Séparateurs à Vaste Marge Optimisant la Fonction F_β

Jérôme Callut and Pierre Dupont

Department of Computing Science and Engineering, INGI
Université catholique de Louvain,
Place Sainte-Barbe 2
B-1348 Louvain-la-Neuve, Belgium
{jcal,pdupont}@info.ucl.ac.be

Abstract : Dans cet article, nous introduisons une nouvelle paramétrisation des Séparateurs à Vaste Marge (SVM) appelée F_β SVM. Cette dernière permet d'effectuer un apprentissage basé sur l'optimisation de la fonction F_β au lieu de l'erreur de classification habituelle. Les expériences montrent les avantages d'une telle démarche par rapport à la formulation *soft-margin* standard (avec les écarts à la marge au carré) lorsque l'on accorde une importance différente à la précision et au rappel. Une procédure automatique basée sur le score F_β de généralisation est ensuite introduite pour sélectionner les paramètres du modèle. Cette procédure repose sur les résultats de Chapelle, Vapnik et al. (Chapelle *et al.*, 2002) concernant l'utilisation de méthodes basées sur le gradient dans le cadre de la sélection de modèles. Les dérivées de la fonction de perte F_β par rapport à la constante de régularisation C et à la largeur σ d'un noyau gaussien sont définies formellement. A partir de là, les paramètres du modèle sont sélectionnés en effectuant une descente de gradient de la fonction de perte F_β dans l'espace des paramètres. Les expériences sur des données réelles montrent les bénéfices de cette approche lorsque l'on cherche à optimiser le critère F_β.

1 Introduction

Support Vector Machines (SVM) introduced by Vapnik (Vapnik, 1995) have been widely used in the field of pattern recognition for the last decade. The popularity of the method relies on its strong theoretical foundations as well as on its practical results. Performance of classifiers is usually assessed by means of classification error rate or by Information Retrieval (IR) measures such as precision, recall, F_β, breakeven-point and ROC curves. Unfortunately, there is no direct connection between these IR criteria and the SVM hyperparameters: the regularization constant C and the kernel parameters. In this paper, we propose a novel method allowing the user to specify his requirement in terms of the F_β criterion. First of all, the F_β measure is reviewed as a user specification criterion in section 2. A new SVM parametrization dealing with the β parameter is introduced in section 3. Afterwards, a procedure for automatic model selection according

7^{e} *Conférence francophone sur l'apprentissage automatique*, CAp 2005, Presses Universitaires de Grenoble, Nice, pp. 79-91, 2005.

to F_β is proposed in section 4. This procedure is a gradient-based technique derived from the results of Chapelle, Vapnik et al. (Chapelle *et al.*, 2002). Finally, experiments with artifical and real-life data are presented in section 5.

2 User specifications with the F_β criterion

Precision and recall are popular measures to assess classifiers performance in an information retrieval context (Sebastiani, 2002). Therefore, it would be convenient to use these evaluation criteria when formulating the user specifications. For instance, let us consider the design of a classifier used to retrieve documents according to topic. Some users prefer to receive a limited list of relevant documents even if this means losing some interesting ones. Others would not want to miss any relevant document at the cost of also receiving non-relevant ones. Those specifications correspond respectively to a high precision and a high recall.

The two previous measures can be combined in a unique F_β measure in which the paramater β specifies the relative importance of recall with respect to precision. Setting β equals to 0 would only consider precision whereas taking $\beta = \infty$ would only take recall into account. Moreover, precision and recall are of equal importance when using the F_1 measure. The contingency matrix and estimations of precision, recall and F_β are given hereafter.

	Target: +1	**Target: -1**
+1	True Pos. ($\#TP$)	False Pos. ($\#FP$)
-1	False Neg. ($\#FN$)	True Neg. ($\#TN$)

Precision π	$\frac{\#TP}{\#TP+\#FP}$
Recall ρ	$\frac{\#TP}{\#TP+\#FN}$
F_β	$\frac{(\beta^2+1)\pi\rho}{\beta^2\pi+\rho}$

3 F_β Support Vector Machines

In this section, we introduce a new parametrization of SVM allowing to formulate user specifications in terms of the F_β criterion. To do so, we establish a relation between the contingency matrix and the slack variables used in the soft-margin SVM setting. Based on this link, we devise a new optimization problem which maximizes an approximation of the F_β criterion regularized by the size of the margin.

3.1 Link between the contingency matrix and the slacks

Let us consider a binary classification task with a training set $Tr = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)\}$ where $\boldsymbol{x_i}$ is an instance in some input space $\mathcal{X}$ and $y_i \in \{-1, +1\}$ represents its category. Let n^+ and n^- denote respectively the number of positive and negative examples. The soft-margin formulation of SVM allows examples to be missclassified or to lie inside the margin by the introduction of slack variables ξ in the problem constraints:

OP1 Minimize $W(\boldsymbol{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C.\Phi(\boldsymbol{\xi})$

$$\text{s.t.} \quad \begin{cases} y_i(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) \geq 1 - \xi_i & \forall\, i = 1..n \\ \xi_i \geq 0 & \forall\, i = 1..n \end{cases}$$

where $\boldsymbol{w}$ and b are the parameters of the hyperplane.

The $\Phi(.)$ term introduced in the objective function is used to penalize solutions presenting many training errors. For any feasible solution $(\boldsymbol{w}, b, \boldsymbol{\xi})$, missclassified training examples have an associated slack value of at least 1. The situation is illustrated in figure 1. Hence, it seems natural to chose a function counting the number of slacks greater or equal to 1 as penalization function $\Phi(.)$. Unfortunately, the optimization of such a function combined with the margin criterion turns out to be a mixed-integer problem known to be NP-hard (Schölkopf & Smola, 2002). In fact, two approximations of the counting function are commonly used: $\Phi(\boldsymbol{\xi}) = \sum_{i=1}^{n} \xi_i$ (1-norm) and $\Phi(\boldsymbol{\xi}) = \sum_{i=1}^{n} \xi_i^2$ (2-norm). These approximations present two peculiarities: 1) The sum of slacks related to examples inside the margin might be considered as errors. 2) Examples with a slack value greater than 1 might contribute as more than one error. However, the use of these approximations is computationally attractive as the problem remains convex, quadratic and consequently solvable in polynomial time. In the sequel, we will focus on the 2-norm alternative.

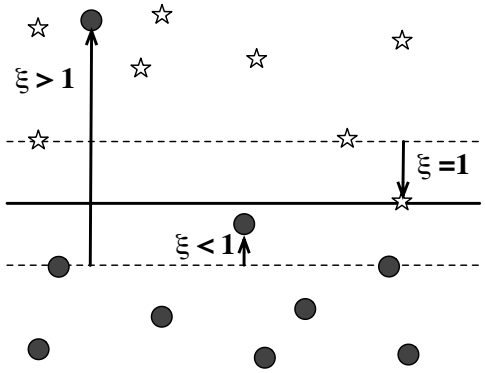

Figure 1: Soft-margin SVM and associated slacks

The computation of the preceding approximations separately for different class labels allows to bound the elements of the contingency matrix.

Proposition 1
Let ($\boldsymbol{w}$,b,$\boldsymbol{\xi}$) be a solution satisfying the constraints of OP1. The following bounds holds for the elements of the contingency matrix computed on the training set:

$$\bullet\ \#TP \geq n^+ - \sum_{\{i|y_i=+1\}} \xi_i^2 \qquad \bullet\ \#FP \leq \sum_{\{i|y_i=-1\}} \xi_i^2$$
$$\bullet\ \#FN \leq \sum_{\{i|y_i=+1\}} \xi_i^2 \qquad \bullet\ \#TN \geq n^- - \sum_{\{i|y_i=-1\}} \xi_i^2$$

These bounds will be called the slack estimates of the contingency matrix. It should be noted that they also could have been formulated using the 1-norm approximation.

3.2 The F_β parametrization

Let us introduce a parametrization of SVM in which a regularized F_β criterion is optimized. The F_β function can be expanded using the definition of precision and recall as:

$$F_\beta = \frac{(\beta^2+1)\pi\rho}{\beta^2\pi+\rho} = \frac{(\beta^2+1)\#TP}{(\beta^2+1)\#TP + \beta^2\#FN + \#FP}$$

The optimal value for F_β (≤ 1) is obtained by minimizing $\beta^2\#FN+\#FP$. Replacing $\#FN$ and $\#FP$ by their slack estimates and integrating this into the objective function leads to the following optimization problem:

OP2 Minimize $\quad W(\boldsymbol{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C.[\beta^2. \sum_{\{i|y_i=+1\}} \xi_i^2 + \sum_{\{i|y_i=-1\}} \xi_i^2]$

s.t. $\quad \begin{cases} y_i(\langle \boldsymbol{w}, \boldsymbol{x_i}\rangle + b) \geq 1 - \xi_i & \forall\ i = 1..n \\ \xi_i \geq 0 & \forall\ i = 1..n \end{cases}$

The relative importance of the F_β criterion with respect to the margin can be tuned using the regularization constant C. Since the slack estimates for $\#FP$ and $\#FN$ are upper bounds, OP2 is based on a pessimistic estimation of the F_β. OP2 can be seen as an instance of the SVM parametrization considering two kinds of slacks with the associated regularization constants C^+ and C^- (Nello Critianini, 2002). In our case, the regularization constants derive from the β value, i.e. $C^+ = C\beta^2$ and $C^- = C$. It should be pointed out that when $\beta = 1$, OP2 is equivalent to the traditional 2-norm soft-margin SVM problem.

The optimization of the F_β criterion is closely related to the problem of training a SVM with an imbalanced dataset. When the prior of a class is by far larger than the prior of the other class, the classifier obtained by a standard SVM training is likely to act as the trivial acceptor/rejector (i.e. a classifier always predicting $+1$, respectively -1). To avoid this inconvenience, some authors (Veropoulos *et al.*, 1999) have introduced different penalities for the different classes using C^+ and C^-. This method has been applied in order to control the sensitivity[1] of the model. However, no automatic procedure has been proposed to choose the regularization constants with respect to the

[1]The sensitivity is the rate of true positive examples and is equivalent to recall.

user specifications. Recently, this technique has been improved by artificially oversampling the minority class (Akbani *et al.*, 2004). Other authors (Amerio *et al.*, 2004) have proposed to select a unique regularization constant C through a bootstrap procedure. This constant is then used as a starting point for tuning C^+ and C^- on a validation set.

4 Model selection according to F_β

In the preceding section, we proposed a parametrization of SVM enabling the user to formulate his specifications with the β parameter. In addition, the remaining hyperparameters, i.e. the regularization constant and the kernels parameters, must be selected. In the case of SVM, model selection can be made using the statistical properties of the optimal hyperplane, thus avoiding the need of performing cross-validation. Indeed, several bounds of the leave-one-out (loo) error rate can be directly derived from the parameters of the optimal hyperplane expressed in dual form (Vapnik & Chapelle, 2000; Schölkopf *et al.*, 1999; Joachims, 2000). A practical evaluation of several of these bounds has been recently proposed in (Duan *et al.*, 2003). Moreover, Chapelle, Vapnik et al. (Chapelle *et al.*, 2002) have shown that the hyperplane dual parameters are differentiable with respect to the hyperparameters. This allows the use of gradient-based techniques for model selection (Chapelle *et al.*, 2002; Chung *et al.*, 2003). In this section, we propose a gradient-based algorithm selecting automatically C and the width σ of a gaussian kernel[2] according to the generalization F_β score.

4.1 The generalization F_β loss function

It has been proved by Vapnik (Vapnik, 1998) that for an example $(\boldsymbol{x_i}, y_i)$ producing a loo error, $4\alpha_i R^2 \geq 1$ holds, where R is the radius of the smallest sphere enclosing all the training examples and α_i is the i-th dual parameter of the optimal hyperplane. This inequality was originally formulated for the hard-margin case. However, it can be applied to the 2-norm soft-margin SVM as the latter can be seen as a hard margin problem with a transformed kernel (Cortes & Vapnik, 1995; Nello Critianini, 2002). Using the preceding inequality, one can build an estimator of the generalization F_β score of a given model. Alternately, it is possible to formulate a loss function following the reasoning developed in section 3.2:

$$L_{F_\beta}(\boldsymbol{\alpha}, R) \triangleq 4R^2 \left(\beta^2 \sum_{\{i|y_i=+1\}} \alpha_i + \sum_{\{i|y_i=-1\}} \alpha_i \right)$$

4.2 The model selection algorithm

We introduce here an algorithm performing automatic model selection according to the F_β criterion. It selects the model by performing a gradient descent of the F_β loss

[2] $k(\boldsymbol{x_i}, \boldsymbol{x_j}) = exp(-\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2 / 2\sigma^2)$

function over the set of hyperparameters. For the sake of clarity, C and σ, are gathered in a single vector $\boldsymbol{\theta}$. The model selection algorithm is sketched hereafter.

Algorithm F_β MODELSELECTION
Input: Training set $Tr = (\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)$
Initial values for the hyperparameters $\boldsymbol{\theta^0}$
Precision parameter ϵ
Output: Optimal hyperparameters $\boldsymbol{\theta^*}$
SVM optimal solution $\boldsymbol{\alpha^*}$ using $\boldsymbol{\theta^*}$

$\boldsymbol{\alpha}^0 \leftarrow \texttt{trainF}_\beta\texttt{SVM}(Tr, \boldsymbol{\theta^0})$;
$(R, \boldsymbol{\lambda})^0 \leftarrow \texttt{smallestSphereRadius}(Tr, \boldsymbol{\theta^0})$;

repeat

$\boldsymbol{\theta}^{t+1} \leftarrow \texttt{updateHyperparameters}(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^t, R^t, \boldsymbol{\lambda}^t)$;
$\boldsymbol{\alpha}^{t+1} \leftarrow \texttt{trainF}_\beta\texttt{SVM}(Tr, \boldsymbol{\theta^{t+1}})$;
$(R, \boldsymbol{\lambda})^{t+1} \leftarrow \texttt{smallestSphereRadius}(Tr, \boldsymbol{\theta^{t+1}})$;
$t \leftarrow t + 1$;

until $|L_{F_\beta}(\boldsymbol{\alpha}^t, R^t) - L_{F_\beta}(\boldsymbol{\alpha}^{t-1}, R^{t-1})| < \epsilon$;
return $\{\boldsymbol{\theta^t}, \boldsymbol{\alpha^t}\}$

The $\texttt{trainF}_\beta\texttt{SVM}$ function solves OP3, the dual problem of OP2, which has the same form as the dual hard-margin problem (Schölkopf & Smola, 2002):

OP3 Maximize $W(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k'(\boldsymbol{x_i}, \boldsymbol{x_j}) + \sum_{i=1}^{n} \alpha_i$

s.t. $\begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i \geq 0 \quad \forall\, i = 1..n \end{cases}$

with a transformed kernel:

$$k'(\boldsymbol{x_i}, \boldsymbol{x_j}) = \begin{cases} k(\boldsymbol{x_i}, \boldsymbol{x_j}) + \delta_{ij} \cdot \frac{1}{C\beta^2} & \text{if } y_i = +1 \\ k(\boldsymbol{x_i}, \boldsymbol{x_j}) + \delta_{ij} \cdot \frac{1}{C} & \text{if } y_i = -1 \end{cases}$$

where δ_{ij} is the Kronecker delta and $k(.,.)$ is the original kernel function.

The radius of the smallest sphere enclosing all the examples computed by the `smallestSphereRadius` function is obtained by taking the square root of the objective function optimal value in the following optimization problem (Schölkopf & Smola, 2002):

OP4 Maximize $W(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i k'(\boldsymbol{x_i}, \boldsymbol{x_i}) - \sum_{i,j=1}^{n} \lambda_i \lambda_j k'(\boldsymbol{x_i}, \boldsymbol{x_j})$

s.t. $\begin{cases} \sum_{i=1}^{n} \lambda_i = 1 \\ \lambda_i \geq 0 \quad \forall\, i = 1..n \end{cases}$

The optimization problems OP3 and OP4 can be solved in polynomial time in n, *e.g.* using an interior point method (Vanderbei, 1994). Furthermore, the solution to OP3, respectively OP4, at a given iteration can be used as a good starting point for the next iteration.

At each iteration, the hyperparameters can be updated by means of a gradient step : $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta . \partial L_{F_\beta} / \partial \boldsymbol{\theta}$ where $\eta > 0$ is the updating rate. However, second order methods often provide a faster convergence, which is valuable since two optimization problems have to be solved at each iteration. For this reason, the `updateHyperparameters` function relies on the BFGS algorithm (Fletcher & Powell, 1963), a quasi-Newton optimization technique. The time complexity of the `updateHyperparameters` function is $\mathcal{O}(n^3)$ since it is dominated by the inversion of a possibly $n \times n$ matrix (see section 4.3). The derivatives of the F_β loss function with respect to the hyperparameters are detailed in the next section. The algorithm is iterated until the F_β loss function no longer changes by more than ϵ.

4.3 Derivatives of the F_β loss function

The derivatives of the transformed kernel function with respect to the hyperparameters are given by:

$$\frac{\partial k'(\boldsymbol{x_i}, \boldsymbol{x_j})}{\partial C} = \begin{cases} -1/(C^2\beta^2) & \text{if } i = j \text{ and } y_i = +1 \\ -1/C^2 & \text{if } i = j \text{ and } y_i = -1 \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial k'(\boldsymbol{x_i}, \boldsymbol{x_j})}{\partial \sigma^2} = k(\boldsymbol{x_i}, \boldsymbol{x_j}) \frac{\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2}{2\sigma^4}$$

The derivatives of the squared radius can then be obtained applying the lemma 2 of Chapelle, Vapnik et al. (Chapelle *et al.*, 2002):

$$\frac{\partial R^2}{\partial \theta} = \sum_{i=1}^{n} \lambda_i \frac{\partial k'(\boldsymbol{x_i}, \boldsymbol{x_i})}{\partial \theta} - \sum_{i,j=1}^{n} \lambda_i \lambda_j \frac{\partial k'(\boldsymbol{x_i}, \boldsymbol{x_j})}{\partial \theta}$$

where $\theta \in \{C, \sigma^2\}$. The derivation of the hyperplane dual parameters proposed in (Chapelle *et al.*, 2002) follows:

$$\frac{\partial(\boldsymbol{\alpha}, b)}{\partial \theta} = -H^{-1} \frac{\partial H}{\partial \theta} (\boldsymbol{\alpha}, b)^T, \quad H = \begin{pmatrix} \boldsymbol{y}^T K \boldsymbol{y} & \boldsymbol{y} \\ \boldsymbol{y}^T & 0 \end{pmatrix}$$

where K is the kernel matrix and $\boldsymbol{y}$ is the vector of examples labels. The H matrix is derived by using the preceding kernel function derivatives. It should be stressed that only examples corresponding to support vectors have to be considered in the above formula. Finally, the derivative of $L_{F_\beta}(.,.)$ with respect to a hyperparameter θ is given by:

$$\begin{aligned}\frac{\partial L_{F_\beta}(\boldsymbol{\alpha}, R)}{\partial \theta} &= 4\frac{\partial R^2}{\partial \theta}\left(\beta^2 \sum_{\{i|y_i=+1\}} \alpha_i + \sum_{\{i|y_i=-1\}} \alpha_i\right) \\ &+ 4R^2\left(\beta^2 \sum_{\{i|y_i=+1\}} \frac{\partial \alpha_i}{\partial \theta} + \sum_{\{i|y_i=-1\}} \frac{\partial \alpha_i}{\partial \theta}\right)\end{aligned}$$

5 Experiments

We performed several experiments to assess the performance of the F_β parametrization and the model selection algorithm. First, the F_β parametrization was tested with positive and negative data in $\mathbb{R}^{10}$ drawn from two largely overlapping normal distributions. The priors for positive and negative classes were respectively 0.3 and 0.7. It is usually more difficult to obtain a good recall when data are unbalanced in this way. Experiments were carried out using training sets of 600 examples, a fixed test set of 1,000 examples and a linear kernel. A comparison between the F_β parametrization and the 2-norm soft-margin SVM with $C = 1$ is displayed in figure 2. For each β considered, the training data were resampled 10 times in order to produce averaged results. In this setting, our parametrization obtained better F_β scores than the standard soft-margin SVM, especially when a high recall was requested. The second part of the figure 2 presents the evolution of precision, recall and the F_β score for different β values.

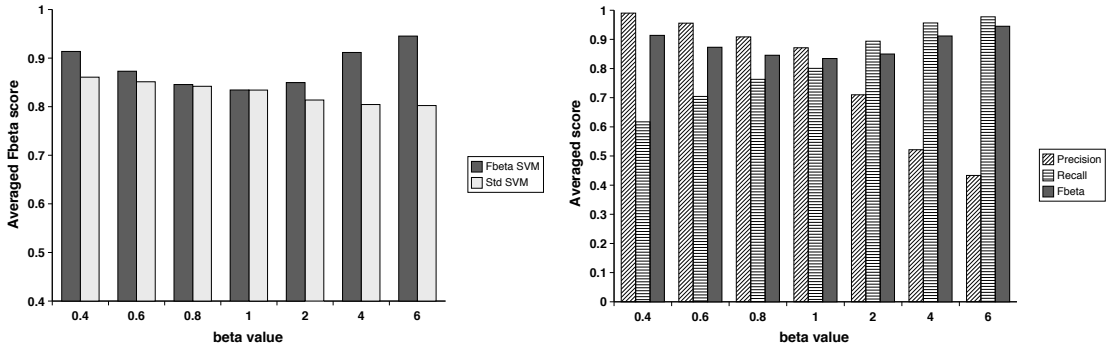

Figure 2: The F_β parametrization tested with artificially generated data. Left: comparison between the standard 2-norm soft-margin SVM and the F_β parametrization. Right: Evolution of precision, recall and of the F_β score accoring to different β values.

Afterwards, our parametrization was tested using several class priors. The experimental setup was unchanged except for the class priors while generating the training and test data. Figure 3 shows the evolution of the F_β score obtained by our parametrization and by the 2-norm soft-margin SVM using several class priors. For the standard 2-norm soft-margin SVM, one notes that the effect of the priors is particularly important when positive examples are few in numbers and that a high recall is requested. In this setting, our parametrization outperformed the standard 2-norm soft-margin SVM by more than 0.1.

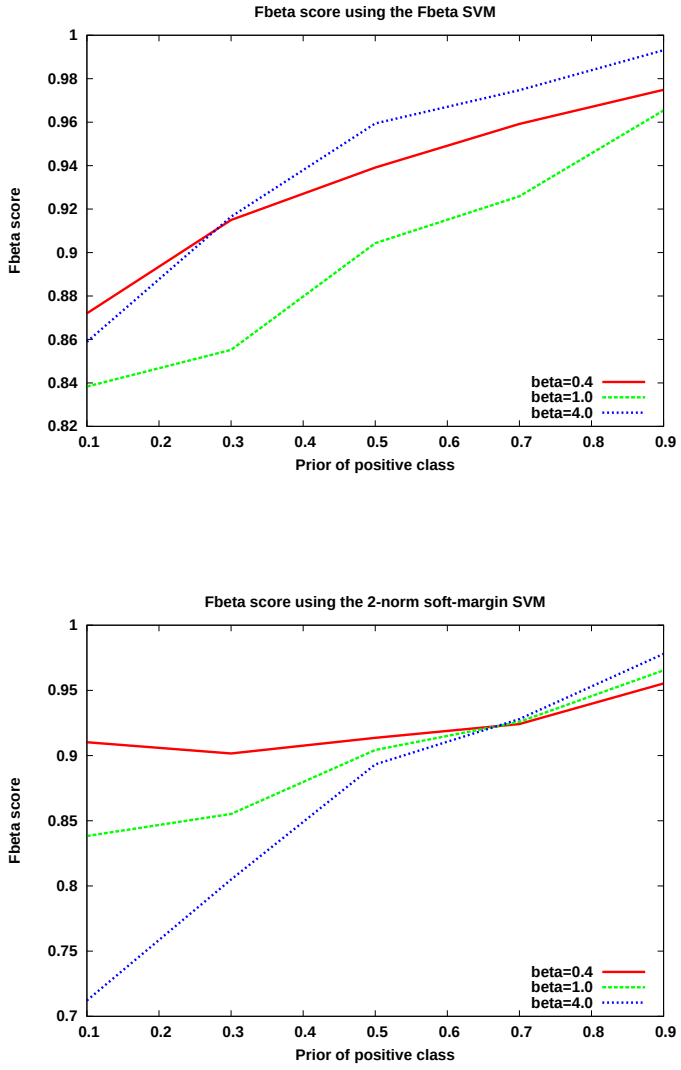

Figure 3: The F_β parametrization tested using artificially generated data with several class priors. Top: F_β scores obtained on the test set using the F_β parametrization. Bottom: F_β scores obtained on the test set using the standard 2-norm soft-margin SVM.

The model selection algorithm was first tested with data generated as in the previous paragraph. The hyperparameters C and σ were initialized to 1 and the precision parameter ϵ was set to 10^{-6}. Our objective was to investigate the relation between the minimization of the F_β loss function and the F_β score obtained on unknown test data. The figure 4 shows the evolution of the F_β loss function during the gradient descent, using $\beta = 2$. The associated precision, recall and F_β scores on test data are displayed in the bottom of the figure 4. Even if the optima of the F_β loss function and the F_β score do not match exactly, one can observe that good F_β scores were obtained when the F_β loss function is low. After 35 iterations, the classifier obtained a F_β score close to 0.9 with the hyperparameters $C = 4.33$ and $\sigma = 1.94$.

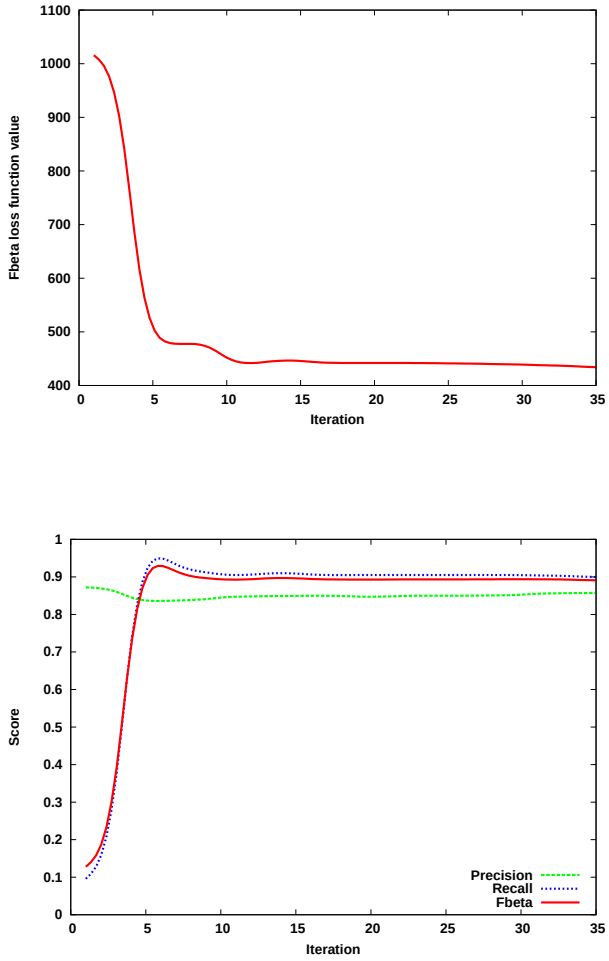

Figure 4: The F_β model selection algorithm tested with artificially generated data and with $\beta = 2$. Top: the evolution of the F_β loss function during the gradient descent. Bottom: the related values of precision, recall and F_β score on independent test data.

The model selection algorithm was then compared to the Radius-Margin (RM) based algorithm (Chapelle *et al.*, 2002) using the `Diabetes` dataset (Blake & Merz, 1998).

This dataset contains 500 positive examples and 268 negative examples. It was randomly split into a training and a test set, each one containing 384 examples. In this setting, it is usually more difficult to obtain a classifier with a high precision. The same initial conditions as before were used. The RM based algorithm select the model parameters of the 2-norm soft-margin SVM according to the RM estimator of the generalization error rate. It should be pointed out that when $\beta = 1$, both methods are equivalent since the same function is optimized. The comparison is illustrated in the first part of the figure 5. As expected, our method provided better results when β moves far away from value 1. The influence of the β parameter on precision, recall and the F_β score can be observed in the second part of the figure 5.

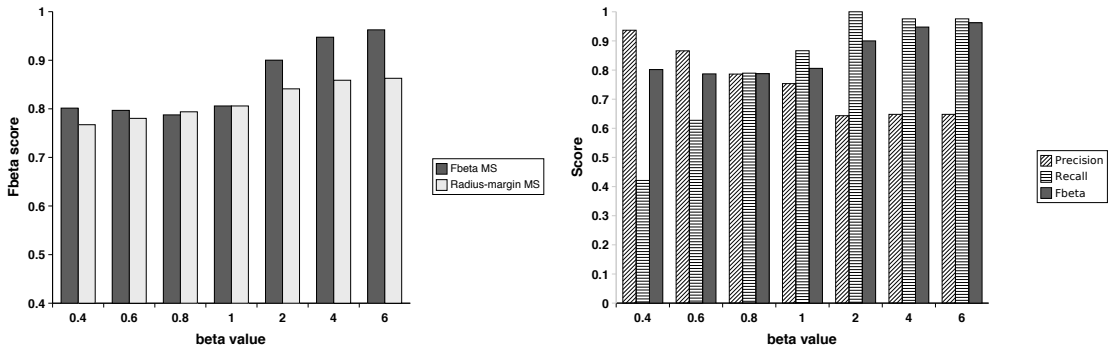

Figure 5: The F_β model selection algorithm tested with the `Diabetes` dataset. Left: Comparison between the F_β model selection algorithm and the radius-margin based method. Right: Evolution of precision, recall and of the F_β score accoring to different β values.

6 Conclusion

We introduced in this paper F_β SVMs, a new parametrization of support vector machines. It allows to formulate user specifications in terms of F_β, a classical IR measure. Experiments illustrates the benefits of this approach over a standard SVM when precision and recall are of unequal importance. Besides, we extended the results of Chapelle, Vapnik et al. (Chapelle *et al.*, 2002) based on the Radius-Margin (RM) bound in order to automatically select the model hyperparameters according to the generalization F_β score. We proposed an algorithm which performs a gradient descent of the F_β loss function over the set of hyperparameters. To do so, the partial derivatives of the F_β loss function with respect to these hyperparameters have been formally defined. Our experiments on real-life data show the advantages of this method compared to the RM based algorithm when the F_β evaluation criterion is considered.

Our future work includes improvements to the model selection algorithm in order to deal with larger training sets. Indeed, it is possible to use a sequential optimization method (Keerthi *et al.*, 2000) in the `smallestSphereRadius` function and chunking techniques (Joachims, 1998; Schölkopf & Smola, 2002) in the `trainF`$_\beta$`SVM` function.

This typically allows to solve problems with more than 10^4 variables. Moreover, we believe that the inverse matrix H^{-1} can be computed incrementally during the chuncking iterations, using the Schur inversion formula for block matrices (Meyer, 2000).

Acknowledgment

This work is partially supported by the *Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture (F.R.I.A.)* under grant reference F3/5/5-MCF/FC-19271.

References

AKBANI R., KWEK S. & JAPKOWICZ N. (2004). Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, p. 39–50, Pisa, Italy.

AMERIO S., ANGUITA D., LAZZIZZERA I., RIDELLA S., RIVIECCIO F. & ZUNINO R. (2004). Model selection in top quark tagging with a support vector classifier. In *Proceedings of International Joint Conference on Neural Networks (IJCNN 2004)*, Budapest, Hungary.

BLAKE C. & MERZ C. (1998). UCI repository of machine learning databases.

CHAPELLE O., VAPNIK V., BOUSQUET O. & MUKHERJEE S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, **46**(1-3), 131–159.

CHUNG K.-M., KAO W.-C., SUN C.-L., WANG L.-L. & LIN C.-J. (2003). Radius margin bounds for support vector machines with the rbf kernel. *Neural Comput.*, **15**(11), 2643–2681.

CORTES C. & VAPNIK V. (1995). Support-vector networks. *Machine Learning*, **20**(3), 273–297.

DUAN K., KEERTHI S. S. & POO A. N. (2003). Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, **51**, 41–59.

FLETCHER R. & POWELL M. J. D. (1963). A rapidly convergent descent method for minimization. *Computer Journal*, **6**, 163–168.

JOACHIMS T. (1998). Making large-scale support vector machine learning practical. In A. S. B. SCHOLKOPF, C. BURGES, Ed., *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

JOACHIMS T. (2000). Estimating the generalization performance of a SVM efficiently. In P. LANGLEY, Ed., *Proceedings of ICML-00, 17th International Conference on Machine Learning*, p. 431–438, Stanford, US: Morgan Kaufmann Publishers, San Francisco, US.

KEERTHI S. S., SHEVADE S. K., BHATTACHARYYA C. & MURTHY K. R. K. (2000). A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE-NN*, **11**(1), 124.

MEYER C. D. (2000). *Matrix analysis and applied linear algebra*. Society for Industrial and Applied Mathematics.

NELLO CRITIANINI J. S.-T. (2002). *An Introduction to Support Vector Machines*. The Press Syndicate of the University of Cambridge.

SCHÖLKOPF B., SHAWE-TAYLOR J., SMOLA A. J. & WILLIAMSON R. C. (1999). *Generalization Bounds via Eigenvalues of the Gram Matrix*. Rapport interne, Australian National University. Submitted to COLT99.

SCHÖLKOPF B. & SMOLA A. (2002). *Learning with Kernels*. Cambridge: MIT Press.

SEBASTIANI F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**(1), 1–47.

VANDERBEI R. (1994). *LOQO: An Interior Point Code for Quadratic Programming*. Rapport interne SOR 94-15, Princeton University.

VAPNIK V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer Verlag.

VAPNIK V. (1998). *Statistical Learning Theory*. New York: Wiley-Interscience.

VAPNIK V. & CHAPELLE O. (2000). Bounds on error expectation for support vector machines. *Neural Computation*, **12**(9).

VEROPOULOS K., CRISTIANINI N. & CAMPBELL C. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI99)*, Stockholm, Sweden.