IDIAP RESEARCH REPORT

Sparse Probabilistic Classifiers

Romain Hérault [1] Yves Grandvalet [2]

IDIAP–RR 07-19

April 25, 2007

to appear in
Proceedings of the 24th International Conference on Machine Learning,
Corvallis, OR, 2007

[1] HEUDIASYC, UMR 6599, Universit de Technologie de Compigne, BP 20529, 60205 Compigne cedex, France, romain.herault@hds.utc.fr
[4] IDIAP Research Institute, CP 592, 1920 Martigny, Switzerland, yves.grandvalet@idiap.ch

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 07-19

Sparse Probabilistic Classifiers

Romain Hérault Yves Grandvalet

April 25, 2007

to appear in

Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, 2007

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 argument 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 also 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 and performances similar to support vector machines.

Contents

1 Motivation

There have been several attempts to turn the scores returned by support vector machines (SVMs) 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 & Tewari, 2004). 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, or equivalently, for the corresponding cone of the ROC curve. 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 and Tewari (2004) proved that sparsity does not occur when the conditional probabilities can be unambiguously estimated everywhere. 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.

The assessment of classification uncertainty and the sparsity of the model are important issues for the class imbalance problem, which is our original motivation for this work. 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, either by over-sampling the minority class, or generating artificial examples of the minority class (Chawla et al., 2002), or by down-sampling the majority class. However, undersampling may discard relevant pieces of information and oversampling is not computationally efficient. Another tactic consists in post-processing standard classification techniques, such as tuning a bias term after learning to correct for the original decision bias, but this scheme fails to discover the changes in the shape or orientation of the decision boundary that may be required to isolate the minority class. Our approach is more 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.

2 Learning criterion

In this paper, we chose to consider only binary classification problems to keep exposition simple, especially regarding the description of optimization algorithms. However, the discussed criteria and models are inherently multi-class.

2.1 Bayes decision rule

Bayes' decision theory is the paramount framework in statistical decision theory. The Bayes decision rule is defined by the true conditional probabilities $p(y|\mathbf{x})$ of class labels y, knowing features $\mathbf{x}$, and by misclassification losses. In binary problems, where the class is tagged $+1$ or -1, the two types of errors are: false positive, where examples labelled -1 are categorized in the positive class, incurring a loss C^-; false negative, where examples labelled $+1$ are categorized in the negative class, incurring a loss C^+.

Pattern $\mathbf{x}$ is then affected to the positive class provided that the expected loss incurred by this decision is smaller than the opposite choice $C^- p(y=-1|\mathbf{x}) \leq C^+ p(y=+1|\mathbf{x})$. The rule is then

$$\texttt{Classify } \mathbf{x} \texttt{ as } +1 \texttt{ iff } p(y=1|\mathbf{x}) \geq \frac{C^-}{C^+ + C^-} \ . \tag{1}$$

Many classifiers first estimate conditional probabilities, and then plug this estimate in (1) to build the decision rule. These classification methods then differ by the functional space used to model conditional probabilities, and by the estimation method, the two mainstream ones being the methods of moments (leading to the minimization of mean squared error in classification) and maximum likelihood.

2.2 Maximum likelihood

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

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

where $\widehat{p}(y|\mathbf{x})$ denotes the estimate of $p(y|\mathbf{x})$.

2.3 Lazy maximum likelihood

Although Bayes' decision rule is defined in terms of $p(y|\mathbf{x})$, 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 (1). This is precisely what SVMs achieve asymptotically (Bartlett & Tewari, 2004) for $p(y|\mathbf{x}) = 0.5$.

Maximizing the log-likelihood (2) amounts to estimate the conditional probabilities on the full range $[0, 1]$. We consider here a weaker estimation problem, where we focus on a small range $[p_{\min}, p_{\max}]$. Outside of this range, we only want to know whether $p(y|\mathbf{x})$ is smaller than $p_{\min}$ or greater than $p_{\max}$. This optimization problem can be formalized as maximizing

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

which is a concave criterion in $\widehat{p}(y=1|\mathbf{x}_i)$. Note that $p_{\min}$ and $p_{\max}$ are tuning parameters of the fitting criterion; they do not enter in the definition of conditional probabilities $\widehat{p}(y=\pm 1|\mathbf{x}_i)$.

This criterion is classification-calibrated provided $\frac{C^-}{C^+ + C^-} \in [p_{\min}, p_{\max}]$. Hence, provided the model $\widehat{p}(y=1|\cdot)$ is rich enough, minimizing (3) will asymptotically provide a classifier with risk close to the Bayes' risk (see Bartlett and Tewari (2004) for definitions and more formal statements).

3 Conditional probability model

We now consider here one of the simplest model of conditional probabilities. We show how its properties are modified by lazy maximum likelihood estimation.

3.1 Logistic regression

Logistic regression is a standard probabilistic model which considers that the log-ratio of conditional probabilities is linear

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

and where the coefficients $(\mathbf{w}, b)$ are estimated by maximizing the likelihood (2) or the penalized likelihood.

Logistic regression is similar to linear discriminant analysis (LDA) in that both models provide linear log-odds (4). They differ however with respect to the estimation process and therefore in computational aspects. From a statistical point of view, logistic regression makes less assumptions and is thus more general: the density $p(\mathbf{X})$ is arbitrary while it is assumed to be a Gaussian mixture in LDA.

3.2 Kernel logistic regression

Logistic regression can be kernelized, by letting the log-ratio of conditional probabilities to be non-linear

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

where f is a function belonging to a given reproducing kernel Hilbert space $\mathcal{H}$.

In this setting, the training criterion should incorporate a regularization term to prevent overfitting (Roth, 2001; Zhu & Hastie, 2001). Maximizing the likelihood (2) penalized by the norm of f results in minimizing

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

where λ is a hyper-parameter that may be tuned by cross-validation.

Unlike SVMs, logistic regression does not yield sparse solutions, in the sense that all examples influence the solution. Indeed, for the penalized likelihood (6), the first order optimality conditions for f imply

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \ , \tag{7}$$

with $\alpha_i = \frac{1}{\lambda}\left(\frac{y_i+1}{2} - \widehat{p}(y=1|\mathbf{x}_i)\right)$.

By definition (5), $0 < \widehat{p}(y=1|\mathbf{x}) < 1$, hence for all examples, $\alpha_i \neq 0$: the exact expansion requires n coefficients.

3.3 Sparse logistic regression

Kernel logistic regression is hardly applicable to large data sets due to the number of non-zero parameters. Zhu and Hastie (2001) propose to alleviate this problem by using a greedy forward selection algorithm looking for an approximation of the full expansion (7) involving a fixed number of non-zero α_i. This approach can be interpreted as the adding to the criterion (6) an extra term penalizing the number of non-zero coefficients. Roth (2004) takes another line of attack by replacing the penalization term in (6) by the ℓ_1 norm of coefficients $\boldsymbol{\alpha}$. Our approach, which could be combined with any of these two, consists in replacing the log-likelihood term by criterion (3). Sparse kernelized logistic regression minimizes

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

where $F_i = -\log \frac{p_{\max}}{1-p_{\max}}$ if $y_i = 1$ and $F_i = \log \frac{p_{\min}}{1-p_{\min}}$ if $y_i = -1$.

Sparsity follows from the truncation of the loss. Training examples with large values of $y_i f(\mathbf{x}_i)$ will not contribute to the final classifier. In this training criterion, it is the fitting term, instead of the penalization term, which causes sparsity. Note that compared to these previous approaches, the optimization problem can be stated without referring to the expansion (7), which will only arise at a later stage, as a consequence of the optimality conditions.

Sparsity could also be improved by using the generaic methods developed for kernel machines (Wu et al., 2006). In particular, the ramp loss (Collobert et al., 2006) could be adapted to our framework by always saturating the loss outside of the $[p_{\min}, p_{\max}]$ interval, that is, by letting positive examples with $p(y|\mathbf{x})$ smaller than $p_{\min}$ having a loss of$-\log(p_{\min})$ and negative examples with $p(y|\mathbf{x})$ greater than $p_{\max}$ having a loss of $-\log(1 - p_{\max})$.

3.4 Relation to other methods

The robust logistic model of Cox and Pearce (1997) estimates the range of log-likelihood ratio such that the logistic function fits conditional probabilities. Compared to our approach, this scheme differs by assuming a roughly constant likelihood ratio outside of the estimated range, while we only assume that $p(y|\mathbf{x})$ is smaller than $p_{\min}$ or greater than $p_{\max}$.

Chakrabartty and Cauwenberghs (to appear) proposed a sparse probabilistic classifier based to a quadratic approximation of maximum entropy discrimination. The mechanism driving sparsity is however quite different, since it relies on saturating conditional probabilities. Hence, the maximum sparsity is obtained when the conditional probabilities are approximated by hard assignments, whereas in our scheme, the maximum sparsity is obtained when the $[p_{\min}, p_{\max}]$ interval of precise conditional probabilities collapses to the single value defining the decision boundary.

4 Training

Kernel logistic regression can be learned in the primal using Newton's method (Roth, 2001), or in the dual (Keerthi et al., 2005). Newton's method is simpler to derive, but even with the standard likelihood, which does not provide sparse solution, Keerthi et al. (2005) report impressive reduction in computational time.

4.1 Sparse logistic regression in the primal

For simplicity sake, we consider here the linear logistic regression model, without regularization, where the bias term (or intercept) b is included here in $\mathbf{w}$, assuming that $\mathbf{x}$ includes a constant term feature.

We first recall the Newton-Raphson update for the standard logistic regression maximizing likelihood

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

where $\mathbf{w}^{(k)}$ is the vector of parameters at step k, $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n]^T$ is the $n \times d$ matrix of stacked patterns, $\mathbf{D}(\mathbf{w})$ is a $n \times n$ diagonal matrix with i-th element equal to $\widehat{p}(y_i = 1|\mathbf{x}_i; \mathbf{w})(1 - \widehat{p}(y_i = 1|\mathbf{x}_i; \mathbf{w})$, $\mathbf{t} = [t_1 \dots t_n]^T$ is the vector of stacked binary targets $t_i = \frac{y_i+1}{2}$ and $\widehat{\mathbf{p}}(\mathbf{w}) = [\widehat{p}(y_1 = 1|\mathbf{x}_i; \mathbf{w}) \dots \widehat{p}(y_n = 1|\mathbf{x}_i; \mathbf{w})]^T$.

For the truncated likelihood (3), the criterion is not differentiable for all $\mathbf{w}$, for which there exists an example i, such as $\widehat{p}(y_i = 1|\mathbf{x}_i; \mathbf{w})$ equals $p_{\min}$ or $p_{\max}$. This problem can be remedied by approximating these discontinuities by a polynomial as in (Chapelle, 2007).

4.2 Sparse logistic regression in the dual

Chapelle (2007) argues about the prevalence of dual algorithms for optimizing kernel machines. However, dual optimization takes great advantage of the sparsity arising from the flat part of the loss, while they cause difficulties with second order methods in the primal formulation. For the kind of application we have in mind, with a high imbalance between classes, most examples from the majority class are expected to be discarded from the expansion (7), hence optimization in the dual is expected to be computationally efficient.

4.2.1 Principle

We propose an active set algorithm, following a strategy that proved to be efficient for SVMs. The SimpleSVM algorithm (Vishwanathan et al., 2003; Loosli & Canu, to appear) 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 & Canu, to appear).

We will follow the same strategy. We first present the dual formulation of sparse logistic regression. Next, assuming that the current membership of the example in the active set are correct, we derive the optimal update of parameters. Then, we show how to update the active set based on the parameters update, and sum up the algorithm.

4.2.2 Dual formulation

As in the SVM dual formulation, we handle the discontinuity introduced by the max function in (8) by introducing slack variables $\boldsymbol{\xi}$

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

with $F_i = -f_{\max} = -\log\left(\frac{p_{\max}}{1-p_{\max}}\right)$ if $y_i = 1$ and $F_i = f_{\min} = \log\left(\frac{p_{\min}}{1-p_{\min}}\right)$ if $y_i = -1$.

The Lagrangian of this convex problem is

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

The solution of (9) is reached at the saddle point of the Lagrangian (10). The Kuhn-Tucker conditions imply

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

where $K(\cdot,\cdot)$ is the reproducing kernel of the Hilbert space $\mathcal{H}$.

Thanks to these conditions, we can eliminate f and $\boldsymbol{\xi}$ from the Lagrangian

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

This expression involves $2n$ variables, but it can be simplified thanks to the partitioning of training examples.

Table 1: ξ_i, α_i and β_i for the three sets of examples

Set	ξ_i	α_i	β_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_ℓ	$-y_i(f(\mathbf{x}_i)+b)$	$\frac{1}{1+e^{y_i(f(\mathbf{x}_i)+b)}}$	0

4.2.3 Partition of the training set

We partition the training set into three sets according to the constraints in (9). The examples indexed by

I_0 are in the saturated part of the loss where $-y_i(f(\mathbf{x}_i)+b) < F_i$;

I_h are at the hinge of the loss, where the two constraints may be active since $-y_i(f(\mathbf{x}_i)+b) = F_i$;

I_ℓ are in the logarithmic part of the loss where $-y_i(f(\mathbf{x}_i)+b) > F_i$.

Table 1 describes the properties of each set, regarding the original variables ξ_i and the Lagrange multipliers α_i and β_i.

For the time being, we assume the repartition in each set to be known. The non-quadratic part of the Lagrangian is decomposed into three components

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

For each set, the expressions of α_i and β_i (extracted from Table 1) are plugged in eq. (11), so that we get

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

We note that L_0 and the first term of L_h are constants, independent of $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$. Furthermore, as $\alpha_i = 0$ for $i \in I_0$, the quadratic term can be restricted to the examples in the active set $I_{\bar{0}} = I_h \cup I_\ell$. Discarding the constant terms provides

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

which only involves $|I_{\bar{0}}| < n$ active variables.

4.2.4 Optimizing the Lagrange multipliers

The optimization problem is now reformulated as

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

For all $i \in I_\ell$, α_i is restricted to the domain of L, *i.e.* $0 < \alpha_i < 1$. Furthermore, since the allocation of every example to I_0, I_h and I_ℓ is assumed to be known, the following box constraints on α_i hold implicitly:

$$\begin{array}{ll} \forall i \in I_0 & \alpha_i = 0 \\ \forall i \in I_h & 0 \leq \alpha_i \leq \frac{1}{1+e^{-F_i}} \\ \forall i \in I_\ell & \alpha_i \geq \frac{1}{1+e^{-F_i}} \end{array} \enspace . \tag{13}$$

Problem (12) being convex, with linear constraints, it can be efficiently solved by Newton's method (Boyd & Vandenberghe, 2004). We first write the Lagrangian:

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

Let G be a $(|I_{\bar{0}}| \times |I_{\bar{0}}|)$ matrix with general term $G_{ij} = \frac{1}{\lambda} y_i y_j K(\mathbf{x}_j, \mathbf{x}_i)$, the Kuhn-Tucker conditions $\frac{\partial \bar{L}}{\partial \alpha_i} = 0$ and $\frac{\partial \bar{L}}{\partial \gamma} = 0$ read

$$\begin{aligned} & \forall i \in I_h, \sum_{j \in I_{\bar{0}}} \alpha_j G_{ij} + F_i + \gamma y_i = 0 \\ & \forall i \in I_\ell, \sum_{j \in I_{\bar{0}}} \alpha_j G_{ij} + \log\left(\frac{\alpha_i}{1 - \alpha_i}\right) + \gamma y_i = 0 \\ & \sum_{i \in I_{\bar{0}}} \alpha_i y_i = 0 \end{aligned} \tag{14}$$

These conditions form a non-linear system that can be solved iteratively by Newton's method. We first write the gradient in vectorial form, where, slightly abusing notations, we partition $\boldsymbol{\alpha} = \left[\boldsymbol{\alpha}_h^T \; \boldsymbol{\alpha}_\ell^T\right]^T$, $\mathbf{y} = \left[\mathbf{y}_h^T \; \mathbf{y}_\ell^T\right]^T$, $\mathbf{F} = \left[\mathbf{F}_h^T \; \mathbf{F}_\ell^T\right]^T$ and $G = \left[\begin{array}{cc} G_{h,h} & G_{\ell,h} \\ G_{\ell,h}^T & G_{\ell,\ell} \end{array}\right]$:

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

where $D(\boldsymbol{\alpha}_\ell)$ is a $(|I_\ell| \times |I_\ell|)$ diagonal matrix with diagonal elements $\log\left(\frac{\boldsymbol{\alpha}_\ell}{\mathbf{1} - \boldsymbol{\alpha}_\ell}\right)$. We then get the Hessian of $\bar{L}$ with respect to $\left[\boldsymbol{\alpha}^T \gamma\right]^T$,

$$\nabla^2 \bar{L}(\boldsymbol{\alpha}, \gamma) = \begin{bmatrix} G_{h,h} & G_{\ell,h} & \mathbf{y}_h \\ G_{\ell,h}^T & G_{\ell,\ell} + D'(\boldsymbol{\alpha}_\ell) & \mathbf{y}_\ell \\ \mathbf{y}_h^T & \mathbf{y}_\ell^T & 0 \end{bmatrix} \enspace , \tag{15}$$

where $D'(\boldsymbol{\alpha}_\ell)$ is a $(|I_\ell| \times |I_\ell|)$ diagonal matrix with diagonal elements $\frac{1}{\boldsymbol{\alpha}_\ell(\mathbf{1}-\boldsymbol{\alpha}_\ell)}$. Note that, since $0 < \alpha_i < 1$ for $i \in I_\ell$, the Hessian is ensured to be positive definite provided the kernel K is positive definite. A Newton step then consists in solving :

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

that is:

$$\begin{bmatrix} \boldsymbol{\alpha}_h^{(k+1)} \\ \boldsymbol{\alpha}_\ell^{(k+1)} \\ \gamma^{(k+1)} \end{bmatrix} = \left[\nabla^2 \bar{L}(\boldsymbol{\alpha}^{(k)}, \gamma^{(k)})\right]^{-1} \begin{bmatrix} -\mathbf{F}_h \\ \boldsymbol{\delta}_\ell^{(k)} \\ 0 \end{bmatrix} , \tag{16}$$

with $\boldsymbol{\delta}_\ell^{(k)} = \left(D'(\boldsymbol{\alpha}_\ell^{(k)}) - D(\boldsymbol{\alpha}_\ell^{(k)})\right)\boldsymbol{\alpha}_\ell^{(k)}$. The Newton steps are iterated until convergence or until the partition of variables in I_0, I_h and I_ℓ has to be modified.

4.2.5 Updating the partition

Given an assumed partition, each Newton step returns an improved solution. The latter should obey the box constraints (13) to ensure consistency with the initial conjecture. If it is not the case, the solution has to be amended. This is done by backtracking on the line search, until all box constraints are satisfied. That is, one computes the largest step size ρ such that $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k)} + \rho(\boldsymbol{\alpha}^{(k+1)} - \boldsymbol{\alpha}^{(k)})$ fulfills (13). Once this is done, noting i the "faulty" component(s) of $\boldsymbol{\alpha}$, the partition is modified as follows:

- if $i \in I_h$ and $\alpha_i = 0$, i is moved to I_0;
- if $i \in I_h$ and $\alpha_i = \frac{1}{1+e^{-F_i}}$, i is moved to I_ℓ;
- if $i \in I_\ell$ and $\alpha_i = \frac{1}{1+e^{-F_i}}$, i is moved to I_h.

Starting from $\boldsymbol{\alpha}$, a Newton update is then performed with the new partition, and the process is iterated until all box constraints are satisfied.

When the fixed point of (16) is reached, and no box constraint in the active set is violated, we may then proceed to the next candidate variable in the inactive set I_0. All $i \in I_0$ such that $-y_i(f(\mathbf{x}_i) + b) > F_i$ are candidate. We simply pick the one which maximizes $-y_i(f(\mathbf{x}_i) + b) - F_i$. Incorporating one variable at a time allows to perform efficient rank-one updates of the Cholesky decomposition of the Hessian matrix (15). Once there is no more candidate variable in I_0, the algorithm terminated at the optimal solution.

Testing for new active variables requires to know b, which may be computed from the examples in the set I_h, for which $-y_i(f(\mathbf{x}_i) + b) = F_i$. Also, as $f(\mathbf{x}_i) = \sum_{j \in I_{\bar{0}}} \alpha_j G_{ij}$, by identification with eq. (14), we see that $b = \gamma$.

5 Experiments

For experimenting with unbalanced two-class problems, we used the Forest database, the largest available UCI dataset. [1] There are 54 features, 10 of which are quantitative and the remaining 44 are binary. Originally, there are 7 classes, but we consider the subproblem of discriminating the positive class Krummholz (20 510 examples) against the negative class Spruce/Fir (211 840 examples). The ratio of positive to negative examples is 8.8%, and the classes are relatively well separated. As no cost matrix is provided with the data set, we arbitrarily chose the costs for false negative and false positive, C^+ and C^-, in order to encourage equal error rates in the two categories, that is $\frac{C^-}{C^+ + C^-} = \pi^+$, where $\pi^+ = .088$ is the proportion of positive examples. The losses are then defined up to an irrelevant factor, and we picked $C^- = \pi^+$ and $C^+ = 1 - \pi^+$.

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

Table 2: Mean test loss for sparse logistic regression and logistic regression ($p_{\max} - p_{\min} = 100\%$)

$p_{\min}$ (%)	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
Mean test loss ($\times 10^{-2}$)	1.86 ± 0.01	1.86 ± 0.01	1.85 ± 0.01	1.85 ± 0.01	1.83 ± 0.02	1.78 ± 0.02
Mean decision threshold (%)	9.5 ± 1.3	9.0 ± 1.1	9.0 ± 0.9	9.0 ± 0.6	8.8 ± 0.2	8.8 ± 0.0
Mean prop. of SVs (%)	100	65.5	53.5	40.5	34.0	27.9

5.1 Experimental setup

To ensure the representativity of results, the data is partitioned in 10 subsets. Each subset is iteratively used as a training set while the remaining ones are used as test sets. The training sets comprise thus 23 235 examples. The proportion of examples in the positive (minority) class is identical in all subsets. The features are normalized (centered and standardized) before each training session.

The experiments reported here were performed with linear classifiers. We optimized the penalization parameter λ for logistic regression (6) and sparse logistic regression (8) by 5-fold cross-validation. We jointly optimized the decision threshold, a procedure that is often applied to classifiers in order to correct for the bias in estimated probabilities. This bias correction was expected to favor logistic regression.

The range of conditional probabilities $[p_{\min}, p_{\max}]$, which is supposed to be user-defined, is not optimized. Better classification results are expected for the smallest intervals, but the range of faithful conditional probabilities then becomes tiny. We report results for various lengths of the interval centered on π^+ on a log scale, that is with $\sqrt{p_{\min} p_{\max}} = \pi^+$.

5.2 Results

We report the mean results (with standard deviations) of sparse logistic regression in table 2. 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). We also display the average decision threshold estimated by cross-validation. It is slightly above $\pi^+ = 8.8\%$ for the standard logistic regression, but the difference may not be significant (usual hypothesis tests can not be applied since the experiments are dependent). The correct decision threshold is always chosen for sparse logistic regression with small $[p_{\min}, p_{\max}]$ intervals. The classifiers are thus well calibrated regarding decision.

Figure 1 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. The figures obtained for the other trials are similar; namely, logistic regression has a wide flat minimum, reflecting that the ratio of correct classification does not change much in the neighborhood of the decision boundary. This reveals that the true conditional probabilities fluctuate non-monotonically in this region. Sparse logistic regression behaves much better, with a lower, narrower minimum centered at π^+, reflecting well-calibrated conditional probabilities in the targeted region.

Table 3 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. This can be arranged by shifting the decision boundary. The corresponding performance is shown under the column "with bias correction". However, a best choice is to modify the hinge loss to accommodate for $C^+ \neq C^-$ (Osuna et al., 1997). The corresponding result, displayed under the column C^+/C^-, reaches performances and percentage of SVs similar to the one of sparse logistic regression with small $[p_{\min}, p_{\max}]$ intervals.

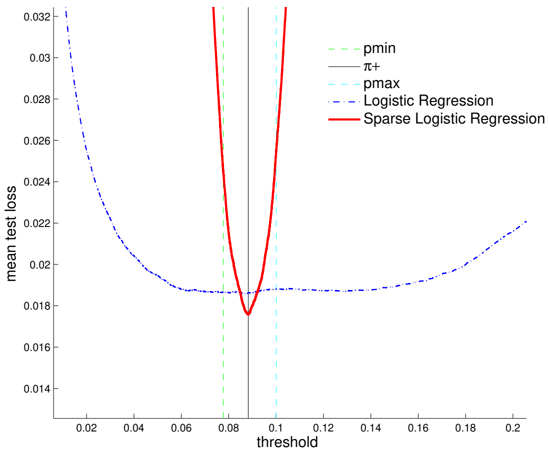

Figure 1: Mean test loss versus decision threshold.

Table 3: Mean test loss obtained for SVMs

SVM	Standard	With bias correction	C^+/C^-
Mean Test Loss ($\times 10^{-2}$)	3.75 ± 0.23	2.31 ± 0.12	1.79 ± 0.02
Mean prop. of SVs (%)	12.84 ± 0.79	13.16 ± 1.13	26.19 ± 0.60

6 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.

Further experiments are in progress to evaluate the practical interest of sparse probabilistic classifiers, but they are a promising way to address the class imbalance problem. Instead of applying different weights to the different categories, like SVMs trained with asymmetrical costs C^+/C^- do, the training criterion tends to select less active examples in the majority class, and only the ambiguous ones. It thus performs a virtual targeted undersampling of the majority class. To our knowledge, there is no other algorithm implementing this principle.

Our preliminary 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.

References

Bartlett, P. L., & Tewari, A. (2004). Sparseness vs estimating conditional probabilities: Some asymptotic results. *Proceedings of the 17th Annual Conference On Learning Theory* (pp. 564–578). Springer.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Chakrabartty, S., & Cauwenberghs, G. (to appear). Gini-support vector machine: Quadratic entropy based multi-class probability regression. *JMLR*.

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

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

Collobert, R., Sinz, F., Weston, J., & Bottou, L. (2006). Trading convexity for scalability. *ICML '06: Proceedings of the 23rd international conference on Machine learning* (pp. 201–208). ACM Press.

Cox, T. F., & Pearce, K. F. (1997). A robust logistic discrimination model. *Statistics and Computing*, *7*, 155–161.

Grandvalet, Y., Mariéthoz, J., & Bengio, S. (2006). A probabilistic interpretation of SVMs with an application to unbalanced classification. *Advances in Neural Information Processing Systems 18* (pp. 467–474). MIT Press.

Keerthi, S. S., Duan, K. B., Shevade, S. K., & Poo, A. N. (2005). A fast dual algorithm for kernel logistic regression. *Mach. Learn.*, *61*, 151–165.

Loosli, G., & Canu, S. (to appear). Comments on the "core vector machines: Fast SVM training on very large data sets". *JMLR*.

Osuna, E., Freund, R., & Girosi, F. (1997). *Support vector machines: Training and applications* (Technical Report A.I. Memo No. 1602). M.I.T. AI Laboratory.

Platt, J. C. (2000). Probabilities for SV machines. *Advances in Large Margin Classifiers* (pp. 61–74). MIT Press.

Roth, V. (2001). Probabilistic discriminative kernel classifiers for multi-class problems. *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition* (pp. 246–253). London, UK: Springer-Verlag.

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

Ting, K. M. (2000). A comparative study of cost-sensitive boosting algorithms. *Proc. 17th International Conf. on Machine Learning* (pp. 983–990). Morgan Kaufmann.

Vishwanathan, S. V. N., Smola, A. J., & Murty, M. N. (2003). SimpleSVM. *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 760–767).

Wu, M., Schölkopf, B., & Bakir, G. H. (2006). A direct method for building sparse kernel learning algorithms. *JMLR*, *7*, 603–624.

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