elles et confrontées à un SVM linéaire dans le domaine temporel et dans le domaine de Fourier sur un problème de classification à deux classes. Deux expériences sont présentées : l'une est fondée sur les signaux *Blocks* et *HeaviSine* (chacun représentant une classe) de la boîte à outils Matlab Wavelab; l'autre est construite sur les enregistrements cardiaques du challenge CHSC [14]. Dans les deux cas, on cherche à apprendre un banc de trois filtres (le facteur de décimation étant fixé à deux) à partir de la base de données bruitées par un bruit gaussien coloré et non stationnaire (*cf.* codes mis en ligne). Le paramètre de coût C est obtenu par validation croisée.

Les résultats sur le jeu de données simulées (figure 3) montrent d'emblée l'intérêt d'une décomposition temps-fréquence dirigée par les données ainsi que l'apport du relâchement semi-défini en terme de classification et de complexité d'apprentissage. Les résultats sur le jeu de données réelles (figure 3) sont en revanche moins probants car le bruit s'ajoute à une forte variabilité intra-classe. Si cette fois, descente de gradient et relâchement semi-défini semblent conduire à des performances de classification comparables, on peut noter que la variabilité des résultats est légèrement plus importante dans le premier cas que dans le second. En outre, les deux autres remarques tirées pour le jeu de données simulées restent valables dans ce cas (intérêt de l'apprentissage et gain en temps).

La figure 1 présente un exemple de banc de filtres appris à partir du jeu de données CHSC (sans ajout de bruit). Les longueurs des filtres ont respectivement été choisies à 64, 32 et 16 de sorte à adopter une approche multi-résolution redondante (facteur de décimation à 2). Sur cet exemple, le filtre le plus court possède une réponse impulsionnelle identiquement nulle. De manière générale, on remarque expérimentalement que les filtres longs sont préférés d'autant que le SNR est faible.

Il est très difficile d'estimer théoriquement la qualité d'un relâchement semi-défini car celui-ci est dépendant de la structure du problème traité. En revanche, on peut analyser expérimentalement les différences moyennes entre les minima locaux obtenus. La figure 2 met en lumière ces différences sur le second jeu de données en faisant figurer les résultats du programme semi-défini (SDP), du relâchement avec tirage aléatoire et d'une descente de gradient initialisée par le précédent résultat. On vérifie tout d'abord sur cette figure que $S(\boldsymbol{X}_+^\dagger, \boldsymbol{\zeta}^\dagger)$ est bien inférieur à tous les autres minima. Il est ensuite intéressant de noter que le relâchement borne supérieurement la descente de gradient, elle même supérieure aux résultats obtenus avec une initialisation adéquate. Pour ce dernier cas, les résultats de classification n'ont pas été décrits par soucis de lisibilité mais restent compris entre ceux des deux stratégies analogues.



(a) Filtre 1 (longueur : 64). (b) Filtre 2 (longueur : 32).

Figure 1: Exemple de banc de filtres appris.

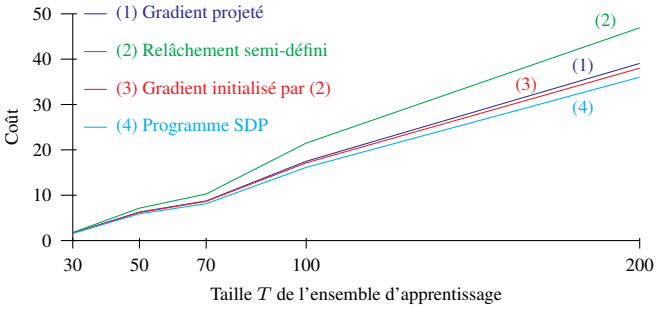
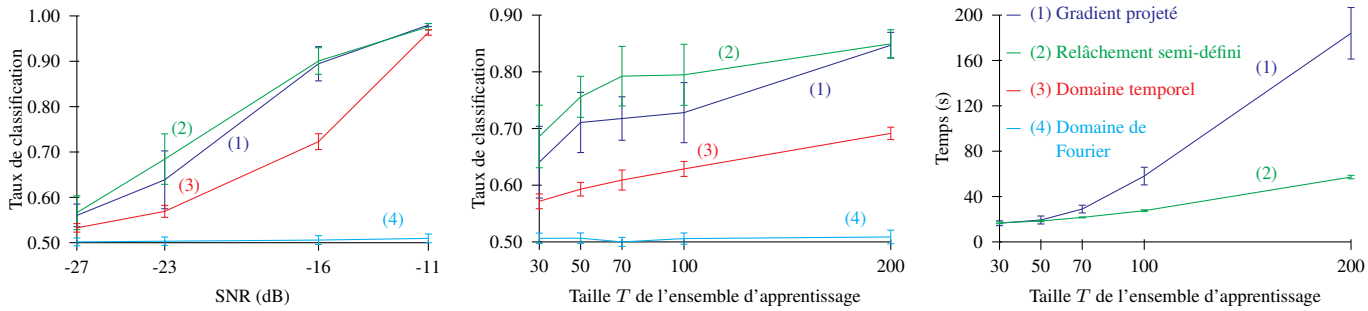


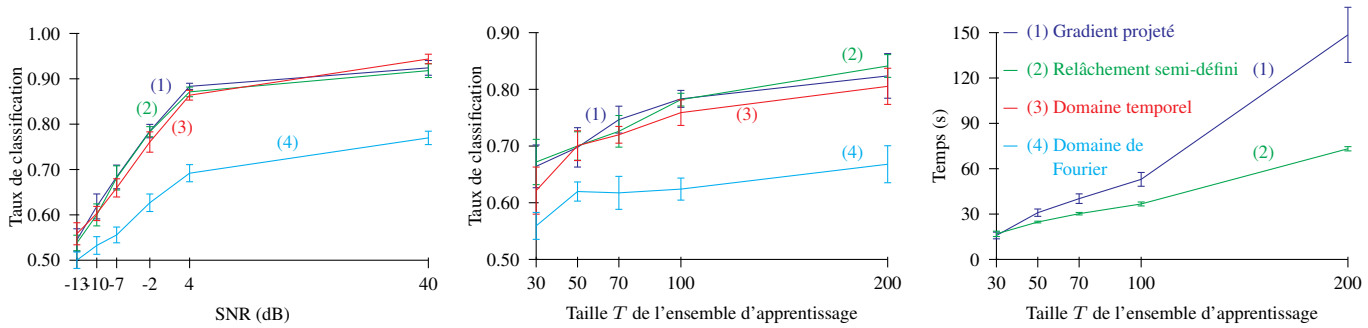
Figure 2: Valeurs optimales moyennes de la fonction de coût.

5 Conclusion

Nous avons présenté, dans cette étude, un cadre particulier qui consiste à apprendre conjointement un classifieur linéaire (SVM) avec une représentation temps-fréquence. En remarquant que le problème se résume à un programme quadratique à contraintes quadratiques, nous avons choisi de le traiter indirectement en résolvant un problème approché qui présente l'avantage d'être linéaire (donc résoluble efficacement et de manière précise). En comparant cette méthode à une transformation classique (Fourier) et à une approche plus commune de descente de gradient, nous avons montré l'intérêt en terme de temps de calcul et de classification de cette approche. L'application sur données réelles relativise toutefois ces résultats et suggère l'introduction de non-linéarités dans le modèle afin de prendre en compte certaines variabilités intra-classes.



(a) Classification sur données simulées. (b) Classification sur données simulées. (c) Temps d'apprentissage sur données simulées.



(d) Classification sur données CHSC. (e) Classification sur données CHSC. (f) Temps d'apprentissage sur données CHSC.

Figure 3: Taux de bonne classification et temps d'apprentissage.

6 Remerciements

Les auteurs remercient la Direction Générale de l'Armement pour le soutien financier apporté à ces recherches ainsi que Dr A. D'Aspremont pour les échanges à propos de l'optimisation semi-définie positive.

References

[1] J. Neumann, C. Schnörr, and G. Steidl. Efficient wavelet adaptation for hybrid wavelet-large margin classifiers. *Pattern Recognition*, 38:1815–1830, 2005.

[2] E. Jones, P. Runkle, N. Dasgupta, L. Couchman, and L. Carin. Genetic algorithm wavelet design for signal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:890–895, 2001.

[3] F. Yger and A. Rakotomamonjy. Wavelet kernel learning. *Pattern Recognition*, 44:2614–2629, 2011.

[4] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[5] K. Huang and S. Aviyente. Sparse representation for signal classification. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[6] F. Rodriguez and G. Sapiro. Sparse representation for image classification: Learning discriminative and reconstructive non-parametric dictionaries. Technical report, University of Minnesota, 2008.

[7] M. Davy, A. Gretton, A. Doucet, and P. J. W. Rayner. Optimized support vector machines for nonstationary signal classification. *IEEE Signal Processing Letters*, 9:442–445, 2002.

[8] P. Honeiné, C. Richard, P. Flandrin, and J.-B. Pothin. Optimal selection of time-frequency representations for signal classification: a kernel-target alignment approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006.

[9] P.P. Vaidyanathan. *Multirate systems and filter banks*. Prentice Hall, 1993.

[10] L. D. Vignolo, H. L. Rufiner, D. H. Milone, and J. Goddard. Evolutionary splines for cepstral filterbank optimization in phoneme classification. *EURASIP Journal on Advances in Signal Processing. Biologically Inspired Signal Processing: Analysis, Algorithms and Applications*, 2011:1–14, 2011.

[11] F. Bonnans and A. Shapiro. Optimization problems with perturbations, a guided tour. *SIAM Journal on Scientific Computing*, 40:228–264, 1996.

[12] J.F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[13] Z.-Q. Luo, W.-K. Ma, A.M.-C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27:20–34, 2010.

[14] P. Bentley, G. Nordehn, M. Coimbra, and Mannor. S. The pascal classifying heart sounds challenge (chsc). http://www.peterjbentley.com/heartchallenge/, 2011.