A generalized quadratic loss for Support Vector Machines

Filippo Portera and **Alessandro Sperduti** [1]

Abstract. The standard SVM formulation for binary classification is based on the Hinge loss function, where errors are considered not correlated. Due to this, local information in the feature space which can be useful to improve the prediction model is disregarded. In this paper we address this problem by defining a generalized quadratic loss where the co-occurrence of errors is weighted according to a kernel similarity measure in the feature space. In particular the proposed approach weights pairs of errors according to the distribution of the related patterns in the feature space. The generalized quadratic loss includes also target information in order to penalize errors on pairs of patterns that are similar and of the same class. We show that the resulting dual problem can be expressed as a hard margin SVM in a different feature space when the co-occurrence error matrix is invertible. We compare our approach against a standard SVM on some binary classification tasks. Experimental results obtained for different instances of the co-occurrence error matrix on these problems, seems to show an improvement in the performance.

1 Introduction

At present Support Vector Machines ([8]) constitute the most effective method in classification and regression tasks involving numerical features. A support vector machine jointly minimizes a combination of a margin based loss function and a regularization term measuring the size of the free parameters.

When considering binary classification problems, the right choice for the loss function would be the 0/1 loss, i.e. the loss which returns the number of classification errors. This loss, however, is difficult to use, since it has a point of discontinuity and elsewhere has null gradient, and so it results to be very difficult to minimize. For this reason, a typical choice for the loss function is the Hinge loss, which is linear for negative margins and constitutes an upper bound to the 0/1 loss. Other losses can be used according to the requirement of the problem at hand. For example, for regression tasks, a quadratic loss is usually preferred. In other cases, other loss functions are more suited, such as the exponential loss or the logistic loss.

The usual approach is to fix a loss and then to use the associated machine to perform the learning. The learning phase usually involves a search in the hyperparameters space (eventually involving a validation set) of the machine so to optimize the final performance ([6]).

When considering an application domain where there is no a priori knowledge about which is the right loss function to use, a typical approach is to independently try different losses, eventually selecting the one returning the best empirical performance.

In this paper, in the context of SVM for binary classification, we explore an approach which tries to address two issues: *i)* try to take into account correlation between patterns, *ii)* try to define a family of loss functions which takes into account the first issue.

The first issue is addressed by defining a quadratic loss on the slack variables ξ_i where the cross terms $\xi_i\xi_j$ are weighted according to the similarity between the corresponding input patterns $\bar{x}_i$ and $\bar{x}_j$, i.e. we consider the cross term $S_{ij}\xi_i\xi_j$, where S_{ij} is the similarity between patterns $\bar{x}_i$ and $\bar{x}_j$. The similarity value S_{ij} is defined on the basis of the kernel used in the SVM, so to use a metric related to the feature space metric. We also study a generalization of this approach where the joint target information for $\bar{x}_i$ and $\bar{x}_j$ is exploited to bias the learning towards solutions where the local concentration of errors on patterns of the same class is discouraged, i.e. we consider the cross term $S_{ij}y_iy_j\xi_i\xi_j$, where $y_i, y_j \in \{-1, 1\}$ are the targets associated to $\bar{x}_i$ and $\bar{x}_j$, respectively.

The second issue is addressed introducing signed slack variables in the objective function of the SVM model. Each ξ_i is considered positive if the associated target value is 1, negative otherwise. Therefore, considering the proposed quadratic loss and given two distinct patterns, the SVM solution will prefer the same amount of error on patterns of different classes versus patterns of the same class. In fact, errors involving patterns of different classes are less penalized than errors on patterns of the same class.

Notice that the generalized quadratic loss is convex if the similarity matrix S is positive definite.

We show that using this generalized quadratic loss function in a Support Vector Machine, the resulting dual problem can be expressed as a hard margin SVM in a new feature space which is related to the original feature space via the inverse of the similarity matrix S and the target information. Thus, in order to get a well-formed dual formulation we need to work with a similarity matrix which is invertible. In this paper, we explore the two cases $S_{ij} = K(\bar{x}_i, \bar{x}_j)$, and $S_{ij} = e^{K(\bar{x}_i, \bar{x}_j)}$, using a RBF kernel for implicitly generating the feature space. In both cases we guarantee that S is positive definite and invertible.

We compare our approach against a standard SVM on some binary classification tasks. The experimental results show an improvement in the performance.

Our approach is described in Section 2, where we develop our SVM model. In Section 3 we motivate our choices for the definition of the similarity matrix. In Section 4 we report experimental results of a comparison of the standard SVM with Hinge loss versus our approach on datasets of binary classification problems, also involving structured objects. Conclusions are drawn in Section 5.

2 SVM definition for a generalized quadratic loss

Suppose that l inputs $(\bar{x}_1, y_1), \ldots, (\bar{x}_l, y_l)$ are given, where $\bar{x}_i \in \mathbb{R}^d$ are the input patterns, and $y_i \in \{-1, 1\}$ are the related target values of our supervised binary classification problem. The SVM model

[1] Dipartimento di Matematica Pura ed Applicata, Universitá di Padova, Padova, Italy, email: {portera,sperduti}@math.unipd.it. This work has been partially supported by MIUR grant 2002093941_004. We thank the anonymous reviewers for their helpful comments.

for classification with a 2-norm loss function ([2]), that we denote QSVM, is:

$$\begin{array}{ll} & \min_{\bar{w},b,\bar{\xi}} \frac{1}{2}||\bar{w}||^2 + c\bar{\xi}'\bar{\xi} \\ \text{s.t.:} & \\ & y_i(\bar{w}\cdot\bar{x}_i + b) \geq 1 - \xi_i, \quad i = 1,\dots,l \end{array} \qquad (1)$$

where the classification error $\bar{\xi}'\bar{\xi}$ of a feasible solution $(\bar{w}, b)$ is an approximation of the true classification error $\sum\Theta(\xi_i - 1)$ where $\Theta(x) = 0$ if $x < 0$ and $\Theta(x) = 1$ if $x \geq 0$. Note that the non negativity constraints over $\bar{\xi}$ components are redundant since the optimal solution has $\xi_i \geq 0, \forall i \in [1..l]$. The solution of (1) can be expressed with:

$$h(\bar{x}) = sign(\sum_{i=1}^{l} \alpha_i^* y_i K(\bar{x}_i, \bar{x}) + b^*) \qquad (2)$$

where $\bar{\alpha}^*$ is the dual optimal solution and b^* can be derived from the KKT conditions.

Notice that in this formulation the errors are assumed to be independent, while in general this assumption is not true, especially when considering patterns that are spatially very close or more in general very similar. Thus, if we want to weight the co-occurrence of errors corresponding to similar patterns, we have to define a quadratic loss that weights also all the cross terms $\xi_i\xi_j$ according to a similarity measure for patterns $\bar{x}_i$ and $\bar{x}_j$. Let S be an $l \times l$ symmetric and positive definite similarity matrix. The SVM formulation we are interested in is:

$$\begin{array}{lll} & \min_{\bar{w},b,\bar{\xi}} \frac{1}{2}||\bar{w}||^2 + c\bar{\xi}'S\bar{\xi} & \\ \text{s.t.:} & & \\ & y_i(\bar{w}\cdot\bar{x}_i + b) \geq 1 - \xi_i, & i = 1,\dots,l \\ & \xi_i \geq 0 & i = 1,\dots,l \end{array} \qquad (3)$$

We here also consider a formulation where, in addition, we prefer not to have errors of the same type for patterns that are very similar each other. Such formulation can be obtained by further exploiting the target information in a generalized quadratic loss:

$$\begin{array}{lll} & \min_{\bar{w},b,\bar{\xi}} \frac{1}{2}||\bar{w}||^2 + c\bar{\xi}'YSY\bar{\xi} & \\ \text{s.t.:} & & \\ & y_i(\bar{w}\cdot\bar{x}_i + b) \geq 1 - \xi_i, & i = 1,\dots,l \\ & \xi_i \geq 0 & i = 1,\dots,l \end{array} \qquad (4)$$

where the Y matrix is diagonal with $Y_{p,p} = y_p$. Note that the matrix YSY is still symmetric and positive definite because Y is symmetric and if $\delta_i = \xi_i y_i$, then $\forall\bar{\delta}_i \in \mathbb{R}^d$, $\exists\, \bar{\xi}$ s.t. $\bar{\xi}'YSY\bar{\xi} = \bar{\delta}'S\bar{\delta} \geq 0$.

In Figure 1 we give a graphical exemplification about which type of error co-occurrence we prefer to penalize.

In this paper we focus on similarity matrices generated according to a kernel function:

$$S_{i,j} = K(\bar{x}_i, \bar{x}_j)$$

where $K(\bar{x}, \bar{y})$ is an inner product in some feature space.

Let X be the $l \times d$ matrix of input patterns. Given problem (4) the corresponding Lagrangian objective function is:

$$\frac{1}{2}||\bar{w}||^2 + c\bar{\xi}'YSY\bar{\xi} + \bar{\alpha}'(\bar{1} - \bar{\xi} - YX\bar{w} - b\bar{y}) - \bar{\lambda}'\bar{\xi} \qquad (5)$$

where $\bar{y}$ is the target vector and $\alpha_i \geq 0$, $i = 1,\dots,l$. The Kuhn Tucker conditions for optimality are:

$$\begin{array}{l} \frac{\partial L}{\partial \bar{w}} = \bar{w} - X'Y\bar{\alpha} = 0 \Rightarrow \bar{w} = X'Y\bar{\alpha} \\ \frac{\partial L}{\partial b} = -\bar{\alpha}'\bar{y} = 0 \Rightarrow \bar{\alpha}'\bar{y} = 0 \\ \frac{\partial L}{\partial \bar{\xi}} = 2cYSY\bar{\xi} - \bar{\alpha} - \bar{\lambda} = 0 \Rightarrow \bar{\xi} = \frac{YS^{-1}Y(\bar{\alpha}+\bar{\lambda})}{2c} \end{array} \qquad (6)$$

if S is invertible. Supposing that S^{-1} exists, substituting (6) in (5) gives:

$$D(\bar{\alpha}, \bar{\lambda}) = \bar{\alpha}'\bar{1} - \tfrac{1}{2}\bar{\alpha}'YKY\bar{\alpha} - \tfrac{1}{4c}(\bar{\alpha} + \bar{\lambda})'YS^{-1}Y(\bar{\alpha} + \bar{\lambda}) \qquad (7)$$

i.e. the dual objective function. Notice that the objective function is still convex because S^{-1} is positive definite since its eigenvalues are positive.

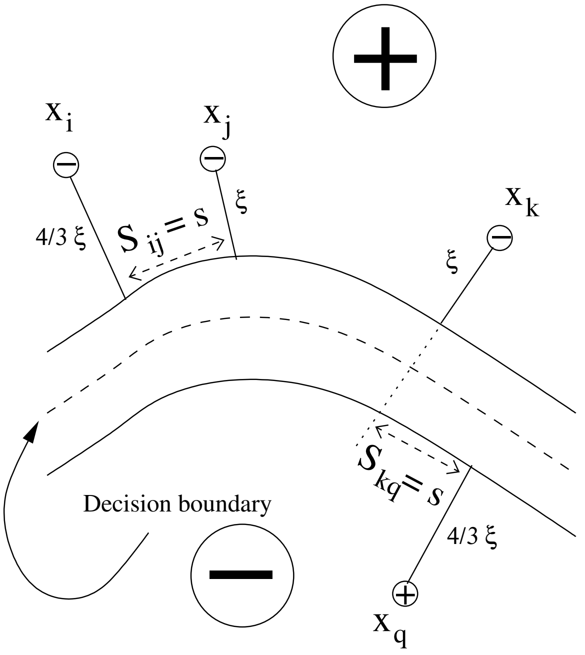

Figure 1. In our generalized quadratic loss, the error configuration generated by patterns $\bar{x}_i$ and $\bar{x}_j$ is more expensive than the error configuration generated by patterns $\bar{x}_k$ and $\bar{x}_q$. Here we assume that $S_{ij} = S_{kq} = s$.

To obtain a separating hyperplane one has to solve the following problem:

$$\begin{array}{lll} & \max_{\bar{\alpha},\bar{\lambda}} D(\bar{\alpha}, \bar{\lambda}) & \\ \text{s.t.:} & & \\ & \bar{\alpha}'\bar{y} = 0 & \\ & \alpha_i \geq 0, \lambda_i \geq 0 & i = 1,\dots,l \end{array} \qquad (8)$$

Since $(\bar{\alpha} + \bar{\lambda})'YS^{-1}Y(\bar{\alpha} + \bar{\lambda})$ is monotonically increasing and convex in the positive domain, problem (8) is equivalent to:

$$\begin{array}{lll} & \max_{\bar{\alpha}} \bar{\alpha}'\bar{1} - \frac{1}{2}\bar{\alpha}'Y(K + \frac{1}{2c}S^{-1})Y\bar{\alpha} & \\ \text{s.t.:} & & \\ & \bar{\alpha}'\bar{y} = 0 & \\ & \alpha_i \geq 0 & i = 1,\dots,l \end{array} \qquad (9)$$

Then, being that K and S^{-1} are positive definite matrices and since the sum of convex functions is still a convex function, we have that the objective function of (9) is convex. The solution is obtained using optimal coefficients of problem (9) with the first condition of (6) and the optimal value of b follows from the Kuhn-Tucker optimality conditions ([8]).

Notice that when S^{-1} exists, problem (9) is equivalent to a hard margin SVM problem with a kernel matrix equal to $K + \frac{1}{2c}S^{-1}$, while the classification function is defined over the feature space induced by kernel K.

Actually, in this case it is also possible to explicitly build a feature map. Let consider the following mapping $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d+l}$ that, for all $i \in [1, \dots, l]$, maps $\bar{x}_i \mapsto \phi(\bar{x}_i)$:

$$\phi(\bar{x}_i) = [\bar{x}_i', (\sqrt{\frac{S^{-1}}{2c}}\bar{e}_i)']'$$

where $\bar{e}_i$ is the i-th vector of the canonical base of $\mathbb{R}^l$. It is not difficult to see that the kernel matrix obtained with this transformation is equal to $K + \frac{S^{-1}}{2c}$.

In the following we denote the overall method with QLYSVM, while the corresponding method obtained for the primal problem (3) is denoted QLSVM.

Dataset (d, l), Algorithms	C	γ_K	γ_S	Error %	σ
Banana (2, 400)					
SVM$_t$	-	-	-	10.9	-
SVM	10000	0.1	-	10.9	0.54
QSVM	1	1	-	10.5	0.55
QLYSVM	1	1	100	10.6	**0.44**
QLYSVM$_N$	1	1	1000	10.6	0.47
QLYSVM$_e$	0.1	1	100	10.8	0.60
QLYSVM$_{eN}$	1	1	1	**10.4**	0.52
QLSVM	1	1	100	10.7	0.45
Breast Cancer (9, 200)					
SVM$_t$	-	-	-	26.9	-
SVM	1000	0.001	-	27.0	**3.22**
QSVM	100	0.001	-	27.4	3.26
QLYSVM	1000	0.001	0.1	26.9	3.63
QLYSVM$_N$	10000	0.001	0.106	26.9	3.34
QLYSVM$_e$	0.01	0.1	0.1	**26.6**	3.40
QLYSVM$_{eN}$	10	0.001	1	28.6	4.15
QLSVM	100	0.001	1	27.5	3.89
German (2, 700)					
SVM$_t$	-	-	-	**22.6**	-
SVM	100	0.001	-	23.4	**1.62**
QSVM	1	0.01	-	**22.6**	1.85
QLYSVM	1	0.01	10000	**22.6**	1.85
QLYSVM$_N$	1	0.01	10000	**22.6**	1.85
QLYSVM$_e$	0.1	0.01	0.001	23.0	1.77
QLYSVM$_{eN}$	0.1	0.01	10	23.4	1.77
QLSVM	1	0.01	10000	**22.6**	1.85
Image (18, 1300)					
SVM$_t$	-	-	-	3.0	-
SVM	10000	0.01	-	3.2	0.72
QSVM	1000	0.1	-	2.9	0.60
QLYSVM	1000	0.1	10000	2.9	0.60
QLYSVM$_N$	1000	0.1	10000	2.9	0.60
QLYSVM$_e$	10000	0.01	0.1	**2.7**	**0.54**
QLYSVM$_{eN}$	10000	0.01	0.01	**2.7**	**0.54**
QLSVM	10000	0.01	100	2.8	0.58
Waveform (21, 400)					
SVM$_t$	-	-	-	10.3	-
SVM	10	0.01	-	10.3	0.40
QSVM	1	0.01	-	**10.0**	**0.25**
QLYSVM	1	0.01	0.1	10.1	0.36
QLYSVM$_N$	1	0.01	0.1	**10.0**	0.36
QLYSVM$_e$	1	0.01	0.1	**10.0**	0.28
QLYSVM$_{eN}$	0.1	0.1	0.1	**10.0**	0.28
QLSVM	1	0.01	10000	**10.0**	**0.25**

Table 1. Comparison of the average generalization error and standard deviation σ, computed over the 10 splits, for five different algorithms on five UCI binary datasets. The results quoted for SVM$_t$ are taken from ([7]).

3 Definition of the similarity matrix

The dual solution of problem (4) is based on the inversion of S. When S is invertible, a method to compute S^{-1} is the Strassen method with a complexity of $\mathbb{O}(n^{log_2 7})$. An inversion algorithm with lower complexity can be found in [1].

Note that when all patterns are distinct points and S is generated with a Gaussian RBF kernel then S is invertible ([5]). Under some extreme experimental conditions, however, a similarity matrix defined in this way may be ill-conditioned and inversion can be problematic.

For this reason we also considered an exponential kernel e^K, defined with:

$$e^K = \sum_{i=0}^{+\infty} \frac{K^i}{i!}. \qquad (10)$$

A kernel matrix obtained with (10) is always invertible and its inverse is:

$$(e^K)^{-1} = e^{-K}$$

and it can be calculated using the Padé approximation method with a complexity of $\mathbb{O}(qn^{log_2 7})$ where q is the order of the approximation. Experimentally we never had problems in computing the inverse of the exponential matrix.

Since error losses in problems (1), (3), and (4) are based on different loss functions, to conduct a more fair comparison of the approaches we also considered versions of the problems (3), and (4) where the norm of S is equalized to the norm of I. If we consider the l$_2$ norm of a matrix S defined as $||S||_2 = \sqrt{\sum_{i,j} S_{i,j}^2}$, we have that $||I||_2 = \sqrt{l}$. Therefore we build a matrix $\hat{S}$ with norm $\sqrt{l}$:

$$\hat{S} = \sqrt{l}\frac{S}{||S||_2} \qquad (11)$$

that can be employed with the QLSVM and QLYSVM methods.

4 Experiments

We tested our models on distinct classification tasks. We report the results obtained with some instances extracted from the well known UCI dataset. We also conducted experiments with logic terms classification problems. To conduct the experiments we used a modified version of SVMLight 5.0 ([3]) enabled to work with a kernel matrix generated with Scilab 2.7 ©INRIA-ENPC.

4.1 Experiments on UCI dataset instances

We considered five binary problems from the benchmark of R$\ddot{a}$tsch available on the web at `http://ida.first.gmd.de/∼raetsch`. Each dataset comprises the first 10 splits of the 100 split available, where each split has a training and a test set. In Table 1 we report a comparison of SVM with Hinge loss and RBF kernel ($K_{i,j} = e^{-\gamma_K ||\bar{x}_i - \bar{x}_j||^2}$), QLYSVM with a RBF kernel used both for K and S (with hyperparameter γ_S), and QLYSVM (referred as QLYSVM$_e$) where S is the exponential matrix e^K, being K the previously defined RBF kernel matrix. The last two algorithms were evaluated also employing a normalized S matrix (QLYSVM$_N$ and QLYSVM$_{eN}$). We also report the hyperparameters used to evaluate the performance of our algorithms. For the SVM and QSVM algorithms we used a calibration procedure that involved a 8×8 grid of the powers of 10 starting from 0.001×0.001 on hyperparameters C, γ_K and we selected the hyperparameters that obtained the best performance on the first split of the dataset.

For the QLYSVM, QLYSVM$_N$, QLYSVM$_e$, QLYSVM$_{eN}$ and QLSVM algorithms we used a calibration procedure that involved a $8 \times 8 \times 8$ grid over the powers of 10 starting from $0.001 \times 0.001 \times 0.001$ on hyperparameters C, γ_K, γ_S and we selected the hyperparameters that obtained the best performance on the first split of the dataset.

It can be observed that the best results are obtained using the generalized quadratic loss, even with different approaches for generating the similarity matrix. The generalized quadratic loss defined in (4) seems to have a better performance than the one define in (3). In Figure 2 the first block of 100×100 elements of the "optimal" S matrices for the QLYSVM algorithm are reported. As it can be seen from the figure, the quadratic losses selected by the calibration procedure are in general different, thus showing the usefulness of having a family of loss functions from which the calibration process can pick the most suitable one.

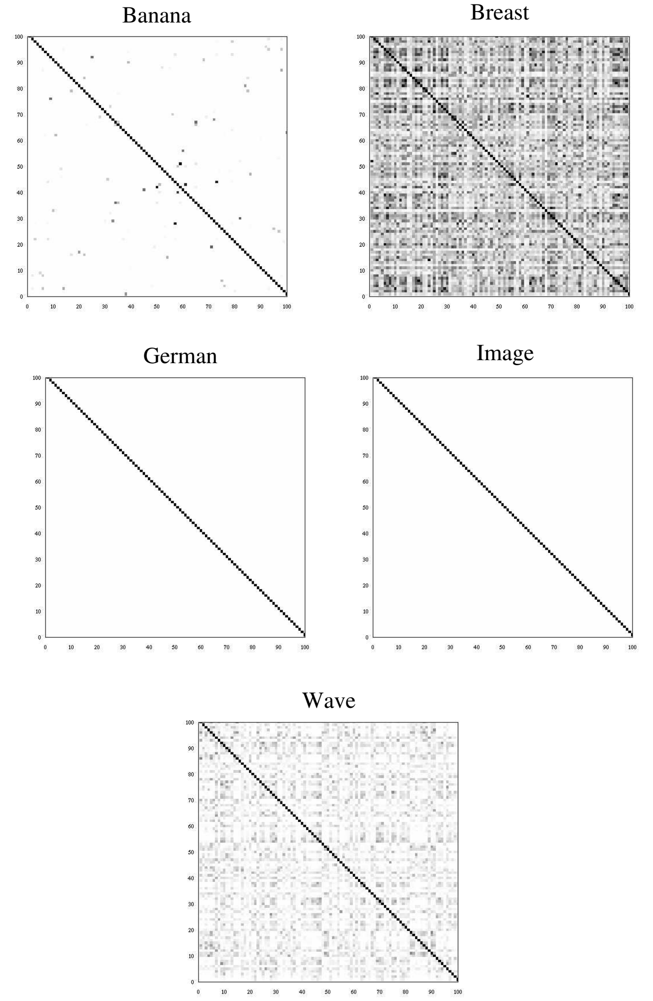

Figure 2. Optimal S matrices obtained calibrating QLYSVM on 5 UCI datasets.

4.2 Experiments on logic terms classification instances

We also performed some experiments involving a structured domain of logic terms. We consider three different problems.

Table 2 summarizes the characteristics of each problem. The first column reports the name of the problem, the second the set of symbols (with associated arity) compounding the terms, the third shows the rule(s) used to generate the positive examples of the problem[2], the fourth reports the number of terms in the training and test set respectively, the fifth the number of subterms in the training and test sets, and the last the maximum depth[3] of terms in the training and test sets. For example, let us consider the problem inst1_long. All the terms in the training and test sets are compounded by three different constants, i.e., **a**, **b**, and **c**, and a functional **f(·,·)** with two arguments. Terms which have to be classified positively are required to have the first argument equal to the second one (**f(X,X)**), e.g., both **f(f(a,f(b,c)),f(a,f(b,c)))** and **f(f(f(a,a),f(b,b)),f(f(a,a),f(b,b)))** have to be classified positively, while **f(a,f(b,b))** has to be classified negatively. For this problem, the training set contains 202 terms, while the test set contains 98 terms; the total number of distinct subterms for the training and test sets is 403 and 204, respectively. Finally, the maximal depth of the terms, both in the training and test sets, is 6.

For each problem about the same number of positive and negative examples is given. Both positive and negative examples were generated randomly. Training and test sets are disjoint and were generated by the same algorithm.

Note that the set of proposed problems ranges from the detection of a particular atom (label) in a term to the satisfaction of a specific unification pattern. Specifically, in the unification patterns for problem inst1_long, the variable X occurs twice making this problem much more difficult than inst4_long, because any classifier for this problem would have to compare arbitrary subterms corresponding to X.

As a final remark, it must be pointed out that the number of training examples is a very small fraction of the total number of terms characterizing each problem. In fact, the total number of distinct terms belonging to a problem is defined both by the number and arity of the symbols, and by the maximal depth allowed for the terms.

We treated this set of problems using a tree kernel. Therefore we give a description of the kernel function operating on trees that we adopted.

We have chosen the most popular and used Tree Kernel proposed in [4]. It is based on counting matching subtrees between two input trees. Given an input tree x, let s_x be a connected subtree of x. We assume that each of the m subtrees in the whole data set is indexed by an integer between 1 and m. Then $h_s(x)$ is the number of times the tree indexed with s occurs in x as a subtree. We represent each tree x as a feature vector $\phi(x) = [h_1(x), h_2(x), \ldots, h_m(x)]$. The inner product between two trees under the representation $\phi(x) = [h_1(x), h_2(x), \ldots h_m(x)]$ is:

$$K(x, y) = \phi(x) \cdot \phi(y) = \sum_{s=1}^{m} h_s(x) h_s(y)$$

Experimental results showed that this kernel may weight larger substructures too highly, producing a kernel matrix with large diagonals. This problem is treated in [4] through a method which dims the effect of the exponential blow-up in the number of subtrees with

[2] Note that the terms are all ground.

[3] We define the depth of a term as the maximum number of edges between the root and leaf nodes in the term's LDAG-representation.

Classification Problems

Problem	Symbols	Positive Examples.	#terms (tr.,test)	#subterms (tr.,test)	depth (pos.,neg.)
termocc1 very long	f/2 i/1 a/0 b/0 c/0	the (sub)terms i(a) or f(b,c) occur somewhere	(280,120)	(559,291)	(6,6)
inst1 long	f/2 a/0 b/0 c/0	instances of f(X,X)	(202,98)	(403,204)	(6,6)
inst4 long	f/2 a/0 b/0 c/0	instances of f(X,f(a,Y))	(290,110)	(499,245)	(7,6)

Table 2. Description of a set of classification problems involving logic terms.

their depth. The idea is to downweight larger subtrees modifying the kernel as follows:

$$K(x,y)=\sum_{s=1}^{m}\lambda^{\text{size}(s)}h_s(x)h_s(y)$$

where $0<\lambda\leq 1$ is a weighting parameter and size(s) is the number of nodes of the subtree s_x. The Tree Kernel can be calculated with a recursive procedure in $O(|NX|\cdot|NY|)$ time where NX and NY are the sets of nodes of trees x and y, respectively.

For the SVM algorithm we used a calibration procedure that is based on a $8\times 8\times 10$ grid involving powers of 10 for hyperparameters C, γ_K starting from 10×0.001, and with λ that varies from 0.1 to 0.9 with step 0.1.

For the QLYSVM, QLYSVM$_N$ and QLYSVM$_e$ algorithms we used a calibration procedure that involved a $8\times 8\times 10\times 8$ grid, that is obtained from the grid defined above by adding the hyperparameter γ_S varying from 0.001 to 10000 with powers of 10.

In both cases we selected the hyperparameters that obtained the best performance on a 3-fold cross validation on the training set.

For this dataset we also tested an exponential kernel obtained form the linear kernel matrix (QLYSVM Lin$_e$).

In this case the results do not lead to a clear improvement using the generalized quadratic loss. This can be due to an undersampling of the discrete input domain that does not allow to exploit the similarity based loss.

We however observe that also in this case the best results, within the generalized loss learning models, are obtained by (4), confirming that errors on correlated patterns of the same class occur more frequently than errors on correlated patterns of different classes.

5 Conclusions

In this paper we proposed a generalized quadratic loss for binary classification problems that addresses two issues: *i)* try to take into account correlation between patterns, *ii)* try to define a family of loss functions which takes into account the first issue.

The proposed generalized quadratic loss weights co-occurrence of errors on the basis of the similarity of the corresponding input patterns. Moreover errors of similar patterns of the same class are discouraged. We derived a SVM formulation for the proposed loss showing that if the similarity matrix is invertible the problem is equivalent to a hard margin SVM problem with a kernel matrix which depends also on the inverse of the similarity loss matrix. Experimental results on some binary classification tasks seems to show an improvement in the performance in several cases. Results on logic terms classification problems showed no improvement probably because of an undersampling of the discrete input domain that does not allow to exploit the similarity based loss.

Dataset	Algorithm	C	γ_K	γ_S	λ	Error %
inst1	SVM Lin	10	-	-	0.8	**9.18**
long	SVM RBF	10	1E-2	-	0.8	10.20
	QSVM RBF	10	1E-2	-	0.7	12.37
	QLYSVM RBF	1E3	1E-3	0.01	0.8	**9.18**
	QLYSVM Lin$_e$	10	-	-	0.5	13.27
	QLYSVM RBF$_e$	10	1E-2	0.1	0.7	11.22
	QLSVM RBF	100	1E-2	0.01	0.8	11.22
inst4	SVM Lin	10	-	-	0.6	**1.82**
long	SVM RBF	1E3	1E-3	-	0.7	**1.82**
	QSVM RBF	1E8	1E-3	-	0.5	2.76
	QLYSVM RBF	1E3	1E-2	10	0.5	4.55
	QLYSVM Lin$_e$	1E4	-	-	0.5	**1.82**
	QLYSVM RBF$_e$	1E3	1E-3	1E3	0.7	2.73
	QLSVM RBF	1E8	1E-3	1E4	0.5	**1.82**
termocc1	SVM Lin	1E3	-	-	0.3	**0.00**
very long	SVM RBF	1E3	1E-2	-	0.4	**0.00**
	QSVM RBF	1E8	1E-2	-	0.4	**0.00**
	QLYSVM RBF	1E3	1E-2	1E3	0.4	**0.00**
	QLYSVM Lin$_e$	1E8	-	-	0.3	**0.00**
	QLYSVM RBF$_e$	1E3	1E-2	1E3	0.8	**0.00**
	QLSVM RBF	1E8	1E-2	1E4	0.4	**0.00**

Table 3. Comparison of generalization error of SVM and QLYSVM on three logic terms classification instances.

A problem with this approach is the need to invert the similarity matrix. Thus further study will be devoted to this issue and to the extension of the framework to multiclass and regression problems.

REFERENCES

[1] D. Coppersmith and S. Winograd, 'Matrix multiplication via arithmetic progressions', *Journal of Symbolic Computation*, **9**(3), 251–280, (1990).

[2] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, 2000.

[3] T. Joachims, 'Text categorization with support vector machines: learning with many relevant features', in *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pp. 137–142, (1998).

[4] M.Collins and N.Duffy. Convolution kernels for natural language, 2001.

[5] C.A. Micchelli, 'Algebraic aspects of interpolation', in *Proceedings of Symposia in Applied Mathematics*, pp. 36:81–102, (1998).

[6] Zhang T. and Oles F.J., 'Text categorization based on regularized linear classification methods', *Information Retrieval*, **4**, 5–31, (2001).

[7] Michael E. Tipping, 'Sparse bayesian learning and the relevance vector machine', *Journal of Machine Learning Research*, **1**, 211–244, (2001).

[8] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.