

# Apprentissage de Mélanges de Gaussiens par Maximisation de la marge avec SMO

Trinh Minh Tri DO<sup>1</sup>, Thierry Artières

LIP6, Université Pierre et Marie Curie Paris, France

**Résumé** : Les modèles de Mélange de lois Gaussiennes (MG) ont été utilisés dans de nombreux domaines, par exemple pour le traitement et la reconnaissance des images ou de la parole, où ils sont traditionnellement appris de façon non discriminante. Récemment des travaux ont porté sur l'apprentissage discriminant de tels modèles, à travers notamment la maximisation de la marge. L'idée de ces travaux consiste à formuler l'apprentissage discriminant de ces modèles comme un problème de maximisation de la marge et de le mettre sous la forme d'un problème d'optimisation convexe, pour lequel des techniques d'optimisation de type descente de gradient projeté peuvent être employées. Nous nous inspirons ici de ces travaux et proposons une nouvelle formulation du problème permettant l'emploi d'un algorithme de type SMO, popularisé pour les Machines à Vecteurs de Support, plus performant et plus rapide que la descente de gradient.

**Mots-clés** : apprentissage discriminant, maximisation de la marge, mélange Gaussien, SMO

## 1 Introduction

Les modèles de Mélange de lois Gaussiennes (MG) ont été utilisés dans de nombreux domaines, par exemple pour le traitement et la reconnaissance des images ou de la parole, afin de construire des classifieurs via l'apprentissage de modèles génératifs. On apprend alors un MG par classe pour modéliser la densité de probabilité conditionnelle, étant donnée la classe  $y$ ,  $p(x|y)$  puis on implémente la règle de décision Bayésienne en supposant, par exemple, que les classes sont a priori équiprobables, donc en identifiant  $\operatorname{argmax}_y p(x|y) * p(y)$ .

Un modèle de mélange Gaussien correspond à une densité de la forme :

$$p(x|y) = \sum_{k=1}^K p(k) \times N(x; \mu_{y_k}, \Sigma_{y_k}) \quad (1)$$

où  $N(x; \mu_{y_k}, \Sigma_{y_k})$  représente la loi gaussienne de moyenne  $\mu_{y_k}$  et de matrice de covariance  $\Sigma_{y_k}$  évaluée en  $x$ , et  $p(k)$  représente la probabilité a priori que  $x$  soit produite par la  $k^{ieme}$  composante du mélange. Les modèles de mélanges Gaussiens doivent une partie de leur popularité d'une part au théorème central limite qui confère à la loi Gaussienne un statut particulier parmi les lois de probabilités paramétriques et d'autre part

à leur généralité. Les mélanges de lois Gaussiennes permettent en effet d'approximer toute densité de probabilité, pourvu qu'elle présente certains caractères de régularité. Egalement, ces modèles se sont avérés plutôt robustes et relativement faciles à employer. Enfin, les lois gaussiennes et les mélanges de lois gaussiennes ont profité de la popularité des modèles Markoviens cachés (MMC), auxquels ils sont traditionnellement attachés, et qui ont été intensivement utilisés depuis une vingtaine d'années dans le cadre du traitement, de la reconnaissance et de la segmentation de séquences, par exemple en reconnaissance de la parole ou de l'écriture manuscrite etc.

Les MG (de même que les MMC) sont traditionnellement appris indépendamment, classe par classe, à l'aide d'un critère de Maximum de Vraisemblance (Dempster *et al.*, 1977; Neal & Hinton, 1998; Afify, 2005). L'optimisation est alors réalisée à l'aide d'un algorithme EM (Expectation - Maximization) qui repose sur une optimisation itérative des paramètres du modèle (moyennes, matrices de covariances, et probabilités a priori des composantes du mélange). L'approche générative consiste à apprendre de façon non discriminante un modèle de densité pour chaque classe, elle est en règle générale moins efficace (du point de vue du taux de classification) qu'une approche purement discriminante. L'approche générative a pourtant été privilégiée depuis longtemps dans le cas de problèmes ouverts ou de données complexes, telles que des données séquentielles, pour lesquelles il est plus délicat de mettre en oeuvre des techniques discriminantes. Ainsi bon nombre de systèmes de reconnaissance de la parole ou du locuteur ont été construits sur des modèles acoustiques appris en modélisation plutôt qu'en discrimination. Nous nous intéressons ici à l'apprentissage discriminant de mélanges Gaussiens, notre but étant d'étendre par la suite ces travaux à l'apprentissage de modèles de séquences de type Markovien.

Divers travaux ont porté sur l'apprentissage discriminant de systèmes qui étaient jusque là appris de façon non discriminante. Ainsi quelques approches discriminantes ont été proposées pour la classification de séquences (plus rarement pour la segmentation). Les premières approches ont consisté à exploiter des critères discriminants tels que le Maximum de Vraisemblance Conditionnelle (Nadas, 1983), le Maximum d'Information Mutuelle (L.R. *et al.*, 1986; Normandin, 1991; Dahmen *et al.*, 1999; Valtchev *et al.*, 1997) ou le Minimum d'Erreur de Classification (Juang & Katagiri, 1992) pour apprendre des modèles génératifs tels que les MG et les MMC. (LeCun *et al.*, 1998) dresse un panorama d'un certain nombre de ces méthodes. (Tong & Koller, 2000) proposent d'apprendre un classifieur discriminant, qui minimise la probabilité d'erreur calculée à l'aide de modèles génératifs. Plus récemment d'autres techniques ont consisté à construire des fonctions discriminantes à partir de modèles génératifs, comme l'utilisation des scores de Fisher (Jaakkola *et al.*, 1999), ou l'exploitation de noyaux entre modèles dans (Moreno *et al.*, 2004).

Enfin, ces dernières années, plusieurs approches ont été proposées pour combiner les modèles Markoviens, exploitant des densités de probabilités de type mélanges de Gaussiennes, et les algorithmes discriminants des machines à vecteurs de supports (Vapnik, 1999; Kruger *et al.*, 2006; Li *et al.*, 2005; Sha & Saul, 2006, 2007). Par exemple, la technique proposée dans (Sha & Saul, 2006, 2007) vise à apprendre des MG (puis des MMC Gaussiens) par maximisation de la marge. Ces travaux sur l'apprentissage de modèles MG par maximisation de la marge sont très prometteurs. Leur application est pourtant

limitée, soit par la nature des modèles appris (e.g. seules les vecteurs moyennes sont appris dans (Li *et al.*, 2005)), soit dans leur efficacité, la convergence de l'algorithme proposé dans (Sha & Saul, 2006, 2007) est par exemple assez lente et sensible à l'initialisation.

Le travail décrit ici est inspiré des travaux de (Sha & Saul, 2006, 2007) et vise le même objectif, l'apprentissage de modèles génératifs basés sur des MG par maximisation de la marge. Nous nous concentrons ici sur l'apprentissage de MG et proposons un algorithme qui diffère en plusieurs points de celui proposé par (Sha & Saul, 2006, 2007). Le coeur de notre travail tient dans la façon dont nous avons traité les contraintes convexes (caractère semi-défini positif des matrices de covariance) et dans la formulation particulière du problème d'optimisation qui rend possible l'utilisation d'un algorithme d'optimisation du type SMO (Séquentiel Minimal Optimization) (Platt, 1998; Crammer & Singer, 2002; Aiolli & Sperduti, 2003), réputé nettement plus performant et plus rapide que les algorithmes de descente de gradient.

Notre algorithme présente des avantages sur l'algorithme de gradient projeté (Bertsekas, 1999) utilisé dans (Sha & Saul, 2006, 2007). Tout d'abord, par notre prise en compte des contraintes, nous évitons l'étape de projection de la solution dans l'espace des contraintes. Cette étape est d'une part assez lourde dans le cas de contraintes sur les matrices de covariance, et est d'autre part source d'erreurs numériques lorsque la dimension des données est importante. Notre approche, basée sur l'algorithme SMO, converge bien plus rapidement et mieux si bien que notre approche est expérimentalement plus performante et moins sensible à l'initialisation que l'algorithme original proposé par les auteurs.

Dans la suite, nous présentons l'algorithme de départ, proposé dans (Sha & Saul, 2006, 2007) en Section 2. Puis en Section 3, nous détaillons notre approche en reformulant le problème sous sa forme duale et nous explicitons l'algorithme SMO dans notre cas. Enfin, nous fournissons en Section 4 des résultats expérimentaux permettant d'évaluer l'apport de notre algorithme en termes de vitesse de convergence et de performance pour le problème de classification de chiffres manuscrits, et sur deux bases de données internationales de référence.

## 2 Classification avec des MG

Nous nous focalisons ici sur un problème de discrimination pour des données vectorielles en dimension  $d$ , le but est de déterminer le label  $y$  correspondant à une donnée  $x = [x^1, x^2, \dots, x^d]$ . Nous décrivons ici brièvement l'approche classiquement employée pour classifier des données avec des modèles Gaussiens, puis nous présentons les travaux proposés dans (Sha & Saul, 2006, 2007).

### 2.1 Approche non discriminante

En supposant une loi de probabilité sur l'ensemble des vecteurs et les classes, l'approche probabiliste consiste à estimer cette loi de probabilité et à classifier en déterminant la classe  $y$  de probabilité a posteriori maximale, ou de façon équivalente la classe

maximisant la probabilité jointe  $P(x, y)$ , c'est cette dernière stratégie qui est implémentée en pratique. La fonction de décision est définie par :

$$\hat{y} = \underset{y}{\operatorname{argmax}} p(x|y) \times p(y) \quad (2)$$

où  $p(x|y)$  est la densité de probabilité de la classe  $y$  donnée par l'équation (1), et  $p(y)$  est la probabilité a priori de la classe  $y$ . Les lois composantes des densités sont des lois normales du type :

$$p(x|N_{y,k}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_{y,k}|}} \exp \left[ -\frac{1}{2} (x - \mu_{y,k}) \Sigma_{y,k}^{-1} (x - \mu_{y,k}) \right] \quad (3)$$

où  $N_{y,k}$  représente la  $k^{ieme}$  loi gaussienne de la densité de la classe  $y$ ,  $\mu_{y,k}$  et  $\Sigma_{y,k}$  représentent respectivement la moyenne et la matrice de covariance de la  $k^{ieme}$  loi gaussienne de la densité de la classe  $y$ .

L'apprentissage consiste à rechercher les paramètres des modèles des densités maximisant la vraisemblance (MV) de l'ensemble des données d'apprentissage  $(x_1, y_1), \dots, (x_N, y_N)$ . L'optimisation est réalisée par un algorithme EM qui consiste à itérer deux étapes jusqu'à convergence : une étape d'estimation des variables cachées (quelle composante a produit quelles données d'apprentissage) et une étape de maximisation. Ces deux étapes sont itérées jusqu'à la convergence.

Dans la pratique des problèmes apparaissent souvent dans l'apprentissage de tels modèles de mélange, notamment pour des données en " grande " dimension. Les matrices de covariance obtenues ne sont pas toujours bien conditionnées et leur inversion pose problème. Une technique répandue consiste à régulariser les solutions. Dans notre implémentation, à chaque étape de EM, après l'étape de ré-estimation, on régularise les matrices de covariances en ajoutant à celles-ci une faible valeur sur la diagonale :

$$\Sigma_{y,k} = \Sigma_{y,k} + \lambda Id \quad (4)$$

où  $\lambda$  est en général choisi en fonction des valeurs sur la diagonale de la matrice. Nous nommons dans la suite cette approche, l'approche MV régularisée.

Notons qu'une règle de décision alternative consiste à affecter à un exemple la classe dont une des lois composantes est a priori la plus probable.

$$\hat{y} = \underset{y}{\operatorname{argmax}} \left[ \max_k p(x|N_{y,k}) \times p(k|y) \times p(y) \right] \quad (5)$$

## 2.2 Approche par maximisation de la marge

Nous décrivons ci-dessous une approche proposée dans (Sha & Saul, 2006, 2007) pour apprendre des mélanges Gaussiennes en maximisant la marge. Nous présentons le principe puis l'algorithme d'apprentissage proposé par les auteurs.

### 2.2.1 Principe

Considérons tout d'abord le cas d'une loi Gaussienne par classe. (Sha & Saul, 2006, 2007) ont proposé de mettre la fonction de décision sous la forme d'une fonction discriminante exploitant une distance de type Mahalanobis. Considérons la matrice  $\Phi_y$  définie à partir des paramètres de la loi Gaussienne de la classe  $y$ , la moyenne  $\mu_y$  et la matrice de covariance inverse  $\psi_y = \Sigma_y^{-1}$  :

$$\Phi_y = \begin{bmatrix} \psi_y & -\psi_y \mu_y' \\ -\mu_y' \psi_y & \mu_y \psi_y \mu_y' + \beta_y \end{bmatrix} \quad (6)$$

où  $\beta_y$  est un paramètre réel qui représente le logarithme de la probabilité a priori de la classe  $y$ . Notons que  $\psi_y$  est une matrice semi définie positive car elle est l'inverse d'une matrice de covariance, qui est elle-même semi définie positive. En notant  $z = [x, 1] = [x^1, x^2, \dots, x^d, 1]$  une forme étendue, la fonction de décision prend la forme :

$$\hat{y} = \operatorname{argmin}_y z \Phi_y z' \quad (7)$$

L'intérêt de la formulation précédente est que la fonction score, que l'on souhaite optimiser, devient linéaire en la matrice  $\Phi_y$ . On peut donc exploiter l'ensemble des techniques d'optimisation développées notamment pour l'apprentissage de Machines à Vecteurs de Supports. (Sha & Saul, 2006, 2007) ont proposé d'apprendre les paramètres  $\Phi_y$  de façon discriminante en minimisant le risque empirique par maximisation de la marge. Un exemple  $z_i$  de la classe  $y_i$  est bien classé si  $z_i \Phi_{y_i} z_i' < z_i \Phi_y z_i' \forall y \neq y_i$ .

Comme on le fait classiquement, on peut également introduire des variables *ressort* pour traiter le cas de données non linéairement séparables. Le problème d'optimisation, pour une base de données d'apprentissage  $(x_1, y_1), \dots, (x_N, y_N)$ , s'écrit donc :

$$\begin{array}{ll} \min_{\Phi, \xi} & \frac{1}{2} \sum_y \|\psi_y\|^2 + C \sum_i \xi_i \\ \text{sous les contraintes} & z_i \Phi_{y_i} z_i' \leq z_i \Phi_y z_i' - 1 + \xi_i \quad \forall i \forall y \neq y_i \\ & \xi_i \geq 0 \quad \forall i \\ & \psi_y \succ 0 \quad \forall y \end{array} \quad (8)$$

où  $\psi_y \succ 0$  signifie que la matrice  $\psi_y$  est semi-définie positive. Notons que le premier facteur du critère ne régularise que partiellement les matrices  $\Phi_y$ , car il ne paraît pas justifié de régulariser les éléments de ces matrices liés à la moyenne de la loi Gaussienne.

Cette formulation est très intéressante dans la mesure où la fonction de coût obtenue est quadratique et où les contraintes sont soit linéaires soit convexes. Notons que les contraintes de symétrie ne posant pas de problème particulier, nous les ignorons afin de simplifier la présentation.

Afin d'étendre cette formulation au cas de mélanges de  $K$  gaussiennes par classe, on note  $y_i$  la classe d'un exemple  $x_i$ ,  $k_i$  la composante qui l'a produit et  $r_i = (k_i, y_i)$  l'identifiant de la gaussienne qui a produit  $x_i$ . Pour un exemple  $x$  quelconque, on notera  $r = (k, y)$  l'identifiant correspondant. Aussi, on notera  $R(y)$  l'ensemble des gaussiennes de la classe  $y$ . Dans le cas où les  $r_i$  sont connus c'est-à-dire que l'on sait quelle composante a produit quelle donnée (nous revenons sur le cas où cela est inconnu plus loin), le problème devient :

$$\begin{aligned}
& \min_{\Phi, \xi} \quad \frac{1}{2} \sum_r \|\psi_r\|^2 + C \sum_i \xi_i \\
& \text{s.c} \quad \begin{aligned}
& z_i \Phi_{r_i} z'_i \leq z_i \Phi_r z'_i - 1 + \xi_i \quad \forall i \forall r \notin R(y_i) \\
& \xi_i \geq 0 \quad \forall i \\
& \psi_r \succ 0 \quad \forall r
\end{aligned}
\end{aligned} \tag{9}$$

### 2.2.2 Optimisation

(Sha & Saul, 2006, 2007) proposent d'éliminer les variable ressort dans l'équation (9) en introduisant la fonction *hinge*, où  $hinge(z) = \max(0, z)$ . Le problème peut alors être transformé en :

$$\begin{aligned}
& \min_{\Phi} \quad \frac{1}{2} \sum_r \|\psi_r\|^2 + C \sum_{i=1:N} \sum_{r \notin R(y_i)} hinge(1 + z_i \Phi_{r_i} z'_i - z_i \Phi_r z'_i) \\
& \text{s.c} \quad \psi_r \succ 0 \quad \forall r
\end{aligned} \tag{10}$$

C'est un problème d'optimisation dont la fonction objectif est convexe et dont les contraintes sont également convexes. On peut le résoudre avec une technique de descente de gradient projeté (Bertsekas, 1999; Ratliff *et al.*, 2007). A chaque mise à jour des paramètres, on vérifie si les contraintes sont satisfaites, si ce n'est pas le cas on cherche la projection de l'ensemble des paramètres dans l'espace des paramètres satisfaisant les contraintes. Ici, on vérifie que les matrices  $\psi_r$ , c'est-à-dire les matrices inverses des matrices de covariance, sont semi-définies positives, et si ce n'est pas le cas on les projette dans l'espace des matrices semi-définies positives. Le but est de trouver la matrice qui est le plus proche de la matrice à projeter mais cette solution n'est pas aisée à trouver. L'étape de projection proposée dans (Sha & Saul, 2006, 2007) est moins coûteuse et consiste à annuler les valeurs propres négatives, s'il en existe, de la matrice  $\psi_r$ . Comme décrit par les auteurs, cette méthode d'optimisation converge très lentement, si bien que l'initialisation doit être la meilleure possible.

En pratique, ces auteurs utilisent comme initialisation (moyennes, matrices de covariance etc) l'ensemble des solutions trouvées par apprentissage d'un mélange de Gaussiennes pour chaque classe par maximum de vraisemblance. Egalement, ils fixent les variables cachées  $r_i$  à partir de la solution MV. Après apprentissage, les exemples d'une classe sont affectés à la composante de la classe de probabilité a posteriori maximale.

Enfin, afin d'accélérer la convergence, ils proposent d'optimiser les matrices racines carrées  $\Omega_y$  où  $\Phi_r = \Omega_r \Omega_r'$ . Cela rend l'optimisation non convexe mais permet d'éviter les contraintes de semi-définition positive et donc les étapes de projections de matrices.

## 3 Optimisation du dual pour l'apprentissage de Mélanges Gaussiens par maximisation de la marge

Dans la section précédente, nous avons décrit le problème d'optimisation obtenu dans (Sha & Saul, 2006, 2007), incluant des contraintes de semi-définition positive (noté SDP dans la suite) des matrices  $\psi_r$ . Il apparaît que la présence de ces contraintes, non linéaires, interdit le passage à la formulation duale du problème d'optimisation, qui a été

démontrée expérimentalement comme étant plus efficace et plus fiable que la descente de gradient pour (Platt, 1998; Crammer & Singer, 2002; Aiolli & Sperduti, 2003). Dans la suite, nous proposons une autre formulation du problème conduisant à une autre technique d'optimisation, plus proche de techniques d'optimisation utilisées pour les machines à vecteurs de support.

### 3.1 Mise sous forme duale

Tout d'abord nous remarquons que  $M \succ 0 \iff \forall x, xMx \geq 0$  et nous proposons de remplacer les contraintes de SDP des matrices  $\psi_r$  par un ensemble de contraintes de type  $xMx \geq 0$ . En ne considérant cette contrainte que pour les points de la base d'apprentissage, l'optimisation dans l'équation (9) devient une instance de programmation quadratique :

$$\begin{aligned} \min_{\Phi, \xi, \theta} \quad & \frac{1}{2} \sum_r \|\psi_r\|^2 + C \sum_i \xi_i \\ \text{s.c} \quad & z_i \Phi_{r_i} z_i' \leq z_i \Phi_r z_i' - 1 + \xi_i \quad \forall i \forall r \notin R(y_i) \\ & \xi_i \geq 0 \quad \forall i \\ & (x_i - \mu_t) \psi_r (x_i - \mu_t)' \geq 0 \quad \forall i, \forall r \end{aligned} \quad (11)$$

où  $\mu_t$  est la moyenne totale des exemples et n'est pas considérée comme une variables dans la suite. Bien entendu la satisfaction des contraintes  $(x - \mu_t) \psi_r (x - \mu_t)' \geq 0$  sur l'ensemble des points d'apprentissage ne garantit pas que  $\psi_r$  soit SPD, mais en pratique nous n'avons pas rencontré de cas dans nos expérimentations où la matrice ne le soit pas.

Pour rendre la présentation plus claire, nous introduisons les variables temporaires  $\theta_i$  (Aiolli & Sperduti, 2003), ce qui ne change pas la solution globale du problème. Nous obtenons le problème primal suivant :

$$\begin{aligned} \min_{\Phi, \xi, \theta} \quad & \frac{1}{2} \sum_r \|\psi_r\|^2 + C \sum_i \xi_i \\ \text{s.c} \quad & z_i \Phi_{r_i} z_i' \leq \theta_i - 1 + \xi_i \quad \forall i \\ & \theta_i \leq z_i \Phi_r z_i' \quad \forall i \forall r \notin R(y_i) \\ & \xi_i \geq 0 \quad \forall i \\ & (x_i - \mu_t) \psi_r (x_i - \mu_t)' \geq 0 \quad \forall i, \forall r \end{aligned} \quad (12)$$

La solution de ce problème d'optimisation est déterminée à partir du Lagrangien :

$$\begin{aligned} L = \quad & \frac{1}{2} \sum_r \|\psi_r\|^2 + C \sum_i \xi_i \\ & + \sum_i \alpha_i^{r_i} [z_i \Phi_{r_i} z_i' - \theta_i + 1 - \xi_i] \\ & + \sum_{i, r \notin R(y_i)} \alpha_i^r [\theta_i - z_i \Phi_r z_i'] \\ & - \sum_i \lambda_i \xi_i \\ & - \sum_{i, r} \gamma_i^r (x_i - \mu_t) \psi_r (x_i - \mu_t)' \end{aligned} \quad (13)$$

où  $\alpha, \gamma, \lambda$  sont des multiplicateurs de Lagrange. La solution est donnée par un point selle du Lagrangien, qui doit être minimisé par rapport aux paramètres  $\Phi, \xi, \theta$  et maximisé par rapport aux multiplicateurs  $\alpha, \gamma, \lambda$ , ceci sous les contraintes  $\alpha_i^r \geq 0, \gamma_i^r \geq 0, \lambda_i \geq 0$ . Au point scelle, la dérivation du Lagrangien par rapport aux variables  $\Phi, \xi, \theta$  doit être nulle, ce qui conduit à :

$$\frac{\delta L}{\delta \xi_i} = 0 \iff C - \alpha_i^{r_i} - \lambda_i = 0 \quad (14)$$

$$\frac{\delta L}{\delta \theta_i} = 0 \iff -\alpha_i^{r_i} + \sum_{r \notin R(y_i)} \alpha_i^r = 0 \iff \sum_r y_i^r \alpha_i^r = 0 \quad (15)$$

$$\text{où } y_i^r = \begin{cases} +1 & \text{si } r = r_i \\ -1 & \text{si } r \notin R(y_i) \\ 0 & \text{sinon} \end{cases} \quad \frac{\delta L}{\delta \phi_q} = 0 \quad (16)$$

$$\iff \begin{cases} \psi_q & = \sum_i \gamma_i^q (x_i - \mu_t)'(x_i - \mu_t) - \sum_i y_i^q a_i^q x_i' x_i \\ \sum_i y_i^q \alpha_i^q x_i' & = 0 \\ \sum_i y_i^q \alpha_i^q & = 0 \end{cases} \quad (17)$$

En remplaçant dans l'expression du Lagrangien les expressions obtenues dans les équations (14), (15), et (17), et en éliminant les termes de valeur nulle, on obtient le problème dual <sup>1</sup> :

$$\begin{aligned} \max_{\alpha, \gamma} \quad & -\frac{1}{2} \sum_r \|\psi_r\|^2 + \sum_i \alpha_i^{r_i} \\ \text{s.c} \quad & \alpha_i^{r_i} > 0, \gamma_i^r > 0, \alpha_i^{r_i} < C \quad \forall i \forall r \\ & \sum_r y_i^r \alpha_i^r = 0 \quad \forall i \\ & \sum_i y_i^r \alpha_i^r = 0 \quad \forall r \\ & \sum_i y_i^r \alpha_i^r x_i = 0 \end{aligned} \quad (18)$$

## 3.2 Optimisation par SMO

Le problème dual de l'équation (18) est une instance de programmation quadratique mais dont les contraintes sont plus simples à manipuler que celles du problème primal de l'équation (11), comme nous allons le voir. Afin d'optimiser efficacement ce problème, nous avons cherché comment le décomposer en plus petits problèmes que l'on peut résoudre analytiquement. Cette stratégie est utilisée dans les algorithmes de type SMO (Optimisation Séquentielle Minimale) pour résoudre un problème donné en un temps linéaire dans le nombre d'exemples d'apprentissage.

### 3.2.1 Principe

L'idée de l'algorithme SMO, utilisé dans les machines à vecteurs de support, consiste à sélectionner itérativement des exemples de l'ensemble d'apprentissage et à optimiser le plus possible la fonction objective par rapport aux variables associées à l'exemple sélectionné (Platt, 1998; Crammer & Singer, 2002; Aiolli & Sperduti, 2003). Cette

<sup>1</sup>Notons que les variables  $\lambda_i$  ont désormais disparu dans la fonction objectif. Pour les éliminer complètement nous avons utilisé l'équation (11) pour remplacer les contraintes  $\lambda_i \geq 0$  par les contraintes  $\alpha_i^{r_i} \leq C$ .

dernière optimisation est réalisée par itération d'une étape d'optimisation minimale, pour une paire de variables qui sont liées par une contrainte. L'idée est que cette étape minimale d'optimisation puisse être réalisée analytiquement. Voici le pseudo code de cet algorithme :

```
Fonction Optimisation_globale
  repeat
    Sélectionner un exemple d'apprentissage x
    Gainx = Optimisation(x)
  until Gainx < epsilon
Fin
```

```
Fonction Gain=Optimisation(x)
  Gain = 0;
  repeat
    Sélectionner deux variables ra, rb
    gain = Optimiser(x, ra, rb)
    Gain = Gain + gain
  until gain < epsilon
  return Gain
Fin
```

Diverses étapes de cet algorithme reposent, au moins partiellement, sur des heuristiques, c'est le cas du choix de l'exemple pour lequel optimiser les variables dans la boucle de la fonction *Optimisation\_Globale* et du choix de la paire de variables dans la boucle de la fonction *Optimisation*.

Dans notre implémentation, la sélection de l'exemple est réalisée en évaluant effectivement pour chaque exemple le gain escompté, celui-ci n'étant calculé que pour une paire de variables bien choisie. Dans la fonction *Optimisation* la sélection de la paire de variables est une étape plutôt coûteuse mais on peut s'appuyer sur les conditions KKT pour déterminer efficacement la paire de variables optimale (Aiolli & Sperduti, 2003).

### 3.2.2 Application de SMO pour l'apprentissage maximisation de la marge de MG

L'application de l'algorithme SMO dans notre cas n'est pas immédiate. En effet, la contrainte  $\sum_i y_i^r \alpha_i^r x_i = 0$  dans l'équation (18) pose problème. Cette contrainte constitue en réalité un système d'équations liant des variables associées à tous les exemples d'apprentissage (les sommes portent sur les indices  $i$  des exemples d'apprentissage). De ce fait il existe (ou il peut exister) des paires de variables que l'on ne peut modifier dans une étape de SMO, c'est-à-dire telles que leurs valeurs ne peuvent être modifiées tout en continuant à satisfaire le système d'équations ci-dessus. En d'autres termes, modifier certaines variables tout en satisfaisant le système d'équations peut requérir la modification de plus de deux variables.

Notons que la contrainte concernée provient des quantités dans la dernière colonne et dans la dernière ligne des matrices  $\Phi_r$ , ce sont les quantités  $\mu_r \psi_r$  et  $\mu_r \psi_r m u_r' + \beta_r$ . Nous considérerons dans la suite que ces quantités sont des variables et nous les

noterons  $\Xi_r$ . Cela fait sens car, pourvu que la matrice  $\psi_r$  soit inversible (en fait strictement définie positive puisqu'elle est déjà SDP) les quantités  $\nu_r = \mu_r \psi_r$  et  $\delta_r = \mu_r \psi_r m u'_r + \beta_r$  peuvent être vues comme des variables indépendantes de  $\psi_r$ . En pratique, les matrices  $\psi_r$  ne sont pas toujours inversibles mais cette stratégie conduit à de bons comportements en termes de convergence. En nous appuyant sur ces éléments nous proposons pour contourner le problème posé par la contrainte évoquée plus haut, de séparer l'ensemble de variables à estimer en deux sous-ensembles, les matrices  $\psi_r$  d'une part et le reste des paramètres  $\Xi_r$  d'autre part, et d'optimiser alternativement l'un puis l'autre de ces ensembles de paramètres. Le fait que la fonction objective soit convexe et que les contraintes soient convexes en  $\Phi_r$  garantit la convergence vers une solution globale. Le cas de l'optimisation par rapport à  $\Xi_r$  est simple car linéaire, nous ne le détaillons pas ici. En revanche, nous revenons maintenant sur l'optimisation par rapport aux  $\psi_r$  et détaillons l'algorithme SMO. Nous reprenons tout d'abord le problème primal de l'équation (11), en considérant que l'optimisation concerne les  $\psi_r$  seules :

$$\begin{aligned} \min_{\psi, \xi, \theta, \delta} \quad & \frac{1}{2} \sum_r \|\psi_r\|^2 + C \sum_i \xi_i \\ \text{s.c} \quad & x_i \psi_{r_i} x'_i - 2x_i \nu_{r_i} + \delta_{r_i} \leq \theta_i - 1 + \xi_i \quad \forall i \\ & \theta_i \leq x_i \psi_r x'_i - 2x_i \nu_r + \delta_r \quad \forall i \forall r \notin R(y_i) \\ & \xi_i \geq 0 \quad \forall i \\ & (x_i - \mu_t) \psi_r (x_i - \mu_t)' \geq 0 \quad \forall i \forall r \end{aligned} \quad (19)$$

Ici, les quantités  $\Xi_r$  sont considérées constantes <sup>2</sup>. Le problème de l'équation (16) est toujours une instance de programme quadratique, et l'on peut obtenir le dual de la même façon que précédemment, mais ici on n'obtient que'une seule équation pour  $\frac{\delta L}{\delta \psi_r} = 0$ , ce qui donne :

$$\psi_r = \sum_i \gamma_i^r (x_i - \mu_t)' (x_i - \mu_t) - \sum_i y_i^r a_i^r x'_i x_i \quad (20)$$

Nous obtenons ainsi le problème dual :

$$\begin{aligned} \max_{\alpha, \gamma} \quad & -\frac{1}{2} \sum_r \|\psi_r\|^2 + \sum_i \alpha_i^r + \sum_{i,r} \alpha_i^r y_i^r (-2x_i \nu_r + \delta_r) \\ \text{s.c} \quad & \alpha_i^r > 0, \gamma_i^r > 0, \alpha_i^r < C \quad \forall i \forall r \\ & \sum_r y_i^r \alpha_i^r = 0 \quad \forall i \\ & \sum_i y_i^r \alpha_i^r = 0 \quad \forall r \end{aligned} \quad (21)$$

où  $\psi_r = \sum_i \gamma_i^r (x_i - \mu)' (x_i - \mu) - \sum_i y_i^r a_i^r x'_i x_i$

### 3.2.3 Etape élémentaire

Le problème de (21) est prêt à être décomposé. Nous présentons maintenant l'étape d'optimisation élémentaire correspondant à un exemple  $x_i$ . Les variables associées à

<sup>2</sup>Dans nos expériences, nous avons également envisagé une implémentation alternative qui consiste à considérer que les paramètres  $\mu_r$  et  $\beta_r$  sont des constantes et à laisser les  $\Xi_r$  varier (car défini à partir de  $\mu_r$  et  $\beta_r$  et  $\psi_r$ ) pendant la phase d'optimisation des  $\psi_r$ .

un exemple  $x_i$  sont  $\alpha_i^r$  et  $\gamma_i^r$ , et les contraintes sur ces variables sont  $\alpha_i^r > 0, \alpha_i^{r_i} \leq C, \gamma_i^r > 0$ , et  $\alpha_i^{r_i} = \sum_{r \notin R(y_i)} \alpha_i^r$ .

L'optimisation des  $\alpha_i^r$ , comme nous l'avons mentionné plus haut, consiste dans un premier temps à sélectionner une paire de variables  $r_a$  et  $r_b$  (correspondant à deux composantes gaussiennes), et à chercher les nouvelles valeurs de  $\alpha_i^{r_a}, \alpha_i^{r_b}$  maximisant le dual, tout en respectant la contrainte qui lie ces variables  $\alpha_i^{r_i} = \sum_{r \notin R(y_i)} \alpha_i^r$ .

On distingue deux cas. Dans le premier cas, une des gaussiennes est la gaussienne associée à l'exemple considéré,  $x_i$ . Sans perdre en généralité supposons que  $r_a = r_i$ . Alors si on ajoute une valeur  $v$  à  $\alpha_i^{r_a}$ , on doit ajouter une valeur  $v$  à  $\alpha_i^{r_b}$  qui n'appartient pas à  $R(x_i)$ . L'optimisation consiste alors à déterminer la valeur  $v$  qui maximise le dual. Celui-ci étant une fonction quadratique de  $v$ , toutes les autres variables étant figées, on détermine la valeur optimale de  $v$  analytiquement. Dans le second cas, les deux variables correspondent à deux gaussiennes n'appartenant pas à  $R(y_i)$ . Alors pour que  $\sum_{r \notin R(y_i)} \alpha_i^r$  ne change pas, si on ajoute une valeur  $v$  à  $\alpha_i^{r_a}$ , on doit retrancher cette valeur à  $\alpha_i^{r_b}$ . Le dual s'exprime ici encore comme une fonction quadratique de  $v$  et on détermine la valeur optimale de  $v$  analytiquement.

Notons que si la valeur  $v$  trouvée  $v^*$  provoque la violation d'une contrainte de type ( $\alpha_i^r \geq 0, \alpha_i^{r_i} \leq C$ ) alors on choisit la valeur  $v$  la plus proche de  $v^*$  mais satisfaisant la contrainte. Par exemple : si  $\alpha_i^r + v^* \leq C$  alors on choisit  $v = v^*$  et sinon on choisit  $v = C - \alpha_i^{r_i}$ .

Pour terminer, l'optimisation des variables  $\gamma_i^r$  ne pose aucune difficulté car il n'y a pas de contraintes liant les  $\gamma_i^r$ .  $\gamma_i^r$  n'intervient que dans l'expression de  $\psi_r$  (Cf. Eq. (21)) si bien que l'on peut là encore sans difficulté déterminer analytiquement le changement optimal  $v$  de la variable  $\gamma_i^r$ .

## 4 Expériences

Nous décrivons ici des résultats expérimentaux obtenus en reconnaissance de chiffres manuscrits sur deux bases de données internationales de référence, la base USPS (LeCun *et al.*, 1989) et la base MNIST<sup>3</sup>. La base USPS contient 7291 exemples d'apprentissage et 2007 exemples de test, chaque chiffre est représenté par une image de dimension 16x16. La base MNIST contient 60000 exemples d'apprentissage et 10000 exemples de test, les images sont en dimension 28x28. Pour ces deux bases, nous avons prétraité les données via une analyse en composantes principales (ACP) afin de réduire la dimension des données à 50 dimensions pour les deux bases (on ne garde que les 50 composantes des images sur les 50 axes principaux d'inertie). Il s'agit d'un prétraitement standard sur ces données, décrit par exemple dans (LeCun *et al.*, 1998).

Nos expériences visent notamment à comparer les résultats obtenus par les deux méthodes d'optimisations : Le gradient projeté utilisé avec la fonction hinge dans (Sha & Saul, 2006, 2007) et l'algorithme SMO dans notre cas. A ce titre, nous nous intéressons tout d'abord à comparer les vitesses de convergence entre les deux méthodes. Puis nous nous intéressons également à la performance pure obtenue avec chacune des deux méthodes, ainsi qu'à la sensibilité à l'initialisation. Dans toutes les expériences décrites ici

<sup>3</sup><http://yann.lecun.com/exdb/mnist/index.html>

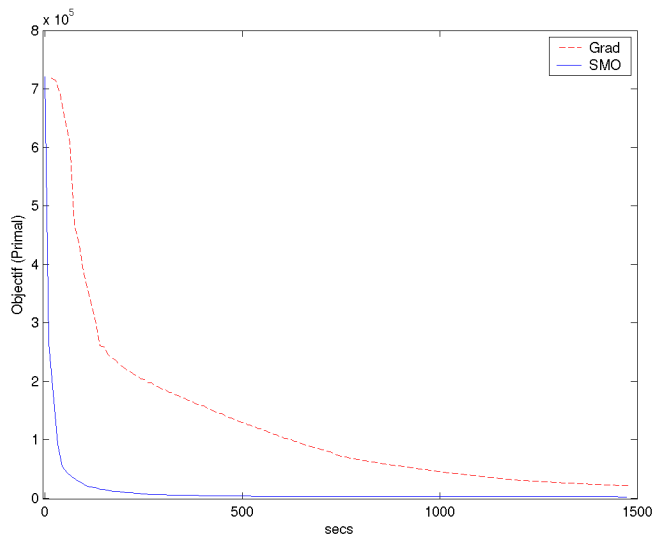


FIG. 1 – Comparaison de la vitesse de convergence pour l’algorithme de (Sha & Saul, 2006) utilisant un gradient projeté et notre algorithme utilisant SMO. L’apprentissage est ici réalisé sur les données USPS.

L’apprentissage des modèles par l’une ou l’autre des deux techniques est initialisé par le résultat d’un apprentissage non discriminant. On réalise une étape d’initialisation en apprenant des modèles MMG indépendamment pour chaque classe avec un algorithme EM et un critère de Maximum de Vraisemblance (MV). Cela permet d’obtenir l’affectation des exemples aux composantes des mélanges (i.e. les  $r_i$ ) et des quantités initiales pour les moyennes de toutes les lois gaussiennes  $\mu_r$ .

Nous commençons par comparer la vitesse de convergence des deux techniques. Les deux algorithmes de maximisation de la marge sont lancés à partir de l’initialisation MV en utilisant le même hyper paramètre C. La figure 1 montre l’évolution du critère d’optimisation (le primal) en fonction du temps pour la méthode de gradient (Grad) et pour notre algorithme (SMO). La valeur absolue du temps importe peu ici, le point qui nous intéresse réellement est la différence de convergence entre les deux méthodes. Notons ici que la méthode Grad optimise directement ce critère primal, tandis que notre méthode SMO l’optimise indirectement à travers la maximisation du dual. On voit bien sur ces courbes que l’approche SMO converge beaucoup plus vite que l’approche de gradient projeté. Dans cette expérience, après 300 secondes, SMO arrive déjà à un point très proche de la convergence alors qu’il faut 3000 secondes à l’algorithme Grad pour arriver au même point. En théorie, le problème d’optimisation dans l’équation (10) est convexe et on peut trouver la solution optimale par une descente de gradient projeté. Mais en pratique la convergence est très lente et l’algorithme nécessite une bonne initialisation. On observe expérimentalement que la solution trouvée par l’algorithme de gradient est très dépendante de l’initialisation, car des problèmes numériques em-

pêchent souvent de converger jusqu'à la solution optimale. De ce point de vue, notre méthode apparaît plus robuste et moins sensible à l'initialisation.

Les figures suivantes (Figures (2) et (3) illustrent la sensibilité à l'initialisation des deux approches. Elles comparent les performances de l'approche non discriminante régularisée (EM) et des deux approches discriminantes pour différentes valeurs du paramètre de régularisation (Cf Eq.(4)). Comme précédemment le résultat de l'apprentissage MV est pris comme initialisation des approches discriminantes. La Figure 2 montre les résultats obtenus avec deux gaussiennes par modèle de mélange (i.e.  $K = 2$  dans l'Eq.(1)), alors que la Figure (3) montre ces résultats avec  $K = 4$ .

On voit ici que quelle que soit l'initialisation EM, et pour ces deux types de modèles ( $K = 2$  et  $K = 4$ ), notre approche obtient de meilleurs résultats que l'approche de référence (Sha & Saul, 2006). On voit aussi que les résultats obtenus par notre algorithme sont moins sensibles à l'initialisation, ce qui est cohérent avec les résultats obtenus précédemment sur la convergence. Ce point est en pratique très intéressant car une bonne initialisation par EM, si elle est simple du point de vue théorique, pose le plus souvent problème et requiert une régularisation. Or il n'est pas facile de déterminer le paramètre de régularisation "optimal" automatiquement. De ce point de vue, notre approche étant assez peu sensible à l'initialisation, on peut se permettre d'utiliser une valeur non optimale du paramètre de régularisation  $\lambda$  obtenue automatiquement puis d'affiner le modèle par maximisation de la marge.

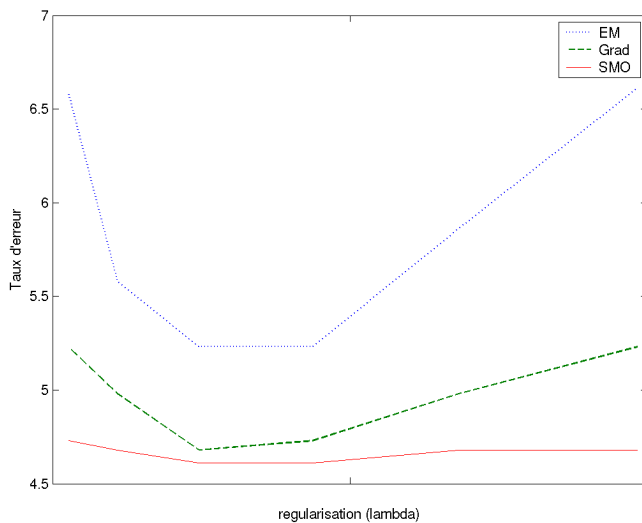


FIG. 2 – Performance, sur la base USPS, de l'apprentissage MV régularisé (EM), de l'approche de (Sha & Saul, 2006) (Grad) et de notre algorithme (SMO) en fonction du paramètre de régularisation de l'apprentissage MV ( $\lambda$  dans l'équation (4)). Les modèles de classe sont des mélanges de deux Gaussiennes.

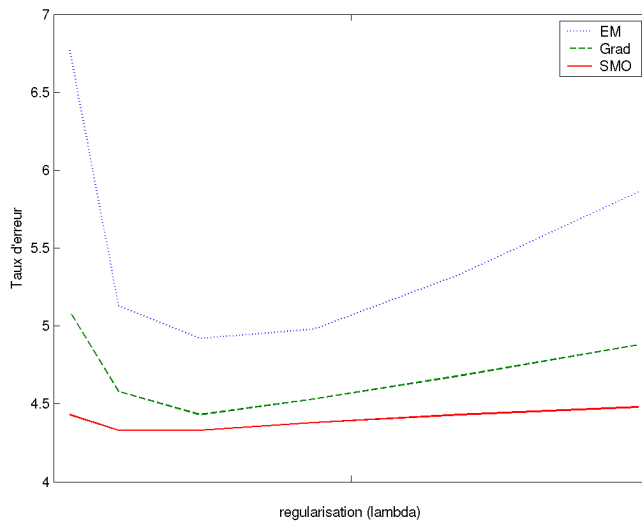


FIG. 3 – Performance, sur la base USPS, de l'apprentissage MV régularisé (EM), de l'approche de (Sha & Saul, 2006) (Grad) et de notre algorithme (SMO) en fonction du paramètre de régularisation de l'apprentissage MV ( $\lambda$  dans l'équation (4)). Les modèles de classe sont des mélanges de quatre Gaussiennes.

TAB. 1 – Taux d'erreur en classification sur la base USPS, pour la méthode de MV régularisée, l'approche de (Sha & Saul, 2006) (Grad) et notre approche (SMO), dans le cas où la solution MV est déterminée avec un paramètre de régulation élevé (a) et faible (b).

K	EM	Grad	SMO	K	EM	Grad	SMO
1	7.22	5.23	4.88	1	5.83	5.13	4.88
2	6.61	5.23	4.68	2	5.30	4.68	4.61
4	5.86	4.88	4.48	4	4.92	4.43	4.33
6	5.46	4.73	4.43	6	4.90	4.43	4.33

(a)

(b)

Nous fournissons pour terminer des tableaux récapitulatifs des performances des trois méthodes sur les bases USPS et MNIST, pour différentes valeurs de  $K$  (nombre de composantes par modèle) et pour deux cas de régularisation, une valeur faible et une valeur forte. Peu importe les valeurs exactes ici, nous voulons ici montrer les différences de comportement pour deux cas très différents de régularisation.

Les tableaux 1a et 1b comparent les différentes méthodes pour une forte valeur de régularisation (Tableau 1a) et une faible valeur de régularisation (Tableau 1b). Tout d'abord, on voit bien ici que notre algorithme permet systématiquement d'obtenir des

TAB. 2 – Taux d’erreur en classification sur la base MNIST, pour la méthode de MV régularisée, l’approche de (Sha & Saul, 2006) (Grad) et notre approche (SMO), dans le cas où la solution MV est déterminée avec un paramètre de régulation élevé (a) et faible (b).

K	EM	Grad	SMO	K	EM	Grad	SMO
1	5.72	2.31	2.03	1	3.93	2.10	2.03
2	5.01	2.24	1.91	2	3.48	2.05	1.90
4	3.72	2.02	1.79	4	2.65	1.99	1.79
8	3.00	1.91	1.69	8	2.07	1.78	1.69

(a) (b)

performances similaires ou meilleures que l’algorithme de Gradient, ce qui correspond à ce qui a été observé dans les Figures précédentes. On note également que notre approche est moins sensible à l’initialisation. Enfin, on remarque que la différence entre les deux algorithmes est moins nette lorsque la régularisation est plutôt faible. Les tableaux 2a et 2b fournissent le même type de résultats pour la base MNIST. Les mêmes commentaires peuvent être tirés de ces résultats.

## 5 Conclusion

Nous avons proposé un nouvel algorithme d’apprentissage de mélanges de Gaussiennes par maximisation de la marge. Pour cela, nous avons repris un formalisme proposé précédemment et qui reposait sur un algorithme de gradient projeté. Nous avons revu la formulation du problème et la prise en compte de contraintes sur la semi-définition positive des matrices de covariance, ce qui nous a permis de dériver un nouvel algorithme d’apprentissage de type SMO. Notre approche s’est montrée expérimentalement plus rapide en convergence, ce qui explique en partie ses meilleures performances testées en reconnaissance de chiffres manuscrits sur deux bases de données réelles de référence. Nous travaillons maintenant à l’extension de ce type d’algorithmes au traitement de séquences par l’apprentissage de modèles Markoviens.

## Références

- AFIFY M. (2005). Extended baum-welch reestimation of gaussian mixture models based on reverse jensen inequality. In *INTERSPEECH*, p. 1113–1116.
- AIOLLI F. & SPERDUTI A. (2003). Multi-prototype support vector machine. In *IJCAI*, p. 541.
- BERTSEKAS D. (1999). *Nonlinear programming*. Athena Scientific, 2nd edition.
- CRAMMER K. & SINGER Y. (2002). On the learnability and design of output codes for multiclass problems. *Machine Learning*, **47**, 201.
- DAHMEN J., SCHLUTER R. & NEY H. (1999). Discriminative training of gaussian mixtures for image object recognition. In *DAGM-Symposium*, p. 205–212.

- DEMPSTER A. P., LAIRD N. M. & RUBIN D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. In *Journal of the Royal Statistical Society (Series B)*, 39 :1-38.
- JAAKKOLA T., DIEKHANS M. & HAUSSLER D. (1999). Using the fisher kernel method to detect remote protein homologies. In *International Conference on Intelligent Systems for Molecular Biology*.
- JUANG B.-H. & KATAGIRI S. (1992). Discriminative learning for minimum error classification. In *IEEE Trans. Acoustics, Speech, and Signal Processing*.
- KRUGER S. E., SCHAFFNER M., KATZ M., ANDELIC E. & WENDEMUTH A. (2006). Mixture of support vector machines for hmm based speech recognition. In *Proceedings of the 18th International Conference on Pattern Recognition*.
- LECUN Y., BOSER B., DENKER J. S., SOLLA S. A., HOWARD R. E. & JACKEL L. D. (1989). Back-propagation applied to handwritten zip code recognition. *Neural Computation*, 1, 541-551.
- LECUN Y., BOTTOU L., BENGIO Y. & HAFFNER P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- LI X., JIANG H. & LIU C. (2005). Large margin hmms for speech recognition. In *Proc. of ICASSP 2005*.
- L.R. B., P.F. B., DE SOUZA P.V. & MERCER R. (1986). Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *ICASSP*, p. 49-52.
- MORENO P. J., HO P. P. & VASCONCELOS N. (2004). A kullback-leibler divergence based kernel for svm classification in multimedia applications. In S. THRUN, L. SAUL & B. SCHÖLKOPF, Eds., *Advances in Neural Information Processing Systems 16*, Cambridge, MA : MIT Press.
- NADAS A. (1983). A decision-theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, p. 31(4) :814-817.
- NEAL R. & HINTON G. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. JORDAN, Ed., *Learning in Graphical Models* : Kluwer.
- NORMANDIN Y. (1991). Hidden markov models, maximum mutual information estimation, and the speech recognition problem. In *PhD dissertation, Dept. of Electrical Eng., McGill Univ., Montreal, Canada*.
- PLATT J. C. (1998). *Sequential Minimal Optimization : A Fast Algorithm for Training Support Vector Machines*. Rapport interne, Microsoft Research. 1998 John Platt.
- RATLIFF N., BAGNELL J. A. & ZINKEVICH M. (2007). Subgradient methods for maximum margin structured learning. In *AISTATS 2007*.
- SHA F. & SAUL L. K. (2006). Large margin gaussian mixture modeling for phonetic classification and recognition. In *Proc. of ICASSP 2006*.
- SHA F. & SAUL L. K. (2007). Large margin hidden markov models for automatic speech recognition. In *Advances in Neural Information Processing Systems 19*.
- TONG S. & KOLLER D. (2000). Restricted bayes optimal classifiers. In *AAAI/IAAI*, p. 658.
- VALTCHEV V., ODELL J., WOODLAND P. & YOUNG. S. (1997). Mmie training of large vocabulary recognition systems. In *Speech Communication*, p. 22 :303-314.
- VAPNIK V. N. (1999). *The Nature of Statistical Learning Theory*. Springer, 2 edition.