Sélection de variables pour la reconnaissance de formes

Sébastien GADAT[1], Laurent YOUNES[2]

[1]Centre de Mathématiques et de Leurs Applications
ENS de Cachan, 61 avenue du président Wilson, 94235 Cachan Cedex, France

[2]Center for Imaging Science
The Johns Hopkins University, 3400 N-Charles Street Baltimore MD 21218-2686, USA

sebastien.gadat@cmla.ens-cachan.fr
laurent.younes@cis.jhu.edu

Résumé – Nous étudions le problème d'extraction d'éléments informatifs d'un signal à partir d'un grand ensemble de variables descriptives $\mathcal{F}$, en préservant des capacités de classification optimales. Notre travail s'appuie sur une nouvelle approche sous-optimale du problème d'extraction de variables. Nous étudions une loi de probabilité sur $\mathcal{F}$ qui permet à la fois d'en extraire des éléments en mesurant leur efficacité, mais aussi d'optimiser un critère de performance. Nous donnons alors des algorithmes stochastiques de minimisation de ce critère et obtenons des améliorations notables de performance sur différents exemples.

Abstract – We introduce a new model formalizing the extraction of meaningfull features from a large set of variables $\mathcal{F}$ describing a signal or an image. Our work is based on a subobtimal study using a probability distribution $\mathbb{P}$ on $\mathcal{F}$. We can extract features with respect to this probability map and optimize the efficiency of the method of extraction. Meaningfull features are then detected by looking at the weights denoted by $\mathbb{P}$ on $\mathcal{F}$. We give stochastic algorithms to learn optimal distribution of weights and we apply our method to several problems like spam detection or face recognition.

1 Introduction

De nombreux problèmes de reconnaissance de formes dans des domaines aussi divers que l'interprétation de scène dans une image, de textes ou de *microarrays* génétiques engendrent la manipulation d'un très grand nombre de variables. Pour des raisons statistiques ou algorithmiques, la multiplication de ces variables descriptives nuit aux performances de classification. En effet, les variables peu informatives d'un signal agissent comme un bruit artificiel tandis que la variance statistique d'un échantillon de données augmente avec la dimension des signaux. Il s'agit donc de réduire la variance des signaux tout en préservant un biais faible sur les décisions prises par les algorithmes de classification [1]. Par ailleurs, certains algorithmes ne sont pas implémentables dans le cadre des grandes dimensions à la vue des temps de calcul qui leur sont alors nécessaires. De plus, le nombre d'échantillons nécessaire pour maintenir un niveau de précision donné croît exponentiellement avec le nombre de variables (piège des grandes dimensions [2]). Enfin, il peut être aussi important qu'optimiser des performances, en génétique par exemple, d'identifier les variables qui permettent l'obtention de ces performances.

Les méthodes telles que l'analyse en composantes principales ou indépendantes ne permettent pas d'identifier les variables importantes puisque celles-ci fournissent des espaces de petite dimension sans pour autant donner les variables les plus importantes pour le problème traité [3]. D'autres méthodes basées purement sur des tests statistiques (critère sur des p-valeurs par exemple) fournissent souvent des coordonnées explicatives sur les signaux sans pour autant assurer l'optimalité du jeu de variables sélectionnées.

Nous formalisons notre modèle de sélection de variables comme la recherche d'une distribution de poids optimale sur ces variables. Notre stratégie est sous-optimale, étant donnée une loi de probabilité sur l'ensemble des variables $\mathcal{F}$, nous munissons notre modèle d'une énergie que nous cherchons à minimiser. Nous donnons alors plusieurs algorithmes destinés à optimiser un tel critère avant d'appliquer notre étude à la détection de spam et à la reconnaissance de visages.

2 Modèle d'extraction de variables

2.1 Cadre statistique

Nous suivons le cadre statistique supervisé classique de la reconnaissance de formes, en utilisant un *Training Set* et un *Test Set*. Le signal I à classer est vu comme la réalisation d'une variable aléatoire et est décrit par des variables explicatives (questions, tests ...) notées $\mathcal{F} = \{\delta_1, \dots \delta_f\}$, on notera alors $\delta_i(I)$ ses coordonnées. L'objectif est alors de prédire parmi $\mathcal{C}_1, \dots \mathcal{C}_N$ la classe à laquelle appartient I. Il peut s'agir d'identifier par exemple si une image contient un visage, ou si un e-mail est en fait du spam. Nous supposons par ailleurs que nous disposons d'un algorithme $\mathbb{A}$ pour classer ces signaux, pouvant s'exécuter à la fois à partir de la totalité des variables $\delta(I)$ et sur un sous-ensemble de variables $\omega(I), \omega \subset \mathcal{F}$. Nous utiliserons ici les algorithmes de Support Vector Machine ou de

plus proches voisins. Ces algorithmes $\mathbb{A}$ permettent alors, pour chaque ω, de définir des indices de performance relatifs au problème de classification étudié. Par exemple, $q(\omega) = P[\mathbb{A}(\omega(I)) \neq \mathcal{C}(I)]$ désigne le taux d'erreur de classification renvoyé par $\mathbb{A}$ que nous utiliserons comme indice de pertinence de notre sous-ensemble ω. Nous estimons alors q par la probabilité empirique mesurée sur l'ensemble d'apprentissage de données :

$$\hat{q}(\omega) = \hat{P}[\mathbb{A}(\omega(I)) \neq \mathcal{C}(I)]$$

2.2 Modèle sous-optimal

Étant donné un entier k, nous cherchons à choisir le meilleur k-uplet de variables ω issues de $\mathcal{F}$ ayant une valeur optimale pour $\hat{q}$. Pour de simples raisons combinatoires, il est impossible de concevoir l'énumération de tous les k-uplets ω afin de déterminer pour chacun $\hat{q}(\omega)$ puisque si $\mathcal{F}$ contient 2000 variables et si $k = 50$, il est alors nécessaire d'exécuter C_{2000}^{50} algorithmes $\mathbb{A}$!

Afin de contourner ce problème combinatoire, nous adoptons une stratégie sous-optimale. Nous munissons l'ensemble des variables $\mathcal{F}$ d'une loi de probabilité $\mathbb{P}$, et pour un entier k quelconque fixé, notre modèle de sélection de variables consiste alors en k tirages de tests de $\mathcal{F}$ indépendants, avec remise, effectués selon $\mathbb{P}$. La loi sur ces k-uplets de $\mathcal{F}$ sera notée $\mathbb{P}^k$. Nous souhaitons alors minimiser l'erreur moyenne commise par $\mathbb{A}$ lorsque les sous-ensembles de k questions suivent la loi $\mathbb{P}^k$, cela nous amène naturellement à considérer l'énergie $\mathcal{E}$:

$$\mathcal{E}(\mathbb{P}) = \mathbb{E}_{\mathbb{P}^k}[q(\omega)] = \sum_{\omega \in \mathcal{F}^k} q(\omega)\mathbb{P}^k(\omega) \tag{1}$$

2.3 Estimation de q

Afin de calculer $\mathcal{E}$, nous devons être capable d'évaluer q. Nous utilisons la procédure suivante pour approcher $q(\omega)$ par $\hat{q}(\omega)$:

- Tirer un échantillon de données T_1 du *Training Set*.
- Éxécuter l'algorithme $\mathbb{A}_{\omega,T_1}$ sur les coordonnées ω et l'échantillon T_1.
- Tirer un échantillon T_2 sur le *Training Set* et mesurer l'erreur empirique de A_{ω,T_1} sur T_2.

L'objectif recherché, étant donné une telle méthode pour mesurer l'erreur empirique sur l'ensemble d'apprentissage, est de minimiser $\mathcal{E}$. Une distribution de poids optimale aura alors tendance à favoriser les tests δ privilégiés pour le problème de classification en chargeant de façon importante ces variables. En revanche, les « mauvaises » variables ne permettant pas à $\mathbb{A}$ d'obtenir de bonnes performances seront faiblement pondérées par $\mathbb{P}$. Nous quantifierons donc l'utilité d'une variable δ de $\mathcal{F}$ par la valeur $\mathbb{P}_\infty(\delta)$: plus $\mathbb{P}_\infty(\delta)$ est grande, plus δ est utile à la tâche de classification où $\mathbb{P}_\infty$ sera alors une distribution optimale.

3 Algorithmes de recherche

Nous cherchons donc à minimiser $\mathcal{E}$. Tout comme il est impossible de calculer toutes les valeurs de $q(\omega)$, il est impossible de se ramener à l'optimisation d'un polynôme de variables $\mathbb{P}$. Nous avons donc choisi d'approcher une distribution $\mathbb{P}$ optimale en utilisant une descente de gradient. L'algorithme d'apprentissage proposé suivra dès lors le schéma récursif résumé par la figure 1.

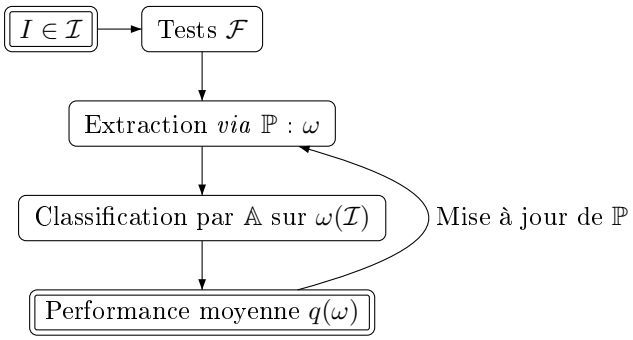

FIG. 1 – Schéma général de l'apprentissage de $\mathbb{P}$

3.1 Descente de gradient euclidienne et riemannienne

Métrique euclidienne L'optimisation de $\mathcal{E}$ se fait sur le simplexe de $\mathbb{R}^f$ noté $\mathcal{S}_\mathcal{F}$ et défini par

$$\sum_{\delta \in \mathcal{F}} \mathbb{P}(\delta) = 1 \qquad \text{et} \qquad \mathbb{P}(\delta) \geq 0$$

En métrique euclidienne, le gradient de $\mathcal{E}$ s'exprime en :

$$\forall \delta \in \mathcal{F} \qquad \nabla\mathcal{E}(\mathbb{P})(\delta) = \sum_{\omega \in \mathcal{F}^k} \frac{C(\omega,\delta)q(\omega)\mathbb{P}(\omega)}{\mathbb{P}(\delta)} \tag{2}$$

où $C(\omega,\delta)$ désigne simplement le nombre d'occurrences du test δ dans le k-uplet ω. Le schéma de descente correspondant est alors

$$\mathbb{P}_{n+1} = \mathbb{P}_n - \alpha_n \pi\left(\nabla\mathcal{E}(\mathbb{P})\right) \tag{3}$$

π désigne simplement la projection vectorielle d'un vecteur sur l'hyperplan de $\mathbb{R}^f$ portant $\mathcal{S}_\mathcal{F}$.

Métrique riemannienne La principale difficulté d'une approche basée sur (3) est que la contrainte de positivité des poids $\mathbb{P}_n$ est difficile à satisfaire. Il est donc naturel de choisir une paramétrisation de $\mathcal{S}_\mathcal{F}$ en variables exponentielles. En posant $y = \log(\mathbb{P})$, on obtient une nouvelle énergie $\tilde{\mathcal{E}}(y) = \mathcal{E}(\mathbb{P})$ et la descente de gradient sur y correspondante s'écrit :

$$y_{n+1} = y_n - \alpha_n \overline{\pi}\left(\nabla\tilde{\mathcal{E}}(y)\right)$$

Cette descente se traduit alors sur les poids par une équation de la forme

$$\mathbb{P}_{n+1} = \mathbb{P}_n \frac{e^{\alpha_n \nabla\tilde{\mathcal{E}}(y_n)}}{K_n} \tag{4}$$

où K_n est une constante de renormalisation positive assurant la normalisation de la somme des poids à 1. Cette paramétrisation permet donc de satisfaire la contrainte de positivité sur $\mathbb{P}_n$ naturellement. Il est possible d'interpréter une telle propriété en remarquant qu'en munissant $\mathcal{S}_\mathcal{F}$ de la métrique riemannienne de Kullback sur les lois de probabilité (*i.e.* la métrique du χ_2), on a alors la proposition suivante.

Proposition 3.1 (Gradient riemannien) *Pour toute distribution de poids strictement positive de $\mathcal{S}_\mathcal{F}$, on a*

$$\nabla_{\chi_2}\mathcal{E}(\mathbb{P}) = \nabla\tilde{\mathcal{E}}(y)$$

Il est ainsi aisé d'interpréter la descente en variables exponentielles comme étant une descente sur les variables $\mathbb{P}$ avec le produit scalaire dérivé de la distance du χ_2. Cette distance a pour propriété de mettre à distance « infinie » un point sur la frontière du simplexe d'un point strictement à l'intérieur.

3.2 Diffusion réfléchie

La descente de gradient riemannienne précédente permet de définir structurellement la positivité des poids, mais son évolution n'est pas aussi rapide que l'évolution euclidienne (paramétrisation exponentielle de petites quantités entre 0 et 1). Afin de garder la vitesse d'évolution de la méthode euclidienne et la stabilité structurelle de $\mathcal{S}_\mathcal{F}$, il est donc judicieux d'utiliser un processus de diffusion réfléchie dans $\mathcal{S}_\mathcal{F}$ avec terme de dérive dirigé par le gradient euclidien de $\mathcal{E}$.

$$\mathrm{d}\mathbb{P}_t = -\pi\left(\nabla\mathcal{E}(\mathbb{P}_t)\right) + \sqrt{\Sigma}\mathrm{d}W_t + \mathrm{d}Z_t \qquad (5)$$

Σ est alors une matrice non-dégénérée dans $\mathcal{S}_\mathcal{F}$, W_t un mouvement brownien standard alors que $\mathrm{d}Z_t$ est un processus de rappel sur la face la plus proche du simplexe lorsque le terme $(\nabla\mathcal{E}(\mathbb{P}_t)) + \sqrt{\Sigma}\mathrm{d}W_t$ amène le processus hors de $\mathcal{S}_\mathcal{F}$.

L'existence et l'unicité de (5) sont assurées dans notre cas et dépendent en réalité de la construction de l'application de Skorokhod [4]. C'est ce processus que nous simulerons lors de nos expériences pour faire diminuer $\mathcal{E}$.

3.3 Approximation stochastique

En étudiant l'expression (2) du gradient de $\mathcal{E}$, on constate que pour implémenter numériquement une équation de la forme (5), il faut évaluer le terme $\nabla\mathcal{E}(\mathbb{P}_n)$ à chaque itération, terme qui nécessite une trop grande énumération de $\hat{q}(\omega)$ pour être réaliste en temps de calculs. Afin de pallier cette difficulté numérique insurmontable de manière exacte, il est possible de mettre en place un algorithme d'approximation stochastique dès qu'on remarque que

$$\nabla\mathcal{E}(\mathbb{P})(\delta) = \mathbb{E}\left[\frac{q(\omega)C(\omega,\delta)}{\mathbb{P}(\delta)} \,|\, \omega\sim\mathbb{P}^k\right] \qquad (6)$$

L'apprentissage stochastique suivant permet alors de simuler asymptotiquement (3) ou (5) [5, 6, 7].

- Initialiser $\mathbb{P}_0$ à la loi uniforme sur l'ensemble des variables $\mathcal{F}$.
- Pour calculer $\mathbb{P}_{n+1}$, extraire ω_n selon $\mathbb{P}_n^k$ et mettre à jour les poids *via* :

$$\mathbb{P}_{n+1} = \mathbb{P}_n - \alpha_n d_n + \sqrt{\alpha_n}\mathrm{d}\xi_n + \mathrm{d}z_n \qquad (7)$$

avec la définition suivante du vecteur d_n :

$$d_n(\delta) = \frac{q(\omega)C(\omega_n,\delta)}{\mathbb{P}_n(\delta)} \qquad (8)$$

3.4 Dynamique d'apprentissage

L'hypothèse fondamentale d'espérance (6) étant également satisfaite pour le terme d_n, on peut alors démontrer ([8]) le théorème fondamental suivant :

Théorème 3.2 (Asymptotique de (7)) *Le processus discret donné par (7) est une trajectoire pseudo-asymptotique de (5). De plus, le processus continu (5) est récurrent positif et les processus continu et discret convergent vers l'unique mesure stationnaire μ de (5) qui est le champ de Gibbs associé à l'énergie $\mathcal{E}$, donné par*

$$\mu(\mathbb{P}) = \frac{e^{-\mathcal{E}(\mathbb{P})}}{Z}$$

Ce théorème assure la convergence des simulations numériques vers des états d'énergie faible, et contourne la difficulté de la simulation exacte de (5).

4 Expériences

Nous implémentons notre algorithme de sélection de variables par apprentissage de poids optimaux sur deux exemples classiques de classification.

4.1 Reconnaissance de visages

Dans le cas de la détection de visages, nous utilisons la base de donnéees fournies par le MIT, des imagettes de taille 19×19, centrées codées sur 8 bits en niveaux de gris.

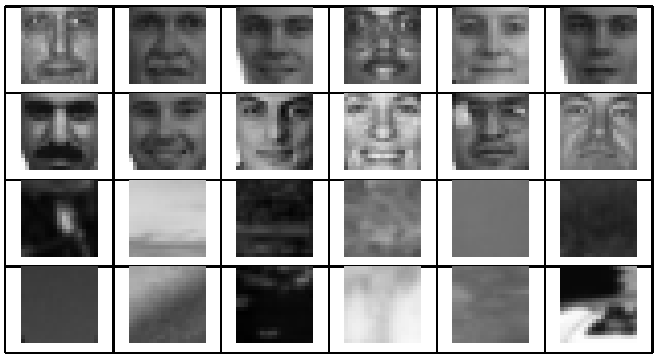

FIG. 2 – Échantillon issu de la base de données

Les variables utilisées sont des détecteurs de bords orientés, binaires décrits dans [9], très rapides à calculer. L'algorithme $\mathbb{A}$ est un algorithme de séparation par hyperplan de *Support Vector Machine* linéaire. Le nombre de variables initiales préselectionnées est de l'ordre de 2000 alors que nous cherchons à ne retenir que quelques dizaines de tests pour obtenir des performances de classification correctes.

La figure 3 suivante représente l'évolution, en fonction du nombre k, du gain apporté par une sélection des variables par une loi optimale plutôt que par la loi uniforme $\mathcal{U}_\mathcal{F}$.

On note une sensible amélioration des performances après apprentissage d'une loi d'extraction, surtout lorsque le nombre de variables à retenir est faible. La performance finale obtenue est 1.6% d'erreur sur le *Test Set* avec (5). Enfin, les détecteurs de bords privilégiés (*i.e.* avec un poids

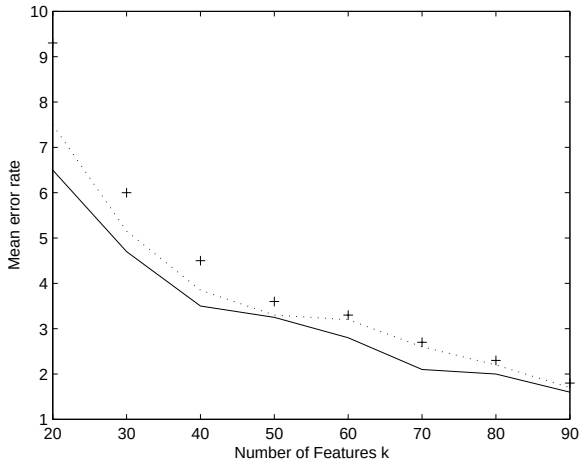

FIG. 3 – Erreur moyenne sur les données Tests avec des extractions par loi uniforme (+) et loi limite $\mathbb{P}_\infty$ apprises par descente riemannienne (pointillés) et diffusion réfléchie (ligne continue) en fonction de $k = |\omega|$.

$\mathbb{P}_\infty$ important) sont localisés sur les zones intuitivement informatives de l'image : bouche, nez, contour du visage comme le représente la figure 4.

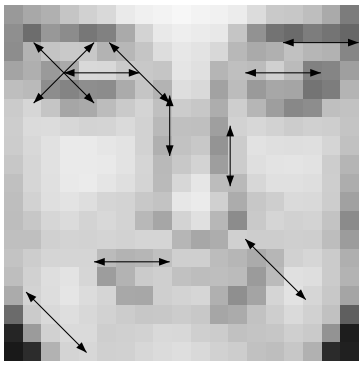

FIG. 4 – Représentation des détecteurs de bords privilégiés après apprentissage.

4.2 Détection de spam

Nous évaluons également nos algorithmes sur des données fournies par l'UCI de messagerie électronique. Il s'agit de messages étiquetés (spam/non spam) séparés en ensembles d'apprentissages et de tests, issus d'une même boîte mail.

Les variables utilisées mesurent la fréquence d'apparition de 54 mots pré-définis dans un e-mail pour former $\mathcal{F}$. $\mathbb{A}$ est cette fois un algorithme de 4-plus proches voisins. La figure 5 représente alors l'évolution de l'erreur moyenne de l'algorithme $\mathbb{A}$ avec le temps.

Là encore, l'algorithme utilisant la diffusion réfléchie semble plus performant que la simple descente de gradient en variables exponentielles. Les mots privilégiés lors de nos expériences sont par ordre décroissant *remove, business, receive, internet, ...* pour le spam et *cs, 857, 415, project, ...* pour les « vrais e-mails ». Si les mots associés au spam nous sont familiers, les mots relatifs aux « vrais e-mails » sont en réalité relatifs au caractère personnel de l'auteur de cette base de données (métier, coordonnées ...). Enfin, avec un vote de 10 détecteurs utilisant 15 mots extraits selon $\mathbb{P}_\infty$, nous obtenons 7.5% d'erreur, la meilleure performance sur cette base étant de 7% d'erreur.

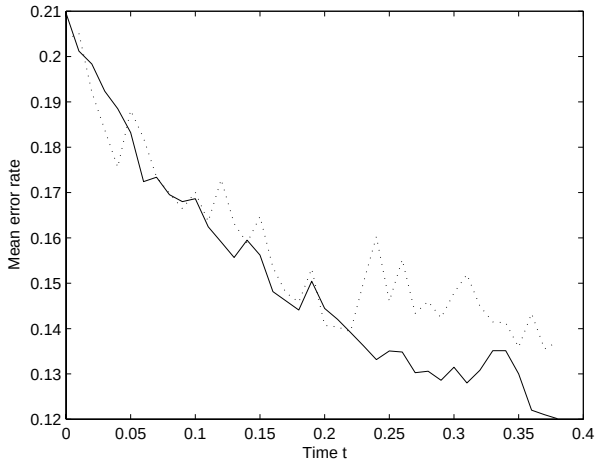

FIG. 5 – Erreur en fonction du temps obtenue par descente riemannienne et diffusion réfléchie (pointillés).

Des travaux futurs généraliseront cette approche en permettant à $\mathcal{F}$ d'évoluer en utilisant une dynamique inspirée des algorithmes génétiques (suppression et composition de variables).

Références

[1] S. Geman, E. Bienenstock and R. Doursat. *Neural networks and the bias/variance dilemma*, Neural Computation, vol. 4, pp. 1-58, 1992.

[2] R. Bellman. *Adaptative Control Processes : A guided Tour*, Princeton University Press, 1961.

[3] C. Jutten and J. Hérault. *Blind separation of sources, part 1 : An adaptative algorithm bases on neuromimetic architecture*, Signal Processing, vol. 24, pp. 1-10, 1991.

[4] P. Dupuis and H.Ishii. *On Lipschitz continuity of the solution mapping to the Skorokhod problem, with applications*, Stochastics Stochastics Rep., vol. 35, no. 1,pp. 31-62,1991.

[5] S. Gadat and L. Younes. *A stochastic algorithm of features extraction for pattern recognition*, Preprint CMLA, pp. 1-22, 2004.

[6] H. Kushner and G. Yin. *Stochastic approximation and recursive algorithms and applications*, Springer-Verlag, 2003.

[7] M. Benaim. *A dynamical system approach to stochastic approximations*, SIAM Control Optim., vol. 34, no. 2, pp. 437-472, 1996.

[8] S. Gadat. *Jump diffusion over Feature space for object recognition*, Preprint CMLA, pp. 1-27, 2005.

[9] Y. Amit and D. Geman. *A computational model for visual selection*, Neural Computation, vol. 11, pp. 1691-1715, 1999.