Classifieurs Probabilistes Parcimonieux

Romain Hérault, Yves Grandvalet

Classifieurs Probabilistes Parcimonieux

Romain HÉRAULT (1), Yves GRANVALET (2)

(1) Laboratoire LITIS, EA 4108

INSA de Rouen, Avenue de l'Université - BP 8

76801 Saint-Étienne-du-Rouvray Cedex

romain.herault@insa-rouen.fr

(2) HEUDIASYC, UMR CNRS 6599

Université de Technologie de Compiègne

BP 20529, 60205 Compiègne cedex, France

yves.grandvalet@hds.utc.fr

6 novembre 2008

Résumé

Les scores retournés par les séparateurs à vaste marge sont souvent utilisés comme mesures de confiance pour la classification de nouveaux exemples. Cependant, il n'y a pas de fondement théorique à cette pratique. C'est pourquoi, lorsque l'incertitude de classification doit être estimée, il est plus sûr de recourir à des classifieurs qui estiment les probabilités conditionnelles des classes. Ici, nous nous concentrons sur l'ambiguïté à proximité de la frontière de décision. Nous proposons une adaptation de l'estimation par maximum de vraisemblance.

Le critère proposé vise à estimer les probabilités conditionnelles, de manière précise à l'intérieur d'un intervalle défini par l'utilisateur, et moins précise ailleurs. Le modèle est aussi parcimonieux, dans le sens où peu d'exemples contribuent à la solution. Nous appliquons ce critère à la régression logistique. Ce modèle de régression logistique parcimonieuse sera ensuite validé par le jeu de données *Forest Covertype* de l'UCI.

Mots-Clés

Apprentissage statistique, Classifieur parcimonieux, Classes déséquilibrées.

1 Introduction

Lorsqu'il existe une vaste majorité d'exemples négatifs « non intéressants », et seulement peu d'exemples appartenant à la classe positive, l'apprentissage a tendance à biaiser ses résultats en faveur de la classe dominante. Ce biais peut être

traité en rééquilibrant la distribution d'apprentissage [Ting, 2000, Elkan, 2001] : les exemples de la classe minoritaire peuvent être répliqués ou générés artificiellement, ou un certain nombre d'exemples de la classe majoritaire peuvent être éliminés. Cependant, l'augmentation du nombre d'exemples de la classe minoritaire aboutit à un calcul inefficace et la réduction de la classe majoritaire peut amener à l'élimination d'informations importantes pour la classification. Le problème d'estimation de règle de décision sur des classes déséquilibrées, qui motive ce travail, doit pouvoir tirer profit d'un classifieur parcimonieux qui permet l'estimation précise des probabilités sur un intervalle d'intérêt.

Le séparateur à vaste marge (SVM) est un modèle parcimonieux des plus répandus. Plusieurs tentatives visent à transformer le score SVM en une estimation de probabilité [Platt, 2000, Grandvalet et al., 2006]. Cependant, rien ne garantit que le score SVM ne reflète une confiance. Au contraire, Bartlett et Tewari [2007] ont démontré que les probabilités conditionnelles des classes ne peuvent être retrouvées sans ambiguïté que sur la frontière de décision. Si ces probabilités doivent être évaluées, il est donc préférable d'utiliser des classifieurs probabilistes, comme la régression logistique, qui estiment, directement, les probabilités conditionnelles de façon consistante.

Nous proposons de construire un classifieur probabiliste qui soit précis sur une « zone grise », là où les étiquettes des classes changent. L'incertitude de classification est quantifiable dans cette zone, puisque les probabilités y sont bien calibrées. Ce classifieur permet ainsi d'obtenir des règles de décision callibrées pour différents coûts associés aux fausses alarmes et aux absences de détection. Se concentrer sur un petit intervalle de probabilités conditionnelles a deux avantages : premièrement, l'objectif de l'apprentissage se rapproche de la minimisation du risque de mauvais classement qui est l'objectif final ; deuxièmement, l'imprécision des probabilités conditionnelles en dehors de l'intervalle d'intérêt est un élément clé de l'efficacité des méthodes à noyau. En effet, Bartlett et Tewari [2007, Corollaire 4] prouvent que, si les probabilités conditionnelles peuvent être estimées partout de manière précise, alors, les modèles à noyau ne peuvent être parcimonieux.

Une méthode à noyau est qualifiée de parcimonieuse si un nombre limité d'exemples d'apprentissage participe à l'estimation, c'est-à-dire, si un nombre important d'exemples d'apprentissage n'intervienne pas dans le calcul de la règle de décision apprise. La parcimonie engendre donc des calculs plus efficaces.

Notre méthode associe des fonctions de pertes différentes pour les exemples positifs et négatifs. En celà, elle ressemble aux méthodes qui pondérent différemment les exemples positifs et négatifs [Osuna et al., 1997]. Elle est cependant plus générale, en ce sens qu'elle retourne une solution bien calibrée pour un intervalle de pondérations.

2 Critère d'apprentissage

Nous avons un ensemble d'apprentissage $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, où chaque exemple est décrit par ses caractéristiques $\boldsymbol{x}_i$ et son étiquette $y_i \in \{-1, 1\}$. En supposant l'indépendance des exemples, l'estimation de $p(Y = y|\boldsymbol{x})$ peut être réalisée par la maximisation de la log-vraisemblance conditionnelle,

$$\sum_{i:y_i=1} \log(\widehat{p}(Y_i = 1|\boldsymbol{x}_i)) + \sum_{i:y_i=-1} \log(1 - \widehat{p}(Y_i = 1|\boldsymbol{x}_i)) \,, \tag{1}$$

où $\widehat{p}(Y_i = y_i|\boldsymbol{x}_i)$ est l'estimateur de $p(Y_i = y_i|\boldsymbol{x}_i)$. Cette fonction de perte est représentée Figure 1.

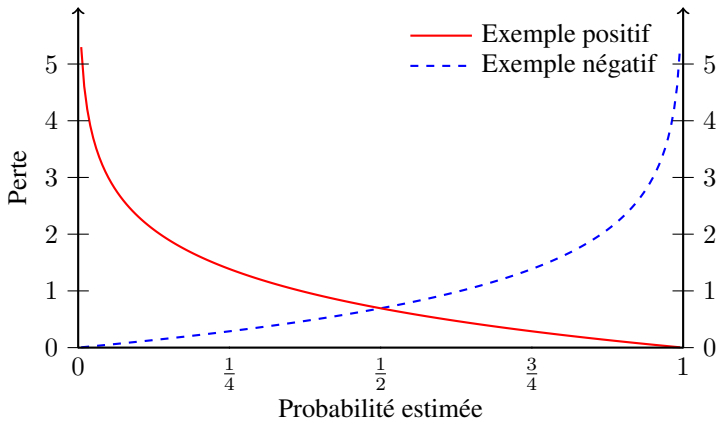

FIG. 1: Fonction de perte de la log-vraissemblance.

La règle de décision de Bayes est définie par les probabilités conditionnelles $p(Y = y|\boldsymbol{x})$, et les coûts de mauvais classement. Les deux types d'erreurs sont

- les faux positifs occasionnant une perte c_- ;
- les faux négatifs occasionnant une perte c_+.

La règle est alors,

$$\text{Décider } +1 \text{ pour } \boldsymbol{x} \text{ ssi } p(Y = 1|\boldsymbol{x}) \geq \frac{c_-}{c_+ + c_-} \quad , \tag{2}$$

Bien que la règle de décision de Bayes soit définie par $p(Y = y|\boldsymbol{x})$, elle n'a pas besoin d'un estimateur précis sur l'ensemble du domaine des probabilités : il suffit d'estimer les probabilités conditionnelles en $\frac{c_-}{c_+ + c_-}$, qui définit la frontière de décision. Asymptotiquement, c'est ce que réalisent les SVM [Bartlett et Tewari, 2007] pour $p(Y = y|\boldsymbol{x}) = 0.5$, ou pour d'autres probabilités lorsque le critère est asymétrique [Osuna et al., 1997].

La maximisation de la log-vraisemblance (1) permet d'estimer les probabilités conditionnelles sur tout l'intervalle $[0, 1]$. Notre objectif est moins ambitieux : estimer des probabilités conditionnelles sur un intervalle $[p_{\min}, p_{\max}]$. Au-delà de cet intervalle, nous voulons juste savoir si $p(Y = y|\boldsymbol{x})$ est plus petit que $p_{\min}$ ou plus grand que $p_{\max}$. Pour arriver à nos fins, nous proposons de maximiser

$$\begin{aligned} &\sum_{i:y_i=1} \log\left(\min\left(\widehat{p}(Y_i = 1|\boldsymbol{x}_i), p_{\max}\right)\right) + \\ &\qquad \sum_{i:y_i=-1} \log\left(\min\left(1 - \widehat{p}(Y_i = 1|\boldsymbol{x}_i), 1 - p_{\min}\right)\right) \quad . \end{aligned} \tag{3}$$

Ce critère, que nous appellerons vraisemblance locale, est concave en $\widehat{p}(Y_i = 1|\boldsymbol{x}_i)$. Il est calibré [Bartlett et al., 2004] pour toutes les règles de décision basées sur une paire de coûts (c_+, c_-) telle que

$$\pi^+ \in [p_{\min}, p_{\max}] \text{ où } \pi^+ = \frac{c_-}{c_+ + c_-} \quad .$$

Réciproquement, en choisissant $p_{\min} < \pi^+$ et $p_{\max} > \pi^+$, la fonction de perte, représentée sur la Figure 2, rempli les conditions nécessaires et suffisantes de calibration basées sur la convexité de la fonction de perte et son gradient en π^+

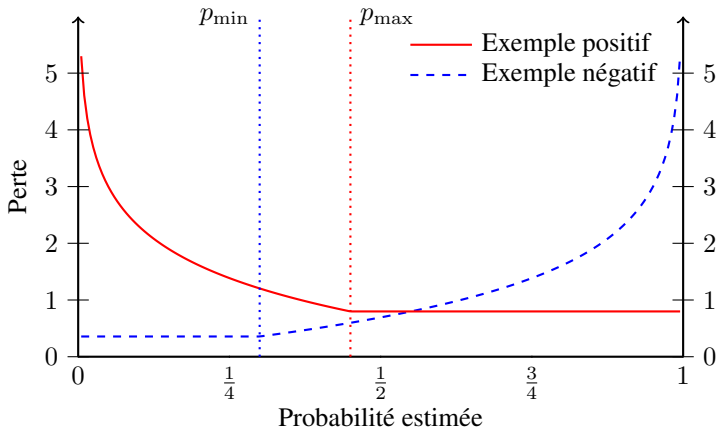

FIG. 2: Fonction de perte de la log-vraisemblance locale.

[Bartlett et al., 2004, Lemme 5]. [1] Ainsi, si la capacité du classifieur est bien contrôlée, les probabilités conditionnelles estimées convergent vers les vraies probabilités sur l'intervalle $[p_{\min}, p_{\max}]$.

La parcimonie du classifieur résulte des propriétés du sous-différentiel de la fonction de perte, et plus particulièrement du domaine sur lequel ce sous-différentiel comprend 0. Des conditions analogue à (A1) et (A2) dans [Bartlett et Tewari, 2007], permettent de dériver des bornes inférieures sur la fraction de vecteur supports [Bartlett et Tewari, 2007, Théorème 3]. Asymptotiquement, tous les exemples positifs tels que $p(Y = 1|\boldsymbol{x}) > p_{\max}$ et tous les exemples négatifs tels que $p(Y = 1|\boldsymbol{x}) < p_{\min}$ n'interviennent pas dans le calcul de l'estimateur.

3 Application à la régression logistique

La régression logistique est un modèle probabiliste classique qui considère que le log-ratio des probabilités conditionnelles est linéaire,

$$\log \frac{\widehat{p}(Y = 1|\boldsymbol{x})}{1 - \widehat{p}(Y = 1|\boldsymbol{x})} = \mathbf{w}^\intercal \boldsymbol{x} + b \ , \tag{4}$$

et où les coefficients $(\mathbf{w}, b)$ sont estimés par la maximisation de la vraisemblance (1), éventuellement pénalisée.

Des non-linéarités peuvent être introduites dans la régression logistique grâce aux fonctions noyaux, en définisssant le log-ratio comme suit :

$$\log \frac{\widehat{p}(Y = 1|\boldsymbol{x})}{1 - \widehat{p}(Y = 1|\boldsymbol{x})} = f(\boldsymbol{x}) + b \ ,$$

où f est une fonction appartenant à un espace de Hilbert à noyau reproduisant $\mathcal{H}$.

Avec ce choix, le critère d'apprentissage doit généralement incorporer une pénalisation pour se prémunir du sur-apprentissage [Roth, 2001, Zhu et Hastie, 2001]. La vraisemblance (1) pénalisée par la norme de f s'écrit

$$\sum_{i=1}^{n} \log \left(1 + e^{-y_i(f(\boldsymbol{x}_i)+b)}\right) + \frac{\lambda}{2}||f||^2_{\mathcal{H}} \ , \tag{5}$$

où λ est un hyper-paramètre, qui peut être ajusté par validation croisée.

La régression logistique à noyau peut être entraînée dans l'espace des variables primales en utilisant la méthode de

[1] Bartlett et al. [2004] et Bartlett et Tewari [2007] se focalisent sur le cas symmétrique, mais il est aisé de transposer leurs résultats au cas général.

Newton [Roth, 2001], ou dans l'espace des variables duales [Keerthi et al., 2005].

Contrairement aux SVMs, la régression logistique n'a pas de solution parcimonieuse, dans le sens où tous les exemples participent à la solution. Notre méthode consiste à remplacer le terme de log-vraisemblance par sa version locale (3). Le critère devient alors

$$\sum_{i=1}^{n} \log\left(1 + e^{\max(-y_i(f(\boldsymbol{x}_i)+b),\mathrm{f}_i)}\right) + \tfrac{\lambda}{2}||f||^2_{\mathcal{H}} \ , \tag{6}$$

$$\text{où} \quad \begin{cases} \mathrm{f}_i = -\log \frac{p_{\max}}{1-p_{\max}} & \text{si} \quad y_i = 1 \ ; \\ \mathrm{f}_i = \log \frac{p_{\min}}{1-p_{\min}} & \text{si} \quad y_i = -1 \ . \end{cases}$$

Nous appelerons l'estimateur ainsi obtenu « régression logistique parcimonieuse à noyau ». C'est du terme d'ajustement, et non pas du terme de pénalisation, que provient la parcimonie, qui résulte de la troncature de la fonction de perte. Les exemples d'apprentissage avec une grande valeur de $y_i f(\boldsymbol{x}_i)$ ne participent pas au classifieur final.

3.1 Apprentissage dans l'espace des variables primales

Par souci de simplicité, nous considérons ici le modèle de régression linéaire, sans régularisation, où le terme de biais b est inclus ici dans $\boldsymbol{w}$, en supposant que le vecteur $\boldsymbol{x}$ inclut un terme constant.

Nous rappelons tout d'abord la mise à jour de Newton-Raphson pour la régression logistique standard maximisant la vraisemblance,

$$\boldsymbol{w}^{(s+1)} = \boldsymbol{w}^{(s)} + \rho\,(\mathbf{X}^\intercal \mathbf{D}^{(s)})\mathbf{X})^{-1}\mathbf{X}^\intercal \mathbf{z}^{(s)} \ ,$$

où

- $\boldsymbol{w}^{(s)}$ est le vecteur de paramètres à l'étape s ;
- $\mathbf{z}^{(s)} = (z_1, ..., z_n)$ et $z_i = \frac{y_i}{1+\exp(y_i \boldsymbol{w}^{(s)\intercal}\boldsymbol{x}_i)}$;
- $\mathbf{X} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_n]^\intercal$
- $\mathbf{D}$ est une matrice diagonale $n \times n$ de terme général $D_{ii} = \widehat{p}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}^{(s)})(1 - \widehat{p}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}^{(s)})$;
- ρ est le pas de descente, fixé habituellement à 1.

La fonction de perte de la vraisemblance locale (3) n'est pas continûment dérivable en $\boldsymbol{w}$ pour :

(i) les exemples positifs pour lesquels $\widehat{p}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}) = p_{\max}$;

(ii) les exemples négatifs pour lesquels $\widehat{p}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}) = p_{\min}$.

Elle est nulle pour :

(iii) les exemples positifs pour lesquels $\widehat{p}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}) > p_{\max}$;

(iv) les exemples négatifs pour lesquels $\widehat{p}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}) < p_{\min}$.

La contribution des autres exemples dans le calcul des dérivées du premier et du second ordre a la même forme que pour la régression logistique.

Dans le cadre de l'optimisation des SVM dans le primal, Chapelle [2007] propose de minimiser un critère plus régulier,

où chaque « coude » est remplacé par par une fonction deux fois dérivable (en l'occurence des polynômes). On peut alors utiliser la méthode de Newton-Raphson, dont la forme est alors très similaire à celle du critère de vraisemblance, si ce n'est que les exemples des catégories (iii) et (iv) n'y interviennent pas. Cependant, même lissées, les discontinuités restent susceptibles d'amener des problèmes de stabilité. Pour les éviter, nous limitons le pas ρ de telle manière que les exemples positifs dont la perte est saturée ne puissent pas produire une réponse inférieure à un seuil, $\widehat{p}_t$, et réciproquement, que les exemples négatif dont la perte est saturée ne puisse pas produire une réponse supérieure à $\widehat{p}_t$. Ce seuil $\widehat{p}_t$ est fixée à $\frac{p_{\min}+p_{\max}}{2}$. L'annexe A détaille le calcul du pas ρ.

3.2 Apprentissage dans l'espace des variables duales

Chapelle [2007] discute de la prévalence des algorithmes duaux pour les machines à noyau. Cependant, l'optimisation dans l'espace des variables duales tire un grand avantage de la parcimonie provenant de la partie constante du coût, alors que cette dernière cause des difficultés avec les méthodes du second ordre dans la formulation primale. Pour le type d'application que nous voulons traiter, avec un grand déséquilibre entre les classes, on s'attend à ce que la plupart des exemples de la classe majoritaire soient absents de la solution, ainsi, on peut espèrer une optimisation plus efficace dans l'espace des variables duales.

3.2.1 Principe

Nous proposons un algorithme de contraintes actives, suivant une stratégie qui a déjà fait preuve de son efficacité pour les SVM [Loosli et al., 2005]. L'algorithme SimpleSVM [Vishwanathan et al., 2003, Loosli et al., 2005] résout le problème d'apprentissage des SVM par une approche de contraintes actives. La répartition des exemples dans l'ensemble des vecteurs support et non-support étant supposée connue, le critère d'apprentissage est optimisé. De l'optimisation résulte une nouvelle partition des exemples en vecteurs support (ensemble actif) et non-support (ensemble inactif). Ces deux étapes sont répétées jusqu'à ce qu'un certain niveau de précision soit atteint.

Nous utilisons la même stratégie. Nous présentons, dans un premier temps, la formulation duale de la régression logistique parcimonieuse. Considérant que la partition entre exemples actifs et inactifs est correcte, nous en déduisons la mise à jour optimale des paramètres. Puis, nous montrons comment mettre à jour l'ensemble actif en se basant sur la réactualisation des paramètres.

3.2.2 Formulation duale

Comme dans la formulation duale des SVM, nous évitons la discontinuité introduite par la fonction $\max$ dans (6) grâce à l'introduction de variables d'écart $\boldsymbol{\xi}$

$$\min_{f,\boldsymbol{\xi},b} \quad \frac{\lambda}{2}||f||^2_{\mathcal{H}} + \sum_{i=1}^{n} \log(1+e^{\xi_i}) \tag{7a}$$

$$\text{t. q.} \qquad \xi_i \geq -y_i(f(\boldsymbol{x}_i)+b) \qquad i=1,\ldots,n \tag{7b}$$

$$\xi_i \geq \mathrm{f}_i \qquad i=1,\ldots,n \ , \tag{7c}$$

où f_i est définie comme en (6).

Le lagrangien de ce problème convexe est

$$\begin{aligned}\mathcal{L}_{\text{primal}} &= \frac{\lambda}{2}||f||^2_{\mathcal{H}} + \sum_{i=1}^{n}\log(1+e^{\xi_i}) \\ &\quad - \sum_{i=1}^{n}\alpha_i[y_i(f(\boldsymbol{x}_i)+b)+\xi_i] + \sum_{i=1}^{n}\beta_i(\mathrm{f}_i-\xi_i) \ . \end{aligned} \tag{8}$$

La solution de (7) est atteinte au point col du lagrangien (8). Les conditions de Kuhn-Tucker impliquent

$$\nabla_f\mathcal{L}_{\text{primal}} = 0 \quad \Leftrightarrow \quad \lambda f(\boldsymbol{x}) - \sum_{i=1}^{n}\alpha_i y_i K(\boldsymbol{x},\boldsymbol{x}_i) = 0 \ , \tag{9a}$$

$$\frac{\partial\mathcal{L}_{\text{primal}}}{\partial b} = 0 \quad \Leftrightarrow \quad -\sum_{i=1}^{n}\alpha_i y_i = 0 \ , \tag{9b}$$

$$\frac{\partial\mathcal{L}_{\text{primal}}}{\partial \xi_i} = 0 \quad \Leftrightarrow \quad \frac{1}{1+e^{-\xi_i}} - (\alpha_i+\beta_i) = 0 \ . \tag{9c}$$

où $K(\cdot,\cdot)$ est le noyau reproduisant de l'espace de Hilbert $\mathcal{H}$.

Grâce à ces conditions, nous pouvons éliminer f et $\boldsymbol{\xi}$ du lagrangien

$$\begin{aligned}\mathcal{L}_{\text{primal}} &= -\frac{1}{2\lambda}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_j,\boldsymbol{x}_i) + \sum_{i=1}^{n}\beta_i\mathrm{f}_i \\ &\quad - \sum_{i=1}^{n}(\alpha_i+\beta_i)\log(\alpha_i+\beta_i) + (1-\alpha_i-\beta_i)\log(1-\alpha_i-\beta_i) \ , \end{aligned} \tag{10}$$

Cette expression fait intervenir $2n$ variables. Nous pouvons réduire ce nombre de variables si nous connaissons pour chaque exemple les contraintes actives (7b) et/ou (7c) de l'équation (7).

3.2.3 Partitionnement de l'ensemble d'apprentissage

Nous séparons l'ensemble d'apprentissage en trois ensembles suivant les contraintes actives en (7b) et (7c). Tout d'abord, il faut remarquer qu'au moins une de ces contraintes est nécessairement active, puisque la fonction objectif (7a) décroit quand ξ_i décroit. Soit $I = \{1,\ldots,n\}$ l'index des exemples d'apprentissage, nous le partageons en trois parties $I = \{I_0, I_h, I_\ell\}$. Pour les exemples indexés par :

- I_0, seule la contrainte (7c) est active, $\xi_i = \mathrm{f}_i$, ce qui implique $\alpha_i = 0$;
- I_h, les deux contraintes sont actives, c'est à dire qu'aucun multiplicateur de Lagrange associé n'est nul, et que $\xi_i = \mathrm{f}_i = -y_i(f(\boldsymbol{x}_i)+b)$;
- I_ℓ, seule la contrainte (7b) est active, c'est à dire que $\beta_i = 0$ et $\xi_i = -y_i(f(\boldsymbol{x}_i)+b)$.

De ces définitions, il découle que :

- I_0 regroupe les exemples situés dans la partie constante de la fonction de perte où $y_i(f(\boldsymbol{x}_i)+b) \geq -\mathrm{f}_i$;
- I_h regroupe les exemples situés sur le point charnière de la fonction de perte où les deux contraintes peuvent être actives puisque $y_i(f(\boldsymbol{x}_i)+b) = -\mathrm{f}_i$;
- I_ℓ regroupe les exemples situés dans la partie logarithmique de la fonction de perte où $-y_i(f(\boldsymbol{x}_i)+b) \leq -\mathrm{f}_i$.

La Table 1 résume les propriétés communes aux exemples appartenant à chaque ensemble, à savoir les valeurs de la variable primale ξ_i et des variables duales α_i et β_i. Les valeurs non-nulles de α_i et β_i sont déduites de la condition (9c), où ξ_i est remplacée par sa valeur.

Ensemble	ξ_i	α_i	β_i
I_0	f_i	0	$\frac{1}{1+e^{-\mathrm{f}_i}}$
I_h	f_i	$\frac{1}{1+e^{-\mathrm{f}_i}} - \beta_i$	$\frac{1}{1+e^{-\mathrm{f}_i}} - \alpha_i$
I_ℓ	$-y_i(f(\boldsymbol{x}_i)+b)$	$\frac{1}{1+e^{y_i(f(\boldsymbol{x}_i)+b)}}$	0

TAB. 1: Valeurs de ξ_i, α_i et β_i pour les exemples indexés par $\{I_0, I_h, I_\ell\}$.

Supposons que la répartition $\{I_0, I_h, I_\ell\}$ soit connue. Le lagrangien se décompose comme suit :

$$\mathcal{L}_{\mathrm{primal}} = -\frac{1}{2\lambda}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_j,\boldsymbol{x}_i) + \mathcal{L}_0 + \mathcal{L}_h + \mathcal{L}_\ell \ .$$

où en substituant les expressions de α_i et β_i par celles de la Table 1 nous avons :

$$\begin{aligned}
\mathcal{L}_0 &= \sum_{i\in I_0}\log(1+e^{\mathrm{f}_i}) \ , \\
\mathcal{L}_h &= \sum_{i\in I_h}\log(1+e^{\mathrm{f}_i}) - \sum_{i\in I_h}\alpha_i\mathrm{f}_i \ , \\
\mathcal{L}_\ell &= -\sum_{i\in I_\ell}\alpha_i\log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) \ .
\end{aligned}$$

Nous remarquons que $\mathcal{L}_0$ et que le premier terme de $\mathcal{L}_h$ sont constants et indépendants des α_i et β_i. De plus, comme $\alpha_i = 0$ pour $i \in I_0$, le terme quadratique peut être réduit aux indices de l'ensemble actif $I_{\bar{0}} = I_h \cup I_\ell$. En enlevant les termes constants qui n'interviennent pas dans la minimisation, le lagrangien $\mathcal{L}_{\mathrm{primal}}$ se réduit à

$$-\frac{1}{2\lambda}\sum_{(i,j)\in I_{\bar{0}}{}^2}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_j,\boldsymbol{x}_i) - \sum_{i\in I_\ell}\alpha_i\log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) - \sum_{i\in I_h}\alpha_i\mathrm{f}_i \ ,$$

qui ne fait intervenir que $|I_{\bar{0}}| < n$ variables actives.

3.2.4 Optimisation des multiplicateurs de Lagrange

Dans cette section, pour ne pas ajouter inutilement des indices sur les variables de travail, nous noterons $\boldsymbol{\alpha}$ et $\boldsymbol{y}$ les vecteur formés des composantes α_i et y_i respectivement, pour $i \in I_{\bar{0}}$. L'appartenance des exemples à I_0, I_h ou I_ℓ étant supposée connue, le problème d'optimisation peut être maintenant reformulé en

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2\lambda} \sum_{(i,j)\in {I_{\bar{0}}}^2} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) \\ & + \sum_{i\in I_\ell} \alpha_i \log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) + \sum_{i\in I_h} \alpha_i \mathrm{f}_i \ , \\ \text{t. q.} \quad & \sum_{i\in I_{\bar{0}}} \alpha_i y_i = 0 \ . \end{aligned} \tag{11}$$

Le problème (11) étant convexe sous contraintes linéaires, il peut être résolu efficacement par la méthode de Newton [Boyd et Vandenberghe, 2004]. Dans un premier temps, nous écrivons son lagrangien

$$\begin{aligned} \mathcal{L}_{\text{dual}} \quad = \quad & \frac{1}{2\lambda} \sum_{(i,j)\in {I_{\bar{0}}}^2} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) + \sum_{i\in I_\ell} \alpha_i \log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) \\ & + \sum_{i\in I_h} \alpha_i \mathrm{f}_i + \gamma \sum_{i\in I_{\bar{0}}} \alpha_i y_i \ . \end{aligned}$$

L'algorithme se décrivant plus directement sous forme matricielle, nous définissons :

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{hh} & \mathbf{G}_{h\ell} \\ \mathbf{G}_{h\ell}^\intercal & \mathbf{G}_{\ell\ell} \end{bmatrix} \ ,$$

où $\mathbf{G}_{h\ell}$ est la matrice formée des composantes G_{ij} pour $i \in I_h$ et $j \in I_\ell$, et $G_{ij} = \frac{1}{\lambda} y_i y_j K(\boldsymbol{x}_j, \boldsymbol{x}_i)$. De plus, $\mathbf{f}_h$ dénote le vecteur formé des composantes f_i pour $i \in I_h$, $\mathbf{d}$ est un vecteur de taille $|I_\ell|$ tel que $d_i = \log\left(\alpha_i/(1-\alpha_i)\right)$ pour $i \in I_\ell$, ce qui permet de définir

$$\mathbf{e} = [\mathbf{f}_h^\intercal \ \mathbf{d}^\intercal]^\intercal \ .$$

Les conditions d'optimalité du premier ordre s'écrivent alors

$$\nabla \mathcal{L}_{\text{dual}}(\boldsymbol{\alpha}) = \mathbf{G}\boldsymbol{\alpha} + \mathbf{e} + \gamma \boldsymbol{y} = 0 \ , \tag{12}$$

et la condition de faisabilité est

$$\boldsymbol{y}^\intercal \boldsymbol{\alpha} = 0 \ . \tag{13}$$

Des équations (12) et (13), on déduit la valeur du paramètre de Lagrange γ :

$$\gamma = -\frac{\boldsymbol{y}^\intercal \mathbf{G}^{-1} \mathbf{e}}{\boldsymbol{y}^\intercal \mathbf{G}^{-1} \boldsymbol{y}} \ ,$$

ce qui permet ensuite de minimiser le lagrangien en $\boldsymbol{\alpha}$ par une méthode de Newton.

La hessienne H de $\mathcal{L}_{\text{dual}}$ s'écrit

$$H(\boldsymbol{\alpha}) = \begin{bmatrix} \mathbf{G}_{hh} & \mathbf{G}_{\ell h} \\ \mathbf{G}_{\ell h}^\intercal & \mathbf{G}_{\ell\ell} + \mathbf{D}_{\ell\ell}(\boldsymbol{\alpha}) \end{bmatrix} \ , \tag{14}$$

où $\mathbf{D}_{\ell\ell}(\boldsymbol{\alpha})$ est une matrice diagonale $|I_\ell| \times |I_\ell|$ telle que $D_{ii} = 1/(\alpha_i(1-\alpha_i))$. Comme $0 < \alpha_i < 1$ pour $i \in I_\ell$, on

constate sans surprise que, pour peu que le noyau K soit défini positif, la hessienne est définie positive.

L'étape s de Newton consiste à résoudre

$$H(\boldsymbol{\alpha}^{(s)})\,(\boldsymbol{\alpha}^{(s+1)} - \boldsymbol{\alpha}^{(s)}) = -\nabla\mathcal{L}_{\mathrm{dual}}(\boldsymbol{\alpha}^{(s)}) \ , \tag{15}$$

c'est-à-dire,

$$H(\boldsymbol{\alpha}^{(s)})\,\boldsymbol{\alpha}^{(s+1)} = \boldsymbol{\delta}^{(s)} \ , \tag{16}$$

avec $\boldsymbol{\delta}^{(s)\mathsf{T}} = \left[\boldsymbol{\delta}_h^{(s)\mathsf{T}}\ \boldsymbol{\delta}_\ell^{(s)\mathsf{T}}\right]$, où $\boldsymbol{\delta}_h^{(s)}$ est un vecteur de taille $|I_h|$ tel que $\delta_i^{(s)} = -\mathrm{f}_i - \gamma y_i$, et $\boldsymbol{\delta}_\ell^{(s)}$ est un vecteur de taille $|I_\ell|$ tel que $\delta_i^{(s)} = -\log(\alpha_i^{(s)}/(1-\alpha_i^{(s)})) - \gamma y_i + 1/(1-\alpha_i^{(s)})$. Les étapes de Newton sont répétées jusqu'à convergence ou jusqu'à ce que la partition $\{I_0, I_h, I_\ell\}$ doive être modifiée.

3.2.5 Mise à jour de la partition

Étant donnée une partition, chaque étape de la méthode de Newton retourne une solution améliorant la fonction objectif du problème (11). Cette solution doit obéir aux conditions de la Table 1 pour être consistante avec la conjecture initiale. Les exemples appartenant à I_h et I_ℓ doivent donc vérifier :

$$\forall i \in I_h \quad 0 \quad \leq \alpha_i \leq \frac{1}{1+e^{-\mathrm{f}_i}} \tag{17a}$$

$$\forall i \in I_\ell \quad \alpha_i \quad \geq \frac{1}{1+e^{-\mathrm{f}_i}} \qquad . \tag{17b}$$

Si ce n'est pas le cas, la partition courante doit être revue. Dans un premier temps, la mise à jour est corrigée en ramenant tous les α_i dans un domaine admissible avec la conjecture de départ ; Pour ce faire, on calcule le plus grand pas ρ tel que

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(s)} + \rho(\boldsymbol{\alpha}^{(s+1)} - \boldsymbol{\alpha}^{(s)}) \tag{18}$$

satisfasse les contraintes (17). Ensuite, notant i une composante atteignant la frontière du domaine, la partition est modifiée comme suit :

$$\begin{cases} \text{si } i \in I_h \text{ et } \alpha_i = 0, & i \text{ est déplacé en } I_0 \ ; \\ \text{si } i \in I_h \text{ et } \alpha_i = \frac{1}{1+e^{-\mathrm{f}_i}}, & i \text{ est déplacé en } I_\ell \ ; \\ \text{si } i \in I_\ell \text{ et } \alpha_i = \frac{1}{1+e^{-\mathrm{f}_i}}, & i \text{ est déplacé en } I_h. \end{cases}$$

Enfin, partant de cette nouvelle conjecture et du $\boldsymbol{\alpha}$ courant, les étape de Newton sont reprises, et la procédure est itérée jusqu'à satisfaction de toutes les contraintes.

Lorsqu'un point fixe de (15) est atteint sans qu'aucune contrainte ne soit violée sur l'ensemble actif, il reste à vérifier si de nouveaux exemples sont candidats pour entrer dans l'ensemble actif. Tout exemple $i \in I_0$ tel que $-y_i(f(\boldsymbol{x}_i) + b) > \mathrm{f}_i$ est candidat. Nous choisissons arbitrairement celui qui maximise $-y_i(f(\boldsymbol{x}_i) + b) - \mathrm{f}_i$. Limiter les modifications de l'ensemble actif à un seul exemple à la fois assure la stabilité de l'algorithme. Cette stratégie permet également une mise à jour efficace de la décomposition de Cholesky de la matrice hessienne (14). Lorsqu'il n'y a plus de candidat dans I_0,

l'algorithme a atteint la solution optimale.

La connaissance de b est requise pour tester l'appartenance des nouveaux exemples aux différents ensembles ; il peut être calculé par les exemples de l'ensemble I_h, pour lesquels $-y_i(f(\boldsymbol{x}_i) + b) = \mathrm{f}_i$. D'autre part, comme

$$f(\boldsymbol{x}_i) = \sum_{j \in I_{\bar{0}}} \alpha_j G_{ij} \ , \tag{19}$$

par identification avec l'équation (12), nous voyons que $b = \gamma$.

4 Expériences

4.1 Données artificielles

Nous générons ici un jeu de données dans le plan, afin de comparer les comportements des classifieurs standards et parcimonieux pour l'estimation de probabilité. La distribution étant connue, nous pouvons évaluer la précision de l'estimation des probabilités conditionnelles.

La distribution des données suit une loi mélange formée de deux composantes gaussiennes de même matrice de covariance, égale à la matrice identité. Les composantes correspondant à la classe positive et à la classe négative sont respectivement centrées en $(1\ 1)^\intercal$ et en $(-1\ -1)^\intercal$. Nous créons un léger déséquilibre des classes en fixant la probabilité *a priori* de la classe positive à 0.3. Les ensembles d'apprentissage et de validation sont formés de 200 exemples (30% de positifs et 70% de négatifs). Nous utilisons la version à noyaux des régressions logistiques, pour les modèles standard et parcimonieux, avec $[p_{\min}, p_{\max}] = [0.2, 0.4]$, puis $[p_{\min}, p_{\max}] = [0.4, 0.6]$ de manière à illustrer comment ces réglages modifient la réponse de l'estimateur. Pour ces trois variantes, nous utilisons le même noyau gaussien, dont la largeur de bande ($10^{-0.3}$) a été sélectionnée par minimisation de la divergence de Kullback-Leibler entre la distribution des caractéristiques $\boldsymbol{x}$ et l'estimateur de Parzen construit à partir de l'ensemble d'apprentissage. Enfin, pour chacun des estimateurs, le paramètre de régularisation est sélectionné de manière à minimiser le critère d'ajustement sur un ensemble de validation (10^0 pour la régression logistique standard, $10^{0.4}$ pour la régression parcimonieuse avec $[p_{\min}, p_{\max}] = [0.2, 0.4]$ et $10^{1.2}$ pour la régression parcimonieuse avec $[p_{\min}, p_{\max}] = [0.4, 0.6]$).

Les Figures 3, 4 et 5 résument les valeurs des probabilités estimées, sur 20 000 exemples indépendants, en fonction de la valeur réelle de la probabilité conditionnelle, pour les différentes versions testées. Plus la distribution des valeurs estimées est proche de la droite $\hat{p} = p$ représentée en pointillés, meilleure est l'estimation. La régression logistique vise à se rapprocher de cette diagonale sur l'intervalle $[0, 1]$, alors que les versions parcimonieuses ne sont précises que sur $[p_{\min}, p_{\max}]$, et tendent à saturer vers $p_{\min}$ ou $p_{\max}$ en dehors de cet intervalle. On voit sur ces figures que l'objectif est bien atteint, avec des estimations mieux calibrées et moins variable dans l'intervalle $[p_{\min}, p_{\max}]$.

La Table 2 donne une évaluation quantitative en termes de coûts de mauvais classement pour des seuils de décision centrés sur les deux intervalles $[p_{\min}, p_{\max}]$ considérés. On y observe que, quand le seuil de décision est dans l'intervalle $[p_{\min}, p_{\max}]$, la régression logistique parcimonieuse est un peu plus précise que la version standard, et qu'elle n'est pas compétitive hors de ce domaine. Le nombre de vecteurs support, également rapporté pour les deux réglages de $[p_{\min}, p_{\max}]$ montre que le mécanisme de parcimonie est efficace, même pour ce problème où les classes ne sont pas très

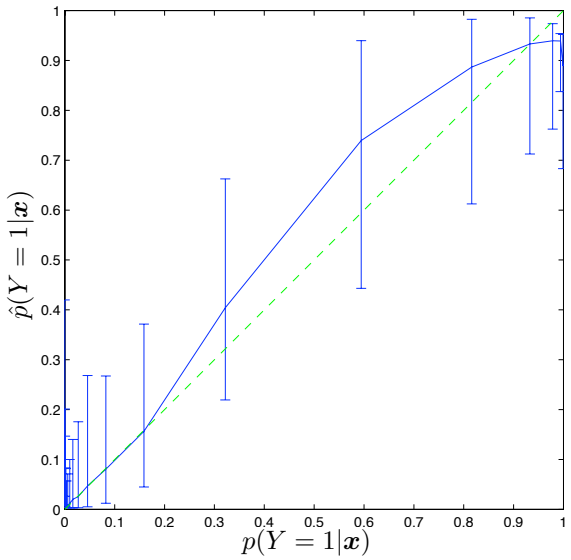

FIG. 3: Médiane et intervalle à 80% des probabilités conditionnelles estimées en fonction de la probabilité réelle pour la régression logistique standard

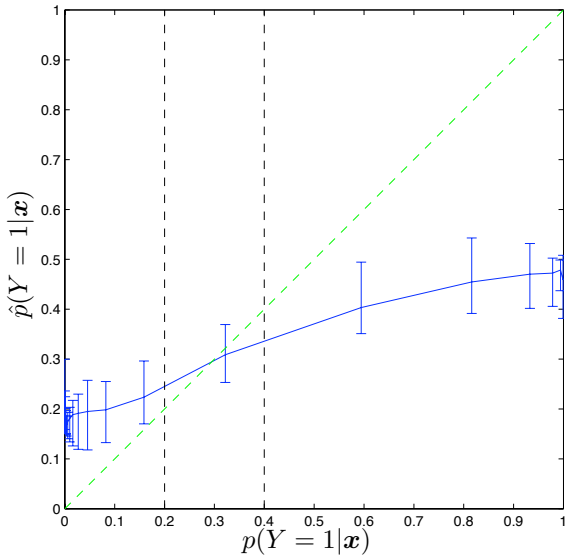

FIG. 4: Médiane et intervalle à 80% des probabilités conditionnelles estimées en fonction de la probabilité réelle pour la régression logistique parcimonieuse, $p_{\min} = 0.2$ et $p_{\max} = 0.4$

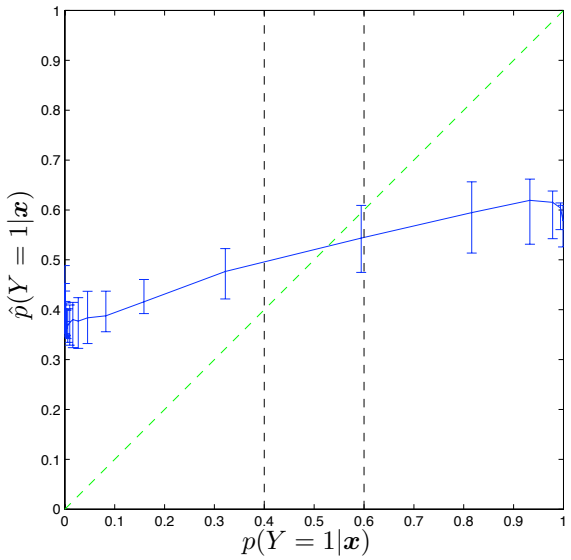

FIG. 5: Médiane et intervalle à 80% des probabilités conditionnelles estimées en fonction de la probabilité réelle pour la régression logistique parcimonieuse, $p_{\min} = 0.4$ et $p_{\max} = 0.6$

TAB. 2: Coût espéré (en %) pour $c_- = \pi^+$ et $c_+ = 1 - \pi^+$ et nombre de vecteurs supports, en fonction du seuil de décision π^+ pour les régressions logistiques caractérisées par l'intervalle $[p_{\min}, p_{\max}]$.

	$[p_{\min}, p_{\max}]$		
	[0, 1]	[0.2, 0.4]	[0.4, 0.6]
$\pi^+ = 0.3$	4.43	3.92	21.02
$\pi^+ = 0.5$	4.28	12.68	4.13
#SV	200	43	64

bien séparées. Sous les hypothèses de [Bartlett et Tewari, 2007, Théorème 3] sur le noyau et le terme de pénalisation, quand la taille de l'échantillon tend vers l'infini, les fractions de vecteurs support devraient être ici de l'ordre de 10% pour les deux versions de la régression logistique parcimonieuse.

4.2 Données réelles

Pour étudier le problème de deux classes déséquilibrées, nous avons choisi la base de donnée *Forest Covertype*, qui est la plus grande base de données de l'UCI. [2] Les exemples sont décrits par 54 caractéristiques ; 10 sont quantitatives et 44 sont binaires. Originellement, il y a 7 classes , mais nous considérons la discrimination de la classe positive *Krummholz* (20 510 exemples) de la classe négative classe *Épicéa/Sapin* (211 840 exemples). La proportion de la classe positive est de 8.8%, et les classes sont relativement bien séparées. Comme il n'y a pas de matrice de coût liée à ces données, nous avons arbitrairement choisi les coûts pour les faux positifs et les faux négatifs, c_- et c_+, de façon à encourager un taux d'erreur équivalent pour les deux catégories, c'est-à-dire

$$\frac{c_-}{c_+ + c_-} = \pi^+ \ , \tag{20}$$

où $\pi^+ = 8.8\%$ est la proportion d'exemples positifs.

[2] Disponible sur `http ://kdd.ics.uci.edu/databases/covertype`.

Les coûts sont alors définies à un facteur près, et nous choisissons

$$\begin{cases} c_- & = \pi^+ \ , \text{ et} \\ c_+ & = 1 - \pi^+ \ . \end{cases}$$

4.2.1 Cadre d'expérimentation

Pour assurer la représentativité des résultats, les données sont réparties en 10 sous-ensembles. Chaque sous-ensemble est itérativement utilisé an tant qu'ensemble d'apprentissage alors que les sous-ensembles restants sont utilisés comme ensembles de test. Ainsi, les ensembles d'apprentissage comprennent 23 235 exemples. La proportion d'exemples positifs (minoritaires) est identique pour tous les sous-ensembles. Les caractéristiques sont normalisées (centrées et réduites) avant chaque session d'apprentissage.

Les expériences rapportées ici ont été effectuées par des classifieurs linéaires. Nous avons optimisé le paramètre de pénalisation λ pour la régression logistique (5) et la régression logistique parcimonieuse (6) par une validation croisée sur 5 blocs. Nous avons optimisé conjointement le seuil de décision, cette procédure est souvent appliquée aux classifieurs dans le but de corriger le biais des probabilités estimées. La correction du biais doit favoriser la régression logistique.

L'intervalle $[p_{\min}, p_{\max}]$ des probabilités conditionnelles, qui est supposé être défini par l'utilisateur, n'est pas optimisé. De meilleurs résultats d'optimisation sont attendus pour de petits intervalles, mais l'intervalle des probabilités conditionnelles fiables se réduit alors. Nous rapportons les résultats pour diverses longueurs d'intervalles centrés sur π^+ sur l'échelle logarithmique, c'est-à-dire,

$$\sqrt{p_{\min} p_{\max}} = \pi^+ \ .$$

4.2.2 Résultats

Nous rapportons les performances moyennes de la régression logistique parcimonieuse, ainsi que leur écart-type dans la Table 3. Comme attendu, la moyenne du coût de test décroît lorsque l'intervalle $[p_{\min}, p_{\max}]$ décroît, $p_{\max} - p_{\min} = 1$ représente la régression logistique standard. Nous montrons aussi le seuil de décision moyen sur les probabilités conditionnelles estimées, seuil estimé par validation croisée. Ce dernier est juste au dessus de $\pi^+ = 8.8\%$ pour la régression logistique standard, mais la différence n'est peut-être pas significative (les tests d'hypothèses usuels ne peuvent être appliqués car les expériences ne sont pas indépendantes). Le seuil de décision correct est toujours choisi pour la régression logistique parcimonieuse sur de petits intervalles $[p_{\min}, p_{\max}]$. Les classifieurs sont donc bien calibrés en décision. La proportion d'exemples de l'ensemble actif (noté SV par identification aux vecteurs supports) est rapportée sur la dernière ligne. Cette proportion diminue aussi lorsque l'intervalle $[p_{\min}, p_{\max}]$ décroît.

La Figure 6 compare, pour un essai, la sensibilité au seuil de détection du coût de test moyen des régressions logistiques standard et parcimonieuse (avec $p_{\max} - p_{\min} = 2.2\%$). Les figures obtenues pour les autres essais sont similaires. A savoir, la régression logistique possède un minimum plat et large, reflétant le fait que la proportion de décisions correctes ne change pas beaucoup autour du seuil de décision. Cela veut dire que les vraies probabilités conditionnelles fluctuent de façon non-monotone dans cette région. La régression logistique parcimonieuse se comporte beaucoup mieux avec un minimum plus étroit et plus bas centré sur π^+, reflétant des probabilités conditionnelles bien calibrées dans la région

ciblée.

La Table 4 résume les résultats obtenus avec les séparateurs à vaste marge. Les résultats des SVM standards sont mauvais parce que le coût optimisé, avec $c_+ = c_-$, n'est pas le bon. Ceci peut être compensé en déplaçant le seuil de décision. Les performances correspondantes sont montrées dans la colonne « Avec correction du biais ». Cependant, un meilleur choix consiste à modifier la perte *hinge* pour faire en sorte que $c_+ \neq c_-$ [Osuna et al., 1997]. Le résultat correspondant, donné dans la colonne c_+/c_-, atteint les performances de la régression logistique parcimonieuse sur des petits intervalles d'intérêt. Le nombre de vecteurs supports (notés SV) pour le cas c_+/c_- et le nombre d'exemples dans l'ensemble actifs pour les petits intervalles $[p_{\min}, p_{\max}]$ de la régression logistique parcimonieuse sont du même ordre de grandeur.

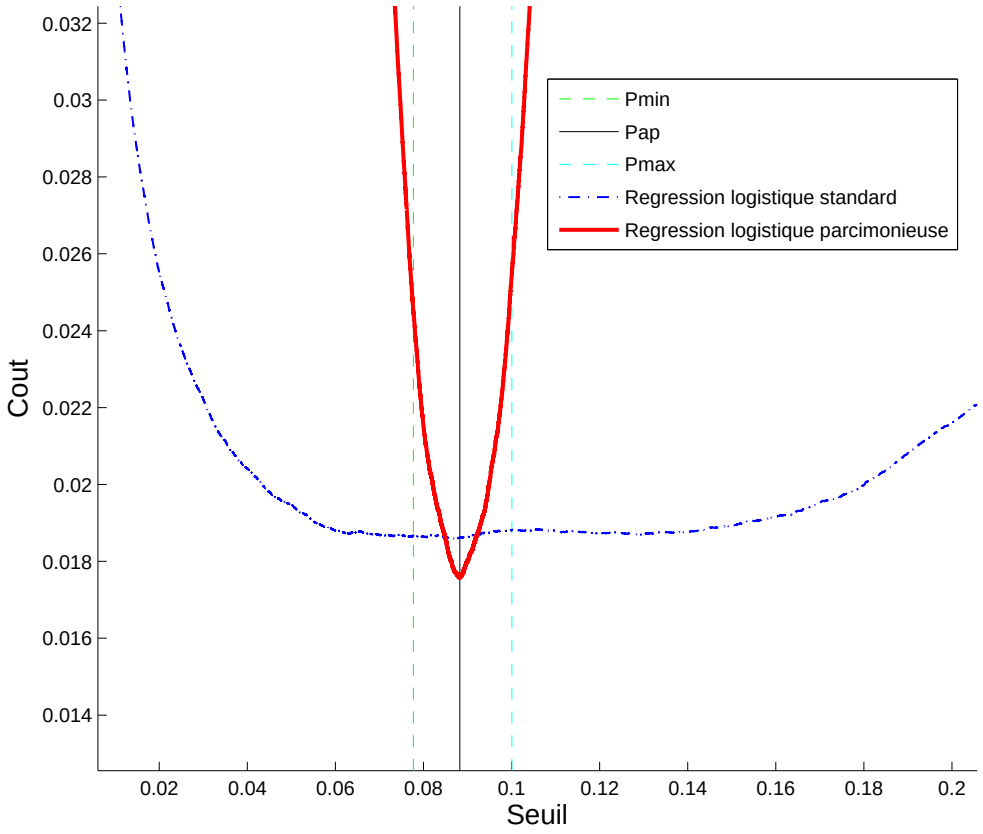

FIG. 6: Coût de test en fonction du seuil de décision.

TAB. 3: Coûts moyens de test pour les régression logistiques parcimonieuse et standard ($p_{\max} - p_{\min} = 100\%$).

$p_{\min}$ (%)	0	0.4	1.0	2.9	4.8	7.8
$p_{\max}$ (%)	100	72.0	47.5	24.1	15.8	10.0
$p_{\max} - p_{\min}$ (%)	100	71.6	46.4	21.2	11.0	2.2
Coût moyen de test ($\times 10^{-2}$)	1.86	1.86	1.85	1.85	1.83	1.78
Écart-type	± 0.01	± 0.01	± 0.01	± 0.01	± 0.02	± 0.02
Seuil moyen de décision (%)	9.5	9.0	9.0	9.0	8.8	8.8
Écart-type	± 1.3	± 1.1	± 0.9	± 0.6	± 0.2	± 0.0
Prop. moyenne de SV (%)	100	65.5	53.5	40.5	34.0	27.9

TAB. 4: Coûts moyens de test obtenus pour les SVMs.

SVM	Standard	avec correction du biais	c_+/c_-
Coût moyen de test ($\times 10^{-2}$)	3.75 ± 0.23	2.31 ± 0.12	1.79 ± 0.02
Prop. moyenne de SV (%)	12.84 ± 0.79	13.16 ± 1.13	26.19 ± 0.60

5 Discussion

Nous avons proposé un nouveau critère d'apprentissage visant à apprendre des probabilités conditionnelles fiables au voisinage de la frontière de décision. Ce critère, qui consiste à tronquer la log-vraisemblance binomiale, produit un classifieur probabiliste parcimonieux. Nous avons examiné en détail comment la régression logistique est modifiée par la maximisation de cette « vraisemblance locale », mais ce principe peut être appliqué sur d'autres modèles de probabilités conditionnelles comme les réseaux de neurones. Bien que nous ayons uniquement discuté du problème de classification binaire, le principe est par essence multi-classe et peut être appliqué à une log-vraisemblance multinomiale. Le problème d'optimisation résultant reste convexe pourvu que l'ensemble des probabilités conditionnelles soit convexe.

Des expériences sont en cours pour confirmer l'intérêt pratique des classifieurs probabilistes parcimonieux sur des problèmes de détection sur des séquences vidéo, mais ils offrent déjà des résultats prometteurs pour le problème des classes déséquilibrées. Le critère d'apprentissage ignore les exemples bien classés hors de la « zone grise », définie par un intervalle sur les probabilités conditionnelles. Les exemples actifs sont des exemples ambigus et mal classés, ce qui permet d'ignorer bon nombre d'exemples de la classe majoritaire. Il y a donc un sous-échantillonnage virtuel et ciblé de la classe majoritaire.

Nos premières expériences sur des classifieurs linéaires montrent que les classifieurs probabilistes tirent profit de la concentration du critère sur la « zone grise » près de la frontière de décision. La régression logistique parcimonieuse fournit de meilleures règles de décision que la régression logistique standard. Non seulement nous gagnons en erreur de test mais également en temps de calcul, grâce à la mise à l'écart des données non pertinentes. Les performances et les temps d'apprentissage sont comparables aux SVM entraînés avec des coûts asymétriques c_+/c_-, et nous pouvons profiter en plus de probabilités bien calibrées dans le voisinage de la frontière de décision.

A Limitation du pas de descente

Pour éviter les effets de bord produit par la non-dérivabilité du critère, nous contrôlons le pas ρ de telle manière qu'un exemple inactif positif rentrant dans l'ensemble actif ne puisse pas produire une réponse inférieure à $\widehat{p}_t$ et qu'un exemple inactif négatif rentrant dans l'ensemble actif ne puisse pas produire une réponse supérieure à $\widehat{p}_t$.

Nous avons,

$$\boldsymbol{w}^{(s+1)\intercal}\boldsymbol{x}_i = (\boldsymbol{w}^{(s)} - \rho\Delta\boldsymbol{w})^\intercal\boldsymbol{x}_i \ ,$$

Si $\Delta\boldsymbol{w}^\intercal\boldsymbol{x}_i < 0$,

$$\boldsymbol{w}^{(s+1)\intercal}\boldsymbol{x}_i > \boldsymbol{w}^{(s)\intercal}\boldsymbol{x}_i \ ,$$

alors

$$\widehat{p}(Y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}^{(s+1)}) > \widehat{p}(Y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}^{(s)}) \ .$$

Respectivement, si $\Delta\boldsymbol{w}^\intercal\boldsymbol{x}_i > 0$,

$$\boldsymbol{w}^{(s+1)\intercal}\boldsymbol{x}_i < \boldsymbol{w}^{(s)\intercal}\boldsymbol{x}_i \ ,$$

alors

$$\widehat{p}(Y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}^{(s+1)}) < \widehat{p}(Y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}^{(s)}) \ .$$

Nous pouvons en déduire que les exemples problématiques vont être lers exemples actifs pour lesquels,

$$y_i\Delta\boldsymbol{w}^\intercal\boldsymbol{x}_i > 0$$

Dans ce cas, nous voulons,

$$\begin{aligned} y_i\widehat{p}_t &< y_i\widehat{p}(Y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}^{(s+1)}) \ , \\ \text{soit } \ y_i \log\left(\frac{\widehat{p}_t}{1-\widehat{p}_t}\right) &< y_i(\boldsymbol{w}^{(s)} - \rho\Delta\boldsymbol{w})^\intercal\boldsymbol{x}_i \ , \\ \text{ou encore } \ y_i\left(\rho\Delta\boldsymbol{w}^\intercal\boldsymbol{x}_i\right) &< y_i\left(\boldsymbol{w}^{(s)\intercal}\right)\boldsymbol{x}_i - \log\left(\frac{\widehat{p}_t}{1-\widehat{p}_t}\right) \ . \end{aligned}$$

Comme nous avons $y_i\Delta\boldsymbol{w}^\intercal\boldsymbol{x}_i > 0$, nous obtenons une borne supérieure pour ρ,

$$\rho < \frac{\boldsymbol{w}^{(s)\intercal}\boldsymbol{x}_i - \log\left(\frac{\widehat{p}_t}{1-\widehat{p}_t}\right)}{\Delta\boldsymbol{w}^\intercal\boldsymbol{x}_i} \ .$$

Nous pouvons alors connaître la limite supérieure ρ_o de ρ pour qu'aucun exemple entrant dans l'ensemble actif ne franchisse la barrière $\widehat{p}_t$:

$$\rho_o = \underset{i\in Ip}{min}\frac{\boldsymbol{w}^{(s)\intercal}\boldsymbol{x}_i - \log\left(\frac{\widehat{p}_t}{1-\widehat{p}_t}\right)}{\Delta\boldsymbol{w}^\intercal\boldsymbol{x}_i} \ .$$

Références

Peter L. Bartlett, Michael I. Jordan, et Jon D. McAuliffe. Large margin classifiers : convex loss, low noise, and convergence rates. Dans *Advances in Neural Information Processing Systems, 16*, 2004.

Peter L. Bartlett et Ambuj Tewari. Sparseness vs estimating conditional probabilities : Some asymptotic results. *Journal of Machine Learning Research*, 8 :775–790, 2007. ISSN 1533-7928.

S. Boyd et L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 0521833787.

Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5) :1155–1178, 2007. ISSN 0899-7667.

C. Elkan. The foundations of cost-sensitive learning. Dans *IJCAI*, pages 973–978, 2001.

Y. Grandvalet, J. Mariéthoz, et S. Bengio. A probabilistic interpretation of SVMs with an application to unbalanced classification. Dans Y. Weiss, B. Schölkopf, et J. C. Platt, éditeurs, *Advances in Neural Information Processing Systems 18*, pages 467–474. MIT Press, 2006.

S. S. Keerthi, K. B. Duan, S. K. Shevade, et A. N. Poo. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61(1-3) :151–165, 2005. ISSN 0885-6125.

G. Loosli, S. Canu, S. Vishwanathan, et M. Chattopadhay. Boîte à outils SVM simple et rapide. *RIA – Revue d'Intelligence Artificielle*, 19(4/5) :741–767, 2005.

E. Osuna, R. Freund, et F. Girosi. Support vector machines : Training and applications. Rapport Technique A.I. Memo No. 1602, M.I.T. AI Laboratory, 1997.

J. C. Platt. Probabilities for SV machines. Dans A. J. Smola, P. L. Bartlett, B. Schölkopf, et D. Schuurmans, éditeurs, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

V. Roth. Probabilistic discriminative kernel classifiers for multi-class problems. Dans *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 246–253, London, UK, 2001. Springer-Verlag. ISBN 3-540-42596-9.

K. M. Ting. A comparative study of cost-sensitive boosting algorithms. Dans *Proceedings of the 17th International Conference on Machine Learning*, pages 983–990. Morgan Kaufmann, 2000.

S. V. N. Vishwanathan, Alex J. Smola, et M. Narasimha Murty. SimpleSVM. Dans *Proceedings of the 20th International Conference on Machine Learning*, pages 760–767, 2003.

J. Zhu et T. Hastie. Kernel logistic regression and the import vector machine. Dans *Advances in Neural Information Processing Systems 13*, 2001.

Auteurs

Romain HÉRAULT

Romain HÉRAULT a obtenu le doctorat en Technologie de l'Information et des Systèmes de l'Université de Technologie de Compiègne pour ses travaux sur le suivi du visage et des gestes faciaux et sur l'apprentissage statistique appliqué à la problématique des classes déséquilibrées. Cette dernière problématique l'a poussé à étudier les classifieurs parcimonieux.

Il est, depuis septembre 2008, Maître de Conférences à l'Institut National des Sciences Appliquées de Rouen.

Yves GRANDVALET

Yves Grandvalet est chargé de recherches au CNRS et membre du laboratoire Heudiasyc, UMR 6599 de l'Université de Technologie de Compiègne depuis 1996. En tant que chercheur visiteur, il a fait partie du corps scientifique de l'Université de Montréal en en 2002-2003, puis de l'institut de recherche Idiap en 2006-2008. Son domaine de recherche est l'apprentissage statistique. Outre le cipaille et la brisolée, ses intérêts vont de la définition de critères d'ajustement à celle de la définition de familles de modèles, en passant par les algorithmes de sélection de modèle.

Sparse probabilistic classifier

Abstract

The scores returned by support vector machines are often used as a confidence measures in the classification of new examples. However, there is no theoretical grounds sustaining this practice. Thus, when classification uncertainty has to be assessed, it is safer to resort to classifiers estimating conditional probabilities of class labels. Here, we focus on the ambiguity in the vicinity of the boundary decision. We propose an adaptation of maximum likelihood estimation, instantiated on logistic regression. The model outputs proper conditional probabilities into a user-defined interval and is less precise elsewhere. The model is sparse, in the sense that few examples contribute to the solution. The computational efficiency is thus improved compared to logistic regression. Furthermore, preliminary experiments show improvements over standard logistic regression with performances similar to support vector machines.

Key words

Statistical learning, Sparse classifier, Imbalanced Classes.

Introduction

When a vast majority of examples belong to the negative "uninteresting" class, and only a few interesting examples are available, learning tends to be biased towards the recognition of the majority class. This problem can be addressed by rebalancing the training distribution Ting [2000], Elkan [2001], either by over-sampling the minority class, or generating artificial examples of the minority class, or by down-sampling the majority class. However, undersampling may discard relevant pieces of information and oversampling is not computationally efficient. The class imbalance problem, which is our original motivation for this work, should benefit of a sparse classifier which remain accurate in a focused range of conditional probabilities.

Support Vector Machine (SVMs) are the most spread sparse models. There have been several attempts to turn their returned scores into probabilistic assignments Platt [2000], Grandvalet et al. [2006]. However, there is no guaranty that these scores reflect a classification confidence ; we even know that the conditional probabilities of class labels cannot be recovered unambiguously except at the decision boundary Bartlett et Tewari [2007]. Thus, when the classification uncertainty has to be assessed, estimating conditional probabilities is better motivated.

We propose to build probabilistic classifiers that are accurate on the "gray zone", where class labels switch. Well-calibrated probabilities in this area allow to assess classification uncertainty. The classifier also provides relevant decision rules for the set of corresponding asymmetric misclassification losses. Focusing on a small range of conditional probabilities instead of estimating them on their full span has two advantages. First, the training objective is closer to the ultimate goal of minimizing the misclassification risk, and second, inaccuracy outside of the focus range is a key element for kernelized models, since Bartlett et Tewari [2007] proved that sparsity does not occur when the conditional probabilities can be unambiguously estimated everywhere. Thus, the inaccuracy of the conditional probabilities outside of the focused interval is a key to kernel methods efficiency.

Sparsity refers here to the limited number of non-zero elements in a kernel expansion. It implies that many training examples have no influence in the training process, thus improving its computational efficiency.

Our approach is closely related to the methods adjusting the training objective by using different losses for positive and negative examples Osuna et al. [1997], Ting [2000]. It however differs from the latter which essentially consist in applying different weights to the different categories.

Learning criterion

We consider the following learning set $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where each example is described by features $\mathbf{x}_i$ and the associated class label $y_i \in \{-1, 1\}$. Assuming independent examples, estimating $p(y|\boldsymbol{x})$ can be performed by maximizing the conditional log-likelihood, equation (1), where $\widehat{p}(y|\boldsymbol{x})$ denotes the estimate of $p(y|\boldsymbol{x})$.

The Bayes decision rule is defined by the true conditional probabilities $p(y|\boldsymbol{x})$ and by misclassification losses. In binary problems, where the class is tagged $+1$ or -1, the two types of errors are : false positive, incurring a loss c_- ; false negative, incurring a loss c_+.

Although Bayes' decision rule is defined in terms of $p(y|\boldsymbol{x})$, it does not require a precise estimate everywhere. It is sufficient to estimate precisely the conditional probabilities at $\frac{c_-}{c_+ + c_-}$, which defines the decision boundary. This is precisely what SVMs achieve asymptotically [Bartlett et Tewari, 2007] for $p(y|\boldsymbol{x}) = 0.5$, or for other probabilities when the criterion is asymetric [Osuna et al., 1997].

We focus on a small range $[p_{\min}, p_{\max}]$. Outside of this range, we only want to know whether $p(y|\boldsymbol{x})$ is smaller than $p_{\min}$ or greater than $p_{\max}$. This optimization problem can be formalized as maximizing equation (3) which is a concave criterion in $\widehat{p}(y = 1|\boldsymbol{x}_i)$. This criterion is classification-calibrated provided $\frac{c_-}{c_+ + c_-} \in [p_{\min}, p_{\max}]$.

Application to logistic regression

Logistic regression is a standard probabilistic model which considers that the log-ratio of conditional probabilities is linear, equation (4) and where the coefficients $(\mathbf{w}, b)$ are estimated by maximizing the likelihood (1) or the penalized likelihood. Logistic regression can be kernelized by letting the log-ratio of conditional probabilities to be non-linear, $f(\boldsymbol{x}_i) + b$, where f belongs to a given reproducing kernel Hilbert space $\mathcal{H}$. The training criterion should incorporate a regularization term to prevent overfitting [Roth, 2001, Zhu et Hastie, 2001].

Maximizing the likelihood (eq. 1) penalized by the norm of f results in minimizing equation (5) Unlike SVMs,

logistic regression does not yield sparse solutions, in the sense that all examples influence the solution.

Our approach consists in replacing the log-likelihood term by criterion (eq. 3). Sparse kernelized logistic regression minimizes equation (eq. 6) where $\mathrm{f}_i = -\log \frac{p_{\max}}{1-p_{\max}}$ if $y_i = 1$ and $\mathrm{f}_i = \log \frac{p_{\min}}{1-p_{\min}}$ if $y_i = -1$. In this training criterion, it is the fitting term, instead of the penalization term, which causes sparsity. Training examples with large values of $y_i f(\boldsymbol{x}_i)$ will not contribute to the final classifier.

Kernel logistic regression can be learned in the primal using Newton's method [Roth, 2001], or in the dual [Keerthi et al., 2005]. We propose an active set algorithm, following a strategy that proved to be efficient for SVMs. The SimpleSVM algorithm [Vishwanathan et al., 2003, Loosli et al., 2005] solves the SVM training problem by a greedy approach, in which one solves a series of small problems. First, the training examples are assumed to be either support vectors or not, and the training criterion is optimized considering that the partition of examples is fixed. This optimization results in a new partition of examples in support and non-support vectors. These two steps are iterated until some level of accuracy is reached [Loosli et al., 2005].

Experimentations

For experimenting with unbalanced two-class problems, we used the Forest database, the largest available UCI dataset. [3]

We report the mean results (with standard deviations) of sparse logistic regression in Table 3. As expected, the mean test loss (that is, the average test errors weighted by c_+ and c_-), and the number of examples in the working set (denoted SVs for support vectors), decrease smoothly as the $[p_{\min}, p_{\max}]$ interval decreases ($p_{\max} - p_{\min} = 1$ is the standard logistic regression).

Figure 6 compares, for one trial, the sensitivity of the mean test loss of logistic regression and sparse logistic regression (with $p_{\max} - p_{\min} = 2.2\%$) according to the decision threshold. Sparse logistic regression behaves much better, with a lower, narrower minimum centered at π^+, reflecting well-calibrated conditional probabilities in the targeted region.

Table 4 summarizes the results obtained with SVMs. Standard SVMs perform very badly because they are optimized with equal costs c_- and c_+ for false positive and false negative. The corresponding result, displayed under the column c_+/c_-, reaches performance and percentage of SVs similar to the one of sparse logistic regression with small $[p_{\min}, p_{\max}]$ intervals.

Discussion

We proposed a new fitting criterion, which consists in truncating the binomial log-likelihood. This criterion produces sparse probabilistic classifiers, which output faithful conditional probabilities in the vicinity of the decision boundary. We detailed how logistic regression is modified by this « lazy likelihood estimation », but the principle can be applied to any model of conditional probabilities, such as feedforward neural networks. Also, though we only discussed binary classification problems, the principle is in essence multi-class and can be applied to the multinomial log-likelihood. The resulting optimization problem remains convex provided that the specified range of « interesting » conditional probabilities defines a convex set.

Our experiments with linear classifiers show that probabilistic classifiers benefit from the focus on the « gray zone » close to the boundary decision. Sparse logistic regression provided better decision rules than logistic regression. Not only it gains in test error rates but it is also much faster to train, thanks to its ability to ignore uninformative data. The performance and training time match SVMs trained with asymmetrical costs c_+/c_-, and we furthermore enjoy well-calibrated probabilities in the vicinity of the boundary decision.

[3] Available at `kdd.ics.uci.edu/databases/covertype`.