# Régression Logistique Parcimonieuse

Romain Hérault, Yves Grandvalet

# Régression Logistique Parcimonieuse

Romain Hérault (1), Yves Grandvale (2)

(1) HEUDIASYC, UMR CNRS 6599,
Université de Technologie de Compiègne
BP 20529, 60205 Compiègne cedex, France
Fax : +33 (0)3 44 23 44 77, Tél : +33 (0)3 44 23 44 23 poste 51 84
romain.herault@hds.utc.fr

(2) IDIAP,
Rue du Simplon 4, Case Postale 592
CH-1920 Martigny, Switzerland
Tél : +41 277 217 753
yves.grandvalet@idiap.ch

**Résumé**

Les résultats 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 d'avoir recours 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, appliquée à la régression logistique. 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. L'efficacité du calcul est ainsi améliorée par rapport à la régression logistique. De plus, nos premières expériences montrent une amélioration des performances par rapport à la régression logistique standard, avec des performances similaires à celles des séparateurs à vaste marge.

## Mots clés

Classifieur probabiliste, Classes déséquilibrées, Parcimonie, Critère local

## 1 Motivation

Plusieurs tentatives ont eu pour but de transformer le résultat retourné par les séparateurs à vaste marge (SVM) en une estimation de probabilité [12, 5]. Cependant, il n'existe aucune garantie théorique que ces résultats représentent une

mesure de confiance. De plus, [1] 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 est précis sur une «zone grise», là où les étiquettes des classes changent. L'incertitude de classification est mesurable dans cette zone, puisque les probabilités y sont bien calibrées. Ce classifieur permet aussi d'obtenir des règles de décision appropriées pour des fonctions de perte asymétriques. Se concentrer sur un petit intervalle de probabilités conditionnelles, plutôt que d'effectuer une estimation sur l'ensemble du domaine, a deux avantages : premièrement, l'objectif de l'apprentissage se rapproche de la minimisation du risque de mauvaise classification 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. [1] ont prouvé que, si les probabilités conditionnelles peuvent être estimées partout sans ambiguïté, alors, les modèles à noyau ne peuvent être parcimonieux. Une méthode à noyau est qualifiée ici de parcimonieuse si un nombre limité d'éléments est pris en compte, par exemple, si un nombre important d'exemples d'apprentissage n'ont pas d'influence sur la solution . La parcimonie est source d'efficacité de calcul.

La mesure de l'incertitude de classification et la parcimonie du modèle sont des problèmes importants quand les effectifs des classes sont déséquilibrés, problème qui est la motivation première de ce travail. Lorsqu'il existe une vaste majorité d'exemples négatifs «inintéressants», et seulement quelques exemples appartenant à la classe positive, les résultats d'apprentissage ont tendance à être biaisés en faveur de la classe dominante. Ce biais peut être traité en rééquilibrant la distribution d'apprentissage : les exemples de la classe minoritaire peuvent être répliqués ou générés artificiellement [4], un certain nombre d'exemples de la classe majoritaire peuvent être éliminés. Cependant, d'une part, l'augmentation du nombre d'exemples de la classe minoritaire aboutit à un calcul inefficace, d'autre part, la réduction de la classe majoritaire peut amener à l'élimination d'informations importantes pour la classification. D'autres tactiques appliquent un post-traitement, tel que la modification du biais après l'apprentissage, mais cette technique ne permet pas de découvrir des changements dans la forme ou l'orientation de la frontière de décision que demanderait une séparation de la classe minoritaire. Notre méthode est proche de celles adaptant l'objectif d'apprentissage par l'utilisation de coûts différents pour les exemples positifs et négatifs [11, 8, 15] mais elle diffère de ces dernières qui consistent à l'application de poids différents pour chaque catégorie.

## 2 Critère d'apprentissage

Dans cet article, nous choisissons de traiter seulement du problème de la classification binaire pour simplifier l'exposition, spécialement pour ce qui concerne les algorithmes d'optimisation. Cependant, les critères et les modèles discutés sont par essence multi-classes.

## 2.1 La règle de décision de Bayes

La théorie de la décision de Bayes est le cadre prépondérant dans le cadre de la décision statistique. La règle de décision de Bayes est définie par les vraies probabilités conditionnelles $p(y|\mathbf{x})$ des étiquettes de classes $y$, sachant les caractéristiques $\mathbf{x}$, et les coût de mauvaises décisions. Dans le cas des problèmes binaires, lorsque les classes sont notées $+1$ ou $-1$, il y a deux types d'erreurs possibles : les faux positifs, où les exemples étiquetés $-1$ sont rangés dans la classe positive, induisant un coût $C^-$ ; les faux négatifs, où les exemples étiquetés $+1$ sont rangés dans la classe négative, induisant un coût $C^+$.

L'exemple $\mathbf{x}$ est affecté à la classe positive si l'espérance du coût de cette décision est plus petite que pour le choix opposé : $C^- p(y = -1|\mathbf{x}) \leq C^+ p(y = +1|\mathbf{x})$. La règle est alors

$$\texttt{Décider} \;\; +1 \;\texttt{pour x ssi}\; p(y = 1|\mathbf{x}) \geq \frac{C^-}{C^+ + C^-} \;\; . \tag{1}$$

La règle de décision de Bayes étant définie par les probabilités conditionnelles, beaucoup de classifieurs estiment ces probabilités puis appliquent, sur les probabilités estimées, la règle (1) pour construire la règle de décision. Ces méthodes de classification se différencient par le modèle des probabilités conditionnelles et par la méthode d'estimation, les deux plus importantes étant la méthode des moments (qui motive la minimisation de l'erreur quadratique moyenne de classification) et le maximum de vraisemblance.

## 2.2 Vraisemblance

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

$$\sum_{i:y_i=1} \log(\widehat{p}(y = 1|\mathbf{x}_i)) + \sum_{i:y_i=-1} \log(1 - \widehat{p}(y = 1|\mathbf{x}_i)) \;\; , \tag{2}$$

où $\widehat{p}(y|\mathbf{x})$ est l'estimateur de $p(y|\mathbf{x})$.

## 2.3 Vraisemblance locale

Bien que la règle de décision de Bayes soit définie par $p(y|\mathbf{x})$, 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 (1). Asymptotiquement, c'est ce que réalisent les SVM [1] pour $p(y|\mathbf{x}) = 0.5$, ou pour d'autres probabilités lorsque le critère est asymétrique [11, 8].

La maximisation de la log-vraisemblance (2) demande l'estimation des probabilités conditionnelles sur tout l'intervalle $[0, 1]$. Notre objectif est moins ambitieux : donner une estimation des probabilités conditionnelles sur un intervalle $[p_{\min}, p_{\max}]$. Au-delà de cet intervalle, nous voulons juste savoir si $p(y|\mathbf{x})$ est plus petit que $p_{\min}$ ou plus grand que $p_{\max}$. Ceci peut-être formalisé par la maximisation de

$$\sum_{i:y_i=1} \log\left(\min\left(\widehat{p}(y=1|\mathbf{x}_i), p_{\max}\right)\right) + \sum_{i:y_i=-1} \log\left(\min\left(1-\widehat{p}(y=1|\mathbf{x}_i), 1-p_{\min}\right)\right) \quad , \tag{3}$$

qui est un critère concave en $\widehat{p}(y=1|\mathbf{x}_i)$.

La fonction de perte pour les exemples positifs et négatifs est représentée en figure 1. Le critère est calibré pour la classification si $\frac{C^-}{C^+ + C^-} \in [p_{\min}, p_{\max}]$. Ainsi, si le modèle $\widehat{p}(y=1|\cdot)$ est suffisamment riche, la minimisation de (3) va asymptotiquement fournir un classifieur dont le risque est proche du risque de Bayes (voir [1] pour les définitions et un traitement plus formel).

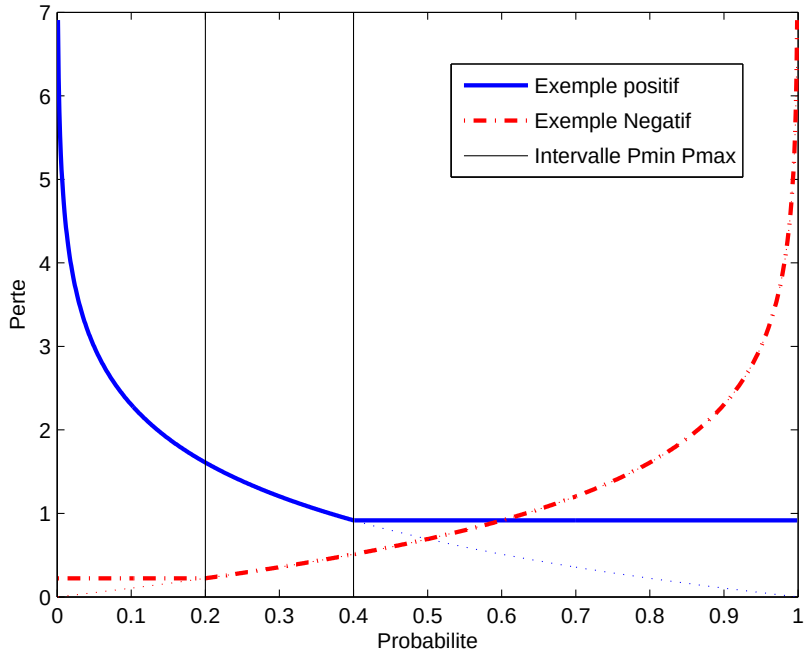


FIG. 1 – Fonction de perte : *traits pleins, tirets* : critère local, *pointillés* : vraisemblance.

# 3 Modèle de probabilité conditionnelle

Nous considérons maintenant un des modèles de probabilité conditionnelle les plus simples. Nous montrons comment ses propriétés sont modifiées par la maximisation de vraisemblance locale.

## 3.1 La régression logistique

La régression logistique est un modèle standard qui considère que le log-ratio des probabilités conditionelles est linéaire.

$$\log \frac{\widehat{p}(y=1|\mathbf{x})}{1-\widehat{p}(y=1|\mathbf{x})} = \mathbf{w}^T\mathbf{x} + b \quad , \tag{4}$$

où les coefficients $(\mathbf{w}, b)$ sont estimés par la maximisation de la vraisemblance (2) ou de la vraisemblance pénalisée.

La régression logistique est similaire à l'analyse discriminante linéaire (ADL) en ce que les deux modèles fournissent le logarithme des rapports des risques (*log-odds*) linéaire. Ils se différencient cependant sur la procédure d'estimation et donc sur les aspects calculatoires. Pour une discussion concise sur les mérites des deux méthodes, le lecteur peut se référer à [6, section 4.4]. D'un point de vue statistique, la régression logistique demande moins de propriétés et est donc plus générale : La densité $p(\mathbf{X})$ est arbitraire alors qu'elle doit être un mélange de gaussienne pour l'ADL. La régression logistique est aussi définie avec moins de paramètres : pour un espace des caractéristiques de dimension $d$, le modèle demande $d$ paramètres alors qu'il demande $\frac{d(d+1)}{2}$ pour l'ADL. Ces différences deviennent importantes dans les espaces de dimension élevée, où l'ADL doit être habituellement appliquée à un ensemble réduit de caractéristiques pré-traitées (comme celles extraites d'une analyse en composantes principales).

## 3.2 La régression logistique à noyaux

Les fonctions à noyaux peuvent être introduites dans la régression logistique en rendant le log-ratio des probabilités conditionnelles non-linéaire

$$\log \frac{\widehat{p}(y=1|\mathbf{x})}{1-\widehat{p}(y=1|\mathbf{x})} = f(\mathbf{x}) + b \ , \tag{5}$$

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

Dans cette configuration, le critère d'apprentissage doit incorporer un terme de régularisation pour se prémunir du surapprentissage [14, 17]. Maximiser la vraisemblance (2) pénalisée par la norme de $f$ revient à minimiser

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

où $\lambda$ est un hyper paramètre, qui peut être ajusté par validation croisée.

Contrairement au SVM, la régression logistique n'a pas une solution parcimonieuse, dans le sens où tous les exemples participent à la solution. En effet, pour une vraisemblance pénalisée (6), les conditions d'optimalité du premier ordre pour $f$ impliquent

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \ , \tag{7}$$
$$\text{avec } \alpha_i = \frac{1}{\lambda}\left(\frac{y_i+1}{2} - \widehat{p}(y=1|\mathbf{x}_i)\right) \ .$$

Comme la définition (5) implique $0 < \widehat{p}(y=1|\mathbf{x}) < 1$, pour tous les exemples $\alpha_i \neq 0$ : le développement exact demande $n$ coefficients.

## 3.3 La régression logistique parcimonieuse

La régression logistique à noyau est difficilement applicable à un grand ensemble de données à cause du nombre de paramètres non nul. [17] proposent de lever ce problème en utilisant un algorithme de sélection glouton recherchant

une approximation du développement complet (7) par un nombre fixé de $\alpha_i$ non-nuls. Cette approche peut être interprétée comme l'ajout d'un terme pénalisant le nombre de coefficients non-nuls dans le critère (6). [13] prend un autre angle d'attaque en remplaçant le terme de pénalisation dans (6) par la norme $\ell_1$ des coefficients $\boldsymbol{\alpha}$.

Notre approche, qui pourrait être combinée à chacune de ces dernières, consiste à remplacer le terme de log-vraisemblance par le critère (3). La régression logistique parcimonieuse à noyau minimise

$$\sum_{i=1}^{n} \log\left(1 + e^{\max(-y_i(f(\mathbf{x}_i)+b), F_i)}\right) + \frac{\lambda}{2}||f||^2_{\mathcal{H}} \ , \tag{8}$$

où $F_i = -\log\frac{p_{\max}}{1-p_{\max}}$ si $y_i = 1$ et $F_i = \log\frac{p_{\min}}{1-p_{\min}}$ si $y_i = -1$.

La parcimonie provient de la troncature du coût. Les exemples d'apprentissage avec une grande valeur de $y_i f(\mathbf{x}_i)$ ne contribuerons pas au classifieur final. Dans ce critère d'apprentissage, c'est le terme d'ajustement et non pas le terme de pénalisation qui provoque la parcimonie. Notons que, par rapport aux approches précédentes, le problème d'optimisation peut être établi sans se référer au développement (7), qui n'apparaîtra que plus tard, comme conséquence des conditions d'optimalité.

# 4 L'apprentissage

La régression logistique à noyau peut être entraînée dans l'espace des variables primales en utilisant la méthode de Newton [14], ou dans l'espace des variables duales [7]. La méthode de Newton est plus simple à dériver, mais même avec la vraisemblance standard, qui ne fournit pas une solution parcimonieuse, [7] rapportent une réduction du temps de calcul importante dans l'espace des variables duales.

## 4.1 La régression logistique parcimonieuse 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 $\mathbf{w}$, en supposant que le vecteur $\mathbf{x}$ inclut un terme constant.

Nous rappelons premièrement la mise à jour de Newton-Raphson pour la régression logistique standard maximisant la vraisemblance

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + (\mathbf{X}^T\mathbf{D}(\mathbf{w}^{(k)})\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{t} - \widehat{\mathbf{p}}(\mathbf{w}^{(k)})) \ ,$$

où $\mathbf{w}^{(k)}$ est le vecteur de paramètres à l'étape $k$, $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n]^T$ est la matrice $n \times d$ des caractéristiques des exemples accumulés, $\mathbf{D}(\mathbf{w})$ est la matrice $n \times n$ diagonale d'élément $i$ égal à $\widehat{p}(y_i = 1|\mathbf{x}_i; \mathbf{w})(1 - \widehat{p}(y_i = 1|\mathbf{x}_i; \mathbf{w})$, $\mathbf{t} = [t_1 \dots t_n]^T$ est le vecteur concaténant les étiquettes des classes des exemples $t_i = \frac{y_i+1}{2}$ et $\widehat{\mathbf{p}}(\mathbf{w}) = [\widehat{p}(y_1 = 1|\mathbf{x}_i; \mathbf{w}) \dots \widehat{p}(y_n = 1|\mathbf{x}_i; \mathbf{w})]^T$.

Pour la vraisemblance locale (3), le critère n'est pas dérivable pour tout $\mathbf{w}$, il peut exister un exemple $i$, tel que $\widehat{p}(y_i = 1|\mathbf{x}_i; \mathbf{w})$ soit égal à $p_{\min}$ ou $p_{\max}$. Nous pouvons remédier à ce problème en approchant cette discontinuité par une fonction deux fois dérivable comme les polynômes utilisés par [3].

## 4.2 La régression logistique parcimonieuse dans l'espace des variables duales

[3] 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 avons en tête, avec un grand déséquilibre entre les classes, on s'attend à ce que la plupart des exemples de la classe majoritaire soit éliminée du développement (7), ainsi on s'attend à une optimisation plus efficace dans l'espace des variables duales.

### 4.2.1 Principe

Nous proposons un algorithme de contraintes actives, suivant une stratégie qui a déjà fait preuve d'efficacité pour les SVM [10]. L'algorithme SimpleSVM [16, 9] résout le problème d'apprentissage des SVM par une approche gloutonne dans laquelle le problème principal est décomposé en une série de petits problèmes. La répartition des exemples dans l'ensemble des vecteurs supports et non-supports étant connue, le critère d'apprentissage est optimisé considérant cette partition fixe. De l'optimisation résulte une nouvelle partition des exemples en vecteurs supports (ensemble actif) et non-supports (ensemble inactif). Ces deux étapes sont répétées jusqu'a ce qu'un certain niveau de précision soit atteint [10].

Nous utilisons la même stratégie. Nous présentons, dans un premier temps, la formulation duale de la régression logistique parcimonieuse. Puis, considérant que la partition entre exemples actifs et de 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.

### 4.2.2 Formulation duale

Comme dans la formulation duale des SVM, nous traitons la discontinuité introduite par la fonction max dans (8) grâce à l'introduction de variables d'écart $\boldsymbol{\xi}$

$$\begin{array}{lll} \min\limits_{f,\boldsymbol{\xi},b} & \dfrac{\lambda}{2}||f||^2_{\mathcal{H}} + \displaystyle\sum_{i=1}^{n} \log(1+e^{\xi_i}) & \\ \text{t. q.} & \xi_i \geq -y_i(f(\mathbf{x}_i)+b) & i=1,\ldots,n \\ & \xi_i \geq F_i & i=1,\ldots,n \ , \end{array} \tag{9}$$

avec $F_i = -f_{\max} = -\log\left(\frac{p_{\max}}{1-p_{\max}}\right)$ si $y_i = 1$ et $F_i = f_{\min} = \log\left(\frac{p_{\min}}{1-p_{\min}}\right)$ si $y_i = -1$.

Le lagrangien de ce problème convexe est

$$L = \frac{\lambda}{2}||f||^2_{\mathcal{H}} + \sum_{i=1}^{n} \log(1+e^{\xi_i}) + \sum_{i=1}^{n} \beta_i(F_i - \xi_i) - \sum_{i=1}^{n} \alpha_i[y_i(f(\mathbf{x}_i)+b)+\xi_i] \ . \tag{10}$$

La solution de (9) est atteinte au point col du lagrangien (10). Les conditions

de Kuhn-Tucker impliquent

$$\begin{aligned}
\nabla_f L &= \lambda f(\mathbf{x}) - \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) = 0 \\
\frac{\partial L}{\partial b} &= -\sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\partial L}{\partial \xi_i} &= \frac{1}{1+e^{-\xi_i}} - (\alpha_i + \beta_i) = 0 \ ,
\end{aligned}$$

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

$$\begin{aligned}
L &= -\frac{1}{2\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_j, \mathbf{x}_i) \qquad (11) \\
&\quad - \sum_{i=1}^{n} (\alpha_i + \beta_i) \log(\alpha_i + \beta_i) + (1 - \alpha_i - \beta_i) \log(1 - \alpha_i - \beta_i) + \sum_{i=1}^{n} \beta_i F_i \ .
\end{aligned}$$

Cette expression implique $2n$ variables, mais elle peut être simplifiée grâce au partitionnement de l'ensemble d'apprentissage.

### 4.2.3 Partitionnement de l'ensemble d'apprentissage

Nous séparons l'ensemble d'apprentissage en trois ensembles suivant les contraintes de (9). Les exemples identifiés par :

- $I_0$ sont dans la partie constante du coût où $-y_i(f(\mathbf{x}_i) + b) < F_i$ ;
- $I_h$ sont sur le point charnière du coût, où les deux contraintes peuvent être actives puisque $-y_i(f(\mathbf{x}_i) + b) = F_i$ ;
- $I_\ell$ sont dans la partie logarithmique du coût où $-y_i(f(\mathbf{x}_i) + b) > F_i$.

La table 1 décrit les propriétés de chaque ensemble, en ce qui concerne la variable originale $\xi_i$ et les multiplicateurs de Lagrange $\alpha_i$ et $\beta_i$.

Supposons que la répartition entre chaque ensemble est connue. La partie non quadratique du lagrangien est répartie en trois composant

$$L = -\frac{1}{2\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_j, \mathbf{x}_i) + L_0 + L_h + L_\ell \ .$$

Pour chaque ensemble, les expressions $\alpha_i$ et $\beta_i$ (extraites de la table 1) sont

TAB. 1 – valeurs de $\xi_i$, $\alpha_i$ et $\beta_i$ pour les trois ensembles

| Ensemble | $\xi_i$ | $\alpha_i$ | $\beta_i$ |
|---|---|---|---|
| $I_0$ | $F_i$ | $0$ | $\frac{1}{1+e^{-F_i}}$ |
| $I_h$ | $F_i$ | $\frac{1}{1+e^{-F_i}} - \beta_i$ | $\frac{1}{1+e^{-F_i}} - \alpha_i$ |
| $I_\ell$ | $-y_i(f(\mathbf{x}_i) + b)$ | $\frac{1}{1+e^{y_i(f(\mathbf{x}_i)+b)}}$ | $0$ |

introduites dans l'équation (11), de telle façon que nous obtenons

$$\begin{aligned} L_0 &= \sum_{i\in I_0} \log(1+e^{F_i}) \\ L_h &= \sum_{i\in I_h} \log(1+e^{F_i}) - \sum_{i\in I_h} \alpha_i F_i \\ L_\ell &= -\sum_{i\in I_\ell} \alpha_i \log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) \ . \end{aligned}$$

Nous remarquons que $L_0$ et que le premier terme de $L_h$ sont constants et indépendants de $\boldsymbol{\alpha}$ ou de $\boldsymbol{\beta}$. De plus, comme $\alpha_i = 0$ pour $i \in I_0$, le terme quadratique peut être réduit aux exemples de l'ensemble actif $I_{\bar{0}} = I_h \cup I_\ell$. En enlevant les termes constants, nous obtenons

$$\begin{aligned} L &= -\frac{1}{2\lambda} \sum_{(i,j)\in {I_{\bar{0}}}^2} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_j,\mathbf{x}_i) - \sum_{i\in I_h} \alpha_i F_i \\ &\quad - \sum_{i\in I_\ell} \alpha_i \log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) \ , \end{aligned}$$

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

### 4.2.4 Optimisation des multiplicateurs de Lagrange

Le problème d'optimisation peut être maintenant reformulé en

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2\lambda} \sum_{(i,j)\in {I_{\bar{0}}}^2} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_j,\mathbf{x}_i) + \sum_{i\in I_h} \alpha_i F_i \\ & + \sum_{i\in I_\ell} \alpha_i \log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) \\ \text{t. q.} \quad & \sum_{i\in I_{\bar{0}}} \alpha_i y_i = 0 \qquad i \in I_{\bar{0}} \ . \end{aligned} \tag{12}$$

Pour tout $i \in I_\ell$, $\alpha_i$ est restreint au domaine de $L$, *i.e.* $0 < \alpha_i < 1$. De plus, puisque l'appartenance des exemples à $I_0$, $I_h$ ou $I_\ell$ est supposée connue, les contraintes $\alpha_i$ impliquent

$$\begin{aligned} &\forall i \in I_0 \quad \alpha_i = 0 \\ &\forall i \in I_h \quad 0 \le \alpha_i \le \tfrac{1}{1+e^{-F_i}} \\ &\forall i \in I_\ell \quad \alpha_i \ge \tfrac{1}{1+e^{-F_i}} \qquad . \end{aligned} \tag{13}$$

Le problème (12) étant convexe sous contraintes linéaires, il peut être résolu efficacement par la méthode de Newton [2]. Nous écrivons premièrement son lagrangien

$$\begin{aligned} \bar{L} &= \frac{1}{2\lambda} \sum_{(i,j)\in {I_{\bar{0}}}^2} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_j,\mathbf{x}_i) + \sum_{i\in I_h} \alpha_i F_i \\ &\quad +\gamma \sum_{i\in I_{\bar{0}}} \alpha_i y_i + \sum_{i\in I_\ell} \alpha_i \log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i) \ . \end{aligned} \tag{14}$$

Soit $G$ une matrice ($|I_{\bar{0}}| \times |I_{\bar{0}}|$) de terme général $G_{ij} = \frac{1}{\lambda} y_i y_j K(\mathbf{x}_j, \mathbf{x}_i)$, les conditions Kuhn-Tucker $\frac{\partial \bar{L}}{\partial \alpha_i} = 0$ et $\frac{\partial \bar{L}}{\partial \gamma} = 0$ s'écrivent

$$\begin{aligned} &\forall i \in I_h, \sum_{j \in I_{\bar{0}}} \alpha_j G_{ij} + F_i + \gamma y_i = 0 \\ &\forall i \in I_\ell, \sum_{j \in I_{\bar{0}}} \alpha_j G_{ij} + \log\left(\frac{\alpha_i}{1-\alpha_i}\right) + \gamma y_i = 0 \\ &\qquad \sum_{i \in I_{\bar{0}}} \alpha_i y_i = 0 \ . \end{aligned} \tag{15}$$

Ces conditions forment un système non-linéaire qui peut être résolu itérativement par la méthode de Newton. En faisant un léger abus de notation pour ne pas indexer les variables de travail, nous noterons $\boldsymbol{\alpha}$, $\mathbf{y}$, $\mathbf{F}$ et $G$ comme suit

$$\begin{aligned} \boldsymbol{\alpha} &= \left[\boldsymbol{\alpha}_h^T \ \boldsymbol{\alpha}_\ell^T\right]^T & \mathbf{y} &= \left[\mathbf{y}_h^T \ \mathbf{y}_\ell^T\right]^T \\ \mathbf{F} &= \left[\mathbf{F}_h^T \ \mathbf{F}_\ell^T\right]^T & G &= \begin{bmatrix} G_{h,h} & G_{\ell,h} \\ G_{\ell,h}^T & G_{\ell,\ell} \end{bmatrix} \end{aligned}$$

où $\mathbf{v_h}$ dénote le vecteur formé des composantes $\mathbf{v_i}$ pour $i \in I_h$. Le gradient s'écrit alors

$$\nabla \bar{L}(\boldsymbol{\alpha}, \gamma) = \begin{bmatrix} G_{h,h} & G_{\ell,h} & \mathbf{y}_h \\ G_{\ell,h}^T & G_{\ell,\ell} + D(\boldsymbol{\alpha}_\ell) & \mathbf{y}_\ell \\ \mathbf{y}_h^T & \mathbf{y}_\ell^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_h \\ \boldsymbol{\alpha}_\ell \\ \gamma \end{bmatrix} + \begin{bmatrix} \mathbf{F}_h \\ 0 \\ 0 \end{bmatrix} \ ,$$

où $D(\boldsymbol{\alpha}_\ell)$ est une matrice diagonale ($|I_\ell| \times |I_\ell|$) avec comme éléments diagonaux $\log\left(\frac{\boldsymbol{\alpha}_\ell}{\mathbf{1}-\boldsymbol{\alpha}_\ell}\right)$. Nous obtenons alors la hessienne $\bar{L}$ par rapport à $\left[\boldsymbol{\alpha}^T \gamma\right]^T$

$$\nabla^2 \bar{L}(\boldsymbol{\alpha}, \gamma) = \begin{bmatrix} G_{h,h} & G_{\ell,h} & \mathbf{y}_h \\ G_{\ell,h}^T & G_{\ell,\ell} + D'(\boldsymbol{\alpha}_\ell) & \mathbf{y}_\ell \\ \mathbf{y}_h^T & \mathbf{y}_\ell^T & 0 \end{bmatrix} \ , \tag{16}$$

où $D'(\boldsymbol{\alpha}_\ell)$ est une matrice diagonale ($|I_\ell| \times |I_\ell|$) avec comme éléments diagonaux $\frac{1}{\boldsymbol{\alpha}_\ell(\mathbf{1}-\boldsymbol{\alpha}_\ell)}$. Remarquons que, puisque $0 < \alpha_i < 1$ pour $i \in I_\ell$, la hessienne est définie positive pour peu que le noyau $K$ soit défini positif.

L'étape du Newton consiste à résoudre

$$\begin{aligned} \nabla^2 \bar{L}(\boldsymbol{\alpha}^{(k)}, \gamma^{(k)})[(\boldsymbol{\alpha}^{(k+1)} - \boldsymbol{\alpha}^{(k)})^T (\gamma^{(k+1)} - \gamma^{(k)})]^T \\ = -\nabla \bar{L}(\boldsymbol{\alpha}^{(k)}, \gamma^{(k)}) \ , \end{aligned}$$

c'est à dire

$$\begin{bmatrix} \boldsymbol{\alpha}_h^{(k+1)} \\ \boldsymbol{\alpha}_\ell^{(k+1)} \\ \gamma^{(k+1)} \end{bmatrix} = \left[\nabla^2 \bar{L}(\boldsymbol{\alpha}^{(k)}, \gamma^{(k)})\right]^{-1} \begin{bmatrix} -\mathbf{F}_h \\ \boldsymbol{\delta}_\ell^{(k)} \\ 0 \end{bmatrix} \ , \tag{17}$$

avec $\boldsymbol{\delta}_\ell^{(k)} = \left(D'(\boldsymbol{\alpha}_\ell^{(k)}) - D(\boldsymbol{\alpha}_\ell^{(k)})\right) \boldsymbol{\alpha}_\ell^{(k)}$. Les étapes de Newton sont répétées jusqu'à convergence ou jusqu'à ce que les partitions $I_0$, $I_h$ et $I_\ell$ doivent être modifiées.

#### 4.2.5 Mise à jour de la répartition

Étant donnée une répartition, chaque étape de la méthode de Newton retourne une solution améliorée. Cette dernière doit obéir aux contraintes (13) pour être consistante avec la conjecture initiale. Si ce n'est pas le cas, cette solution doit être remaniée. Si les variables sortent de l'ensemble actif, le calcul est corrigé en limitant la taille du pas dans la direction de descente. Pour ce faire, on calcule le plus grand pas $\rho$ tel que $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k)} + \rho(\boldsymbol{\alpha}^{(k+1)} - \boldsymbol{\alpha}^{(k)})$ satisfasse (13). Une fois cela fait, notant $i$ le(s) composant(s) en faute de $\boldsymbol{\alpha}$, la répartition est modifiée comme suit :

- si $i \in I_h$ et $\alpha_i = 0$, $i$ est déplacé en $I_0$ ;
- si $i \in I_h$ et $\alpha_i = \frac{1}{1+e^{-F_i}}$, $i$ est déplacé en $I_\ell$ ;
- si $i \in I_\ell$ et $\alpha_i = \frac{1}{1+e^{-F_i}}$, $i$ est déplacé en $I_h$.

Partant du $\boldsymbol{\alpha}$ courant, l'étape de Newton est appliquée avec la nouvelle répartition, et la procédure est itérée jusqu'à satisfaction de toutes les contraintes.

Lorsqu'un point fixe de (17) est atteint et qu'aucune contrainte n'est violée par les exemples de l'ensemble actif, nous pouvons alors procéder à l'inclusion d'un nouveau candidat de l'ensemble inactif $I_0$ à l'ensemble actif. Tout exemple $i \in I_0$ tel que $-y_i(f(\mathbf{x}_i) + b) > F_i$ est candidat. Nous choisissons simplement celui qui maximise $-y_i(f(\mathbf{x}_i) + b) - F_i$. Inclure un exemple à la fois permet une mise à jour efficace de la décomposition de Cholesky de la matrice hessienne (16). 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 lesquelles $-y_i(f(\mathbf{x}_i) + b) = F_i$. D'autre part, comme $f(\mathbf{x}_i) = \sum_{j \in I_{\bar{0}}} \alpha_j G_{ij}$, par identification avec l'équation (15), nous voyons que $b = \gamma$.

# 5 Expérimentations

Pour étudier le problème de deux classes déséquilibrées, nous avons choisi la base de donnée Forest, la plus grande base de données de l'UCI. [1] 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 choisit 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 $\frac{C^-}{C^+ + C^-} = \pi^+$, où $\pi^+ = 8.8\%$ est la proportion d'exemples positifs. Les pertes sont alors définies à un facteur près, et nous choisissons $C^- = \pi^+$ et $C^+ = 1 - \pi^+$ .

## 5.1 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

[1] Disponible sur `kdd.ics.uci.edu/databases/covertype`.

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 $\lambda$ pour la régression logistique (6) et la régression logistique parcimonieuse (8) 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 $\pi^+$ sur l'échelle logarithmique, c'est à dire $\sqrt{p_{\min} p_{\max}} = \pi^+$.

## 5.2 Résultats

Nous rapportons les performances moyennes de la régression logistique parcimonieuse, ainsi que leur écart-type dans la table 2. 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 petit 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 2 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 $\pi^+$, reflétant des probabilités conditionnelles bien calibrées dans la région ciblée.

La table 3 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 $C^+ \neq C^-$ [11]. 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

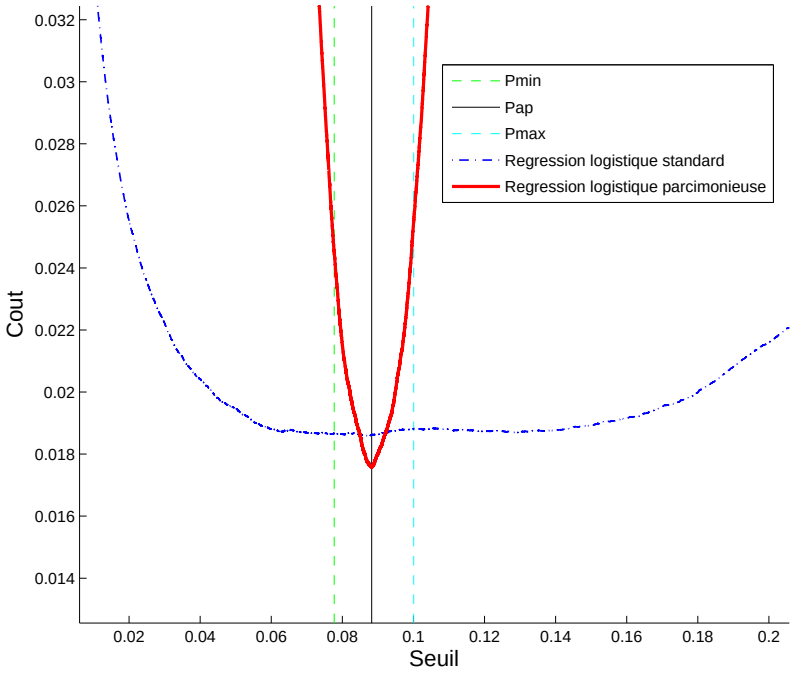


FIG. 2 – Coût de test en fonction du seuil de décision.

TAB. 2 – 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 | $\pm 0.01$ | $\pm 0.01$ | $\pm 0.01$ | $\pm 0.01$ | $\pm 0.02$ | $\pm 0.02$ |
| Seuil moyen de décision (%) | 9.5 | 9.0 | 9.0 | 9.0 | 8.8 | 8.8 |
| Écart-type | $\pm 1.3$ | $\pm 1.1$ | $\pm 0.9$ | $\pm 0.6$ | $\pm 0.2$ | $\pm 0.0$ |
| Prop. moyenne de SV (%) | 100 | 65.5 | 53.5 | 40.5 | 34.0 | 27.9 |

TAB. 3 – Coûts moyens de test obtenus pour les SVMs

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

(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.

# 6 Discussion

Nous avons proposé un nouveau critère d'apprentissage, qui consiste à tronquer la log-vraisemblance binomiale. Ce critère produit un classifieur probabiliste parcimonieux, qui fournit des probabilités conditionnelles fiables au voisinage de la frontière de décision. Nous avons examiné en détail comment la régression logistique est modifiée par la maximisation de «la 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 multiclasse et peut être appliqué à une log-vraisemblance multinomiale. Le problème d'optimisation résultant reste un problème convexe pourvu que l'intervalle «intéressant» des probabilités conditionnelles soit un ensemble convexe.

Des expériences sont en cours pour confirmer l'intérêt pratique des classifieurs probabilistes parcimonieux, 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-échantillonage 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 aussi elle est plus rapide à entraîner, grâce à sa capacité d'ignorer les données non pertinentes. Les performances et les temps d'apprentissage sont comparables aux SVM entrainés avec des coûts asymétriques $C^+/C^-$, et nous pouvons profiter en plus de probabilités bien calibrées dans le voisinage de la frontière de décision.

# Références


[1] P. L. Bartlett and A. Tewari. Sparseness vs estimating conditional probabilities : Some asymptotic results. In *Proceedings of the 17th Annual Conference on Learning Theory*, volume 3120 of *Lecture Notes in Computer Science*, pages 564–578. Springer, 2004.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[3] O. Chapelle. Training a support vector machine in the primal. *Neural Computation (accepted)*, 2007.

[4] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. Smote : Synthetic minority oversampling technique. *Journal Of Artificial Intelligence Research*, 16 :321–357, 2002.

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

[6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning : data mining , inference, and prediction*. Springer series in statistics. Springer, 2001.

[7] S. S. Keerthi, K. B. Duan, S. K. Shevade, and A. N. Poo. A fast dual algorithm for kernel logistic regression. *Mach. Learn.*, 61(1-3) :151–165, 2005.

[8] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Mach. Learn.*, 46(1-3) :191–202, 2002.

[9] G. Loosli. SimpleSVM toolbox, 2005. http ://asi.insa-rouen.fr/ gloosli/simpleSVM.html.

[10] G. Loosli and S. Canu. Comments on the "core vector machines : Fast SVM training on very large data sets". *Journal of Machine Learning Research*, pages 291–301, Feb. 2007.

[11] E. Osuna, R. Freund, and F. Girosi. Support vector machines : Training and applications. Technical Report A.I. Memo No. 1602, M.I.T. AI Laboratory, 1997.

[12] J. C. Platt. Probabilities for SV machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[13] V. Roth. The generalized LASSO. *IEEE Transactions on Neural Networks*, 15(1) :16–28, 2004.

[14] Volker Roth. Probabilistic discriminative kernel classifiers for multi-class problems. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 246–253, London, UK, 2001. Springer-Verlag.

[15] Kai Ming Ting. A comparative study of cost-sensitive boosting algorithms. In *Proc. 17th International Conf. on Machine Learning*, pages 983–990. Morgan Kaufmann, 2000.

[16] S. V. N. Vishwanathan, Alex J. Smola, and M. Narasimha Murty. SimpleSVM. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 760–767, 2003.

[17] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. In *Advances in Neural Information Processing Systems 13*, 2001.