Empirical Risk Minimization for Support Vector Classifiers

Fernando Pérez-Cruz, *Member, IEEE*, Angel Navia-Vázquez, *Member, IEEE*,
Aníbal R. Figueiras-Vidal, *Senior Member, IEEE*,
and Antonio Artés-Rodríguez, *Senior Member, IEEE*[*]

Dept. of Signal Theory and Communications, Universidad Carlos III de Madrid.
Avda. de la Universidad 30. 28911 Leganés, Madrid, Spain.

Abstract

In this paper, we propose a general technique for solving Support Vector Classifiers (SVCs) for an arbitrary loss function, relying on the application of an Iterative Re-Weighted Least Squares (IRWLS) procedure. We further show that three properties of the SVC solution can be written as conditions over the loss function. This technique allows the implementation of the Empirical Risk Minimization (ERM) inductive principle on large margin classifiers obtaining, at the same time, very compact (in terms of number of Support Vectors) solutions. The improvements obtained by changing the SVC loss function are illustrated with synthetic and real data examples.

1 Introduction

Support Vector Classifiers (SVCs) are state-of-the-art tools to solve pattern recognition problems [1, 2], able to build linear and nonlinear decision boundaries. Given a labeled training data set $((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$) and a nonlinear mapping usually to a higher dimensional space, ($\boldsymbol{\phi}(\cdot)$, $\mathbb{R}^d \xrightarrow{\boldsymbol{\phi}(\cdot)} \mathbb{R}^H$), we need to solve

$$\min_{\mathbf{w},b} \left\{ \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} L(e_i) \right\} \tag{1}$$

[*]This work has been partially supported by CICYT grant TIC2000-0380-C03-03.

where $e_i = y_i - (\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b)$ measures the difference between the desired output, y_i, and the achieved output, $\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b$. The SVC loss function, $L(e_i)$, is

$$L(e_i) = \begin{cases} 0, & e_i y_i < 0 \\ e_i y_i, & e_i y_i \geq 0 \end{cases} \tag{2}$$

It has been recently proven that the SVC minimizes a kind of generalization error measurement [3], but it does not minimize the number of erroneous decisions. The minimization of missclassifications is achieved following the Empirical Risk Minimization (ERM) inductive principle [1], which states that, to solve any learning problem from samples, one should minimize the number of training errors in order to minimize the error over the distribution function the samples were generated from. For a pattern recognition problem, this principle leads to a shifted step loss function over $e_i y_i$,

$$L(e_i) = \theta(e_i y_i - 1) = \begin{cases} 0, & e_i y_i < 1 \\ 1, & e_i y_i \geq 1 \end{cases} \tag{3}$$

but, solving (1) with this loss function is an NP-complete problem, as noted in [1].

We propose a generalized technique to solve the SVC for an arbitrary loss function and, following the ideas of Telfer and Szu [4], we will specifically use two approximations to the step loss function in (3). To formulate the SVC with an arbitrary loss function, we will join two previous proposals: the SVC with a hyperbolic tangent loss function, used to obtain reduced complexity classifiers [5], and a Polynomial loss function used to obtain unbiased SVC [6]. These loss functions can approximate to an arbitrary degree the loss function in (3), leading to the solution with the least number of training errors.

The SVC has been previously solved with other loss functions: in [7] $L(e_i) = e_i^2$ was used; and in [8, 9, 10] $L(e_i) = e_i^2$ only for $e_i y_i \geq 0$ was proposed. These loss functions were introduced to avoid the application of Quadratic Programming (QP). For regression problems, a research linking the optimal loss functions with probability densities of the target variable y_i was carried out in [11], although the problem was only solved for convex loss functions.

Here, we will show that the SVC solution can be obtained for any convex or non-convex loss function using an Iterative Re-Weighted Least Squares (IRWLS) procedure, which has been formulated in [12, 13] to solve the SVC and in [14, 15] for Support Vector Regressors. We will

also present three conditions over the loss function that will guarantee that the SVC presents bounded solutions, leads to maximum margin classifier for separable problems, and leads to convex optimization problem.

The paper is structured as follows. The SVC optimization procedure with an arbitrary loss function is detailed in Section 2. The loss function analysis for pattern recognition problems is carried out in Section 3. Section 4 is devoted to illustrate, by means of computer experiments, the properties of several loss functions. We close with some concluding remarks in Section 5.

2 SVC with a general loss function.

The SVC is usually posed as a constrained optimization procedure [17],

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i L(\xi_i) \right\} \tag{4}$$

subject to

$$y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{5}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{6}$$

$$L(\xi_i) = \xi_i \tag{7}$$

To solve (4), linear restrictions (5) and (6) are introduced, using Lagrange multipliers, α_i and μ_i, thus arriving to the minimization of

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i L(\xi_i) - \sum_i \mu_i\xi_i - \sum_i \alpha_i(y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) - 1 + \xi_i) \tag{8}$$

with respect to $\mathbf{w}$, ξ_i, and b, and to its maximization with respect to α_i and μ_i. The solution is defined by the Karush-Kuhn-Tucker theorem [18], which imposes several conditions on the

variables in (8), namely the KKT conditions, conditions that are: (5), (6), and

$$\frac{\partial L_p}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i) = \mathbf{0} \tag{9}$$

$$\frac{\partial L_p}{\partial b} = -\sum_i \alpha_i y_i = 0 \tag{10}$$

$$\frac{\partial L_p}{\partial \xi_i} = C\frac{dL(\xi_i)}{d\xi_i} - \alpha_i - \mu_i = 0 \qquad \forall i = 1, \ldots, n \tag{11}$$

$$\alpha_i, \mu_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{12}$$

$$\alpha_i\{y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) - 1 + \xi_i\} = 0 \qquad \forall i = 1, \ldots, n \tag{13}$$

$$\mu_i \xi_i = 0 \qquad \forall i = 1, \ldots, n \tag{14}$$

Conventional solutions for the SVC, by means of QP procedures with the loss function (7), require to substitute (9), (10) and (11) into (8), obtaining a quadratic functional, known as the dual problem [17]. If the loss function is neither (7) nor $L(\xi_i) = \xi_i^2$, the dual problem is not quadratic (nor is the primal problem) and the SVC cannot be solved by means of QP. An IRWLS has been proposed [12, 13] to get a solution for the loss function in (7); we will show that such a procedure can be used to solve the SVC using any loss function.

Accordingly, we need to minimize

$$\begin{aligned} L_P &= \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i \alpha_i(1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b)) + \sum_i (CL(\xi_i) - \xi_i(\alpha_i + \mu_i)) \\ &= \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\sum_i a_i e_i^2 + \sum_i (CL(\xi_i) - \xi_i(\alpha_i + \mu_i)) \end{aligned} \tag{15}$$

with respect to $\mathbf{w}$, ξ_i, and b, where

$$a_i = \frac{2\alpha_i}{1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b)} \tag{16}$$

$$e_i = y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b \tag{17}$$

L_P in (15) comprises a least squares functional plus a regularizing term ($\frac{1}{2}\|\mathbf{w}\|^2$) and a term depending on ξ_i in (17). From (5) and the definition of e_i, it can be seen that

$$\xi_i = \xi_i(e_i) = \xi_i(\mathbf{w}, b) = \begin{cases} 0, & e_i y_i < 0 \\ e_i y_i, & e_i y_i \geq 0 \end{cases} \tag{18}$$

The minimum of (15) cannot be obtained in a single step, because $a_i = a_i(e_i)$, we can apply an IRWLS procedure [19], that alternatively minimizes (15) considering fixed the values of a_i

and recalculates a_i with the obtained $\mathbf{w}$ and b.

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \mathbf{\Phi}^T \mathbf{D}_a [\mathbf{y} - \mathbf{\Phi}\mathbf{w} - \mathbf{1}b] + \left(C\frac{dL(\boldsymbol{\xi})}{d\boldsymbol{\xi}} - (\boldsymbol{\alpha} + \boldsymbol{\mu}) \right)^T \nabla_{\mathbf{w}} \boldsymbol{\xi} = \mathbf{0} \tag{19}$$

$$\frac{\partial L_P}{\partial b} = -\mathbf{a}^T [\mathbf{y} - \mathbf{\Phi}\mathbf{w} - \mathbf{1}b] + \left(C\frac{dL(\boldsymbol{\xi})}{d\boldsymbol{\xi}} - (\boldsymbol{\alpha} + \boldsymbol{\mu}) \right)^T \nabla_b \boldsymbol{\xi} = 0 \tag{20}$$

where $\mathbf{a}$, $\mathbf{y}$, $\boldsymbol{\alpha}$, $\boldsymbol{\xi}$, $\boldsymbol{\mu}$ and $\mathbf{1}$ are n-dimensional column vectors, having obvious expressions, and

$$\mathbf{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]^T \tag{21}$$

$$(\mathbf{D}_a)_{ij} = a_i \delta_{ij} \qquad \forall i, j = 1, \ldots, n \tag{22}$$

$$(\nabla_{\mathbf{w}} \boldsymbol{\xi})_{ij} = \frac{\partial \xi_i}{\partial w_j} \qquad \forall i = 1, \ldots, n \quad \forall j = 1, \ldots, H \tag{23}$$

$$\nabla_b \boldsymbol{\xi} = \left[\frac{\partial \xi_1}{\partial b}, \frac{\partial \xi_2}{\partial b}, \ldots, \frac{\partial \xi_n}{\partial b} \right]^T \tag{24}$$

and δ_{ij} is the Kronecker delta. Last term in (19) and (20) includes the KKT condition (11), expressed in matrix notation. We drop these terms from (19) and (20), because, at the solution, all KKT conditions must hold. Joining (19) and (20), we construct the equations:

$$\begin{bmatrix} \mathbf{\Phi}^T \mathbf{D}_a \mathbf{\Phi} + \mathbf{I} & \mathbf{\Phi}^T \mathbf{a} \\ \mathbf{a}^T \mathbf{\Phi} & \mathbf{a}^T \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}^T \mathbf{D}_a \mathbf{y} \\ \mathbf{a}^T \mathbf{y} \end{bmatrix} \tag{25}$$

The solution of (25) provides the values of $\mathbf{w}$ and b for the first step of the IRWLS procedure. Linear equations in (25) require to explicitly know the nonlinear mapping $\boldsymbol{\phi}(\cdot)$, which is seldom the case. To solve (25) using Reproducing Kernels Hilbert Space (RKHS), as in the classical SVC formulation [1], we need to stipulate that

$$\mathbf{w} = \sum_{i=1}^{n} \beta_i \boldsymbol{\phi}(\mathbf{x}_i) = \mathbf{\Phi}^T \boldsymbol{\beta} \tag{26}$$

The system of equations that results, when (26) is introduced into (25) (see [20]), is

$$\begin{bmatrix} \mathbf{H} + {\mathbf{D}_a}^{-1} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \beta \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \tag{27}$$

where $(\mathbf{H})_{ij} = \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$.

From the solution of (27) we have to recalculate the value of vector $\mathbf{a}$, this is done enforcing the KKT conditions (5), (6), (11), (12), (13) and (14), where:

$$a_i = \frac{2\alpha_i}{y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{\Phi}^T\mathbf{D}_y\boldsymbol{\beta} + b)} = \begin{cases} 0, & e_i y_i < 0 \\ \dfrac{2C}{y_i e_i} \dfrac{dL(\xi_i)}{d\xi_i}\Big|_{\xi_i}, & e_i y_i \geq 0 \end{cases} \tag{28}$$

In summary, the IRWLS procedure consists in the following steps:

1. Solve (27) to obtain $\boldsymbol{\beta}$ and b.
2. Compute the value of Eq. 4 using the present $\boldsymbol{\beta}$ and b (that we denote as $\boldsymbol{\beta}_{pres}$ and b_{pres}) and compare them with value of Eq. 4 using the values $\boldsymbol{\beta}$ and b obtained in the previous iteration (that we denote as $\boldsymbol{\beta}_{prev}$ and b_{prev}). If the present value of Eq. 4 is less than the previous one, go to Step 3, otherwise find the value of $k \in [0, 1]$ that minimizes Eq. 4 fixing $\boldsymbol{\beta}_{pres} = (1-k)\boldsymbol{\beta}_{prev} + k\boldsymbol{\beta}_{pres}$ and $b_{pres} = (1-k)b_{prev} + kb_{pres}$.
3. Recalculate $\mathbf{a}$ with the obtained $\boldsymbol{\beta}$ and b using (28).
4. Repeat until convergence.

For non-convex loss function, as we will proposed in the next section, the problem is no longer convex and might, therefore, present local minima. In this scenario the the IRWLS procedure guarantees to converge to one of them, dealing with not ending in a local minima is covered by the next section, as well.

3 Loss functions for Pattern Recognition

In the previous section we have shown the SVC can be solved with any loss function $L(\cdot)$, using an IRWLS procedure. We did not impose any condition on $L(\cdot)$ and, eventually, any one could be used. In this section, we will present sufficient conditions that can be imposed on a particular loss function to guarantee that the solution is of a specific type. The loss function under consideration must be an increasing function of ξ_i, because otherwise the loss function will infringe a higher penalty to samples closer to its correct class than those further apart.

Condition 1 : $\left.\dfrac{dL(\xi_i)}{d\xi_i}\right|_{\xi_i} < \infty \quad \forall \xi_i \geq 0$ *to ensure that* $\mathbf{w}$ *is finite.*

The classifier is defined as a linear combination of the training samples weighted by values α_i, see (9). In order to guarantee that $\mathbf{w}$ is finite, we need to guarantee that every α_i is finite.

These α_i can be obtained from the KKT conditions (specifically (11), (13) and (14)) as

$$\alpha_i = \begin{cases} C\dfrac{dL(\xi_i)}{d\xi_i}\Big|_{\xi_i} - \mu_i, & \xi_i = 0 \\ C\dfrac{dL(\xi_i)}{d\xi_i}\Big|_{\xi_i}, & \xi_i > 0 \end{cases} \tag{29}$$

which leads to Condition 1. ■

Condition 2 : $\dfrac{dL(\xi_i)}{d\xi_i}\Big|_{\xi_i=0} > 0$ *to obtain maximum margin classifiers for linearly separable problems in the feature space.*

If the training set is linearly separable in the feature space, every ξ_i can be zero because ξ_i are positive slack variables introduced to accommodate non-separable problems [17], and nonzero α_i are given by those training samples that fulfill (5) with equality sign, being $\alpha_i \in \left[0, C\dfrac{dL(\xi_i)}{d\xi_i}\Big|_{\xi_i=0}\right]$. If $\dfrac{dL(\xi_i)}{d\xi_i}\Big|_{\xi_i=0} = 0$, not every ξ_i can be zero, because in that case $\mathbf{w} = \mathbf{0}$, and if any ξ_i is nonzero, one cannot guarantee that a maximum margin classifier is obtained. ■

Condition 3 : $\dfrac{d^2L(\xi_i)}{d\xi_i^2} \geq 0 \quad \forall \xi_i \geq 0$ *to obtain a single minimum functional.*

To guarantee that a single set of variables ($\mathbf{w}^o$, b^o, ξ_i^o, α_i^o and μ_i^o) fulfills the KKT conditions, it is needed that (4) and constraints (5) and (6) be convex [22]. Constraints are convex and (4) is convex iff the loss function is convex. ■

Among the above conditions, only the first seems to be critical, because a loss function not fulfilling it might lead to infeasible solutions, i.e. $\mathbf{w} = \boldsymbol{\infty}$. The second condition is not critical for non-separable problems, but for linearly separable problems in the feature space one could directly used (7), that fulfills such property.

Loss function that do not fulfill the third condition, as the ones we propose in here, might present local minima. We propose a two-step procedure for optimizing the SVC with non-convex loss function. In the first step, we solve the original SVC (loss function (7)), because this is a convex problem known to be a good solution to most learning problems [1]. In the second step, we solve the SVC with the non-convex loss function using as initial values of $\boldsymbol{\beta}$ and b the solution attained in the first step. This way of training does not guarantee that the global minimum is found, but it does ensure that the achieved solution presents less empirical error that the SVC with its original loss function.

Most appropriate approximations to the step function will present a zero first derivative for most ξ_i values, leading to a large number of training samples with $\alpha_i = 0$ and very compact machines as a consequence, which is a desirable property when classes overlap [16].

We will analyze the following loss functions:

- "Vapnik" loss function:

$$L(\xi_i) = \xi_i^s \tag{30}$$

- Sigmoid loss function:

$$L(\xi_i) = \frac{\tanh(\eta(\xi_i - 1)) + 1}{2} \tag{31}$$

- Polynomial loss function:

$$L(\xi_i) = \begin{cases} \dfrac{(\xi_i - k_\varepsilon^{1/s})^s}{2(1 - k_\varepsilon^{1/s})^s}, & \xi_i < 1 \\ 1 - \dfrac{(1 - k_\varepsilon^{1/s})^s}{2(\xi_i - k_\varepsilon^{1/s})^s}, & 1 \leq \xi_i \end{cases} \tag{32}$$

The value k_ε is a nonzero constant, needed to fulfill the second condition.

The Sigmoid and Polynomial loss functions actually implement the ERM inductive principle because they can approximate to an arbitrary degree the shifted step loss function (3). On the other hand "Vapnik" loss function tends to $\theta(\xi_i)$ (the non shifted step loss function) as $s \to 0$, counting as errors samples with $\xi_i \in (0, 1)$, which in fact are not. Moreover "Vapnik" loss function might lead to infeasible solutions for $s \leq 1$, because it does not fulfill the first condition. Sigmoid and Polynomial loss functions do not fulfill the third condition.

4 Computer experiments

We now illustrate the use of SVCs with several loss functions different from that traditionally employed. We will first show the advantages of using a loss function that resembles the shifted step loss function for a simple classification problem. Then, we solve 7 learning problems taken from the "UCI Machine Learning Repository" to show the advantages of modifying the loss function in real machine learning problems.

4.1 Classification of a synthetic data set

We have devised a simple one-dimensional problem to illustrate the need for a different loss function from (7). It consists in a two-class problem with the following conditional densities:

$$p(\mathbf{x}|y=1) = \frac{3}{4}N(0.75, 0.15) + \frac{1}{4}N(0.2, 0.02) \qquad p(\mathbf{x}|y=-1) = U(0, 0.5)$$

with $p(y=1) = \frac{2}{3}$. We have depicted these densities in Figure 1(a). Here, $N(\mu, \sigma)$ represents a Gaussian probability density function (p.d.f.) having mean μ and standard deviation σ, and $U(l, h)$ represents a uniform p.d.f. with l and h, respectively, as the lowest and highest attainable values. We first solve this problem with a linear RKHS employing several loss functions:

- "Vapnik" loss function with $s = 1$.
- "Vapnik" loss function with $s = 2$, as suggested in [8].
- Sigmoid loss function (31) with $\eta = 1, 2, 3$, as used in [5].
- Polynomial loss function (32) with $s = 2, 4$, as used in [6].

We also have employed to solve this learning problem the Least Square Support Vector Machines (LS-SVMs) proposed in [7]. We work with 20 training sets with 400 training samples each and we set $C = 4096$, which has been selected using cross-validation. For each loss function, we can compute a threshold value, because it is a one dimensional problem: threshold is $th = -b/w$. Samples with $x_i > th$ will belong to class $+1$, and otherwise to class -1. We have plotted in Figure 1b the probability of error for every threshold, defined by

$$P_{err} = \int_{-\infty}^{th} p(x|y=+1)dx + \int_{th}^{\infty} p(x|y=-1)dx \tag{33}$$

We show in Table 1 the value of th, the train error, the probability of error according to (33), and the number of Support Vectors (SVs). We observe that the Sigmoid and the Polynomial loss functions yield solutions that are very close to the optimal linear classifier, their results not differing significantly. "Vapnik" loss function and the LS-SVM are not able to obtain a solution close to the optimal linear solution because the samples belonging to $N(0.2, 0.02)$ pull the solution towards them. "Vapnik" loss function with $s = 1$ presents better results than "Vapnik" loss function with $s = 2$ and than the LS-SVM, because the former is the best convex

approximation to the step loss function. Additionally, both the Sigmoid and Polynomial loss functions yield a nearly optimal solution with significantly fewer support vectors than using (7).

The results achieved by the Polynomial and Sigmoid loss functions were alike for this experiment and others carried out, therefore we will only report those of the Sigmoid loss function, but one must expect similar results if the Polynomial were used. "Vapnik" loss function with $s = 1$ achieved lower probability of error than the rest of the convex loss function due to is closer to the shifted step loss function as noted in [23], therefore the used of any other convex will perform as or worse than the SVC with "Vapnik" loss function with $s = 1$. From now on, we will only compare the SVC with the original loss function and the Sigmoid (ERM-SVC).

The optimal Bayesian classifier for this problem is nonlinear, and any linear decision surface will always lead to suboptimal classifiers. To achieve a solution close to the optimal Bayesian classifier (classification error of 8.37%), we have solved this problem with two nonlinear kernels: a Polynomial RKHS, $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^n$, with $n = 3$[1]; and a Radial Basis Function (RBF) RKHS, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$, with $\sigma = 0.5$ (selected by cross-validation). We have considered 20 training sets with 400 samples each and its corresponding test sets with 4000 samples. We have plotted in Figure 2 the error rate over the test and training set for the original and Sigmoid loss functions.

The SVC with the Sigmoid loss function achieves for both kernels a test error performance that is better than that achieved with the original loss function and very close to the Bayes error. It can be noticed that the training error is below the Bayes error, which is not an infeasible situation, because the learning machine is biased by the training samples. This is even the way machine learning algorithms must perform, as noted in [1], where it is explained that the train error must tend to the Bayes error from below and the test error must tend to the Bayes error form above, as the number of training samples increases.

[1] From figure 1, we can understand that the optimal solution for this problem is three thresholds (Class +1: $0.17 \leq x \leq 0,23$ and $x \geq 0.5$; Class -1 $x < 0.17$ and $0.23 < x < 0,5$, we do not take into consideration what happens for $x < 0$ because no sample will be beyond that point), therefore a 3^{rd} degree Polynomial has sufficient versatility to achieved the optimal thresholds.

4.2 Experiments with "UCI Machine Learning Repository"

Once we have shown for a synthetic data set that the performance of the proposed algorithm is the expected, i.e. it finds the minimum empirical error over the training set and leads to a better solution on the test set, we are going to test the proposed algorithm for some well known data sets from "http://www.ics.uci.edu/~mlearn/MLSummary.html". We have taken the data sets: Balance, Car, Contraceptive, Ecoli, Haberman, Liver and Tic-Tac-Toe.

The Balance data set contains 576 left or right-unbalanced samples and 4 features. The Car data sets contains 1728 samples and 6 ordered attributes and we have formed the positive class from the subclasses "acc", "good" and "v-good". The Contraceptive data set contains 1473 samples and 9 attributes, that measure the way of living of Indonesian women in 1987 and the goal is to know whether she takes contraceptive measures. The Ecoli data set contains 336 samples and 7 predictive attributes. Haberman data set contains 306 samples and 3 attributes and the goal is to predict if a woman survives more than 5 years after a breast cancer surgery. The Liver data set contains 345 samples and 6 predictive attributes. The Tic-Tac-Toe data set contains 958 samples and 9 attributes.

We have addressed these problems with two loss functions: the original SVC ("Vapnik" with $s = 1$), and the Sigmoid with $\eta = 2$, because we have already observed there is no significant difference in the results when using the Polynomial loss function. We have employed two types of kernels: linear and RBF, and the all features in every the data set have been preprocessed to present zero mean and unit standard deviation, the ordered attributes have been first represented by natural numbers. The data sets have been split 20 times, in 80% of samples for training and 20% for testing purposes. The obtained results are shown in Table 2 for the linear kernel and in Table 3 for the RBF kernel. The values of C and σ have been set using 8-fold cross-validation over the training set.

We point out that in all the cases except for the Ecoli data set results using the Sigmoid loss function are better than using "Vapnik" loss function for the linear RKHS. For the RBF kernel, the SVM gets a better result with "Vapnik" loss function in Ecoli and Haberman data sets and the results are better with the Sigmoid loss function for Car, Contraceptive and Liver. For the other two data sets (Balance and Tic-Tac-Toe), the problems are separable and both loss functions obtain the same result.

The data sets in which the results are similar for both loss functions (Haberman and Ecoli) are those that give a similar testing error for the linear and RBF kernel. For the Haberman data set, the best result is obtained with the Sigmoid loss function and the linear kernel. Then, if one had to select the best machine for each data set, we would choose the SVM with the Sigmoid loss function in every case except for the Ecoli data set.

The results for the Ecoli data set are hardly significant due to the reduced number of training and test errors, respectively, below 10 and 3 errors. We would need more training samples to properly draw conclusive arguments about whether this is just an atypical result due to the lack of train and test samples (or errors) or it is an atypical problem in which the original loss function is best.

The Tic-Tac-Toe is a linearly separable set in the feature space, when using an RBF kernel with $\sigma = \sqrt{d}/2$. So the solutions with both loss functions are equal to me maximum margin solution (Note that both (7) and (31) obey condition 2). The SVC with (7) and linear kernel reaches the minimum empirical error solution over the Tic-Tac-Toe data set, due to the symmetries of the underlying distributions. This also explains why the SVC with both, loss function (7) and (31), obtains the same solution (the value of C does not affect the solution over the tested range from 0.1 to 1000).

In the other four data sets the results are as expected, i.e. the empirical error is reduced and so is the test error. It can also be noted that the observed gains using linear kernels are more significant than those with RBFs kernels. This is due to the values the slack variables, ξ_i, take. For the RBF kernel, the slack variables do not present very large value (largest value around 3), meaning that the achieved solution is not far apart from the minimum of the empirical risk. In the case of the linear kernel, the slack variables tend to present larger values (over 10), so that the SVC solution is not as close as it could be to the empirical error and the change in the loss function can provide a much better solution. As the number of available samples increases the minimum of the empirical error tends to the Bayes classifier, which is not the attained solution by the SVC with (7) except in special cases (as the Tic-Tac-Toe data set) as noted in [8].

We have included the standard deviation for the 20 trials in which one can see that for the linear kernel the SVC with the Sigmoid loss function performs significantly better than the SVC with the original loss function, except in the Ecoli case as already commented. This result is

not so clear for the RBF kernel in which the results are not significantly different. This fact can be explained by the fact that when using the RBF kernel the values of ξ_i do not take very large values (seldom greater than 2) so although the loss function is not an appropriate approximation to the step in that range ($\xi_i < 2$) it is very similar and not departing the solution significantly from the solution of minimum empirical error. As the number of samples increases the number of samples with error greater than 1 will start effecting the solution and the use of the Sigmoid will improve with respect to the classic SVM solution.

Finally, we will illustrate the convergence behaviour of the IRWLS for non-convex loss functions. We have computed the value of (4) for each iteration in the two-step procedure for solving the SVC with non-convex loss function. We have used the Liver data set with RBF kernel, $C = 10$ and $\sigma = 2\sqrt{d}$. In Figure 3a, we show the value of (4) for "Vapnik" loss function with $s = 1$ for the first 31 iterations (1^{st} step). In Figure 3b, we show the value of (4) for Sigmoid loss function with $\eta = 2$ from 32^{nd} to 47^{th} iterations (2^{nd} step). We can observe from these figures that the IRWLS procedure decreases (4) in each iteration for both loss functions.

5 Conclusions

We have presented a general technique to train Support Vector Classifiers (SVCs) for an arbitrary loss function using an Iterative Re-Weighted Least-Squares (IRWLS) algorithm. We have also shown the relationship between three SVC properties (namely, weight finiteness, maximum margin, and convexity of the functional to be minimized) and their corresponding conditions on the loss function. Based on these results, we have demonstrated the feasibility of implementing large margin classifiers following an approximation to the Empirical Risk Minimization (ERM) inductive principle by means of either Sigmoid or Polynomial loss functions. The improve with these loss function is more significant for the linear kernel and the solution for a given problem also improves as the number of training samples increases

The resulting classifiers show better performance in both probability of missclassification and number of SVs than those trained following the conventional SVC loss function across a wide range of experiments and databases.

References

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

[2] B. Schölkopf and A. Smola, *Learning with kernels*, M.I.T. Press, to appear.

[3] J. Shawe-Taylor and N. Cristianini, "On the generalisation of soft margin algorithms," Tech. Rep. NC-TR-00-082, Royal Holloway College, University of London, UK, 2000,

[4] B. A. Telfer and H. H. Szu, "Energy function for minimizing missclasification error with minimum-complexity networks," *Neural Networks*, pp. 214–219, 1994.

[5] F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez, "Support vector classifier with hyperbolic tangent penalty function," in *Proc. of the ICASSP'00*, Istambul, Turkey, June 2000.

[6] A. Navia-Vázquez, F. Pérez-Cruz, A. Artés-Rodríguez, and A. R. Figueiras-Vidal, "Unbiased support vector classifer," in *Proc. Neural Network for Signal Processing NNSP'01 Intl. Conference*, Falmouth, Boston, MA, september 2001, pp. 183–192.

[7] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[8] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machine*, Cambridge University Press, 1999.

[9] O. L. Mangasarian and D. Musicant, "Active support vector machine classification," in *Advances in Neural Information Processing Systems 13*, Cambridge, MA, Nov. 2000, M.I.T. Press.

[10] M. Lehtokangas, "Pattern recognition with novel support vector machine learning method," in *Proc. of the EUSIPCO'00*, Tampere, Finland, Sept. 2000.

[11] A. J. Smola, B. Schölkopf, and K.-R. Müller, "General cost functions for support vector regression," in *Proc. of the Ninth Australian Conf. on Neural Networks*, Brisbane, Australia, 1998, pp. 79–83.

[12] F. Pérez-Cruz, A. Navia-Vázquez, J. L. Rojo-Álvarez, and A. Artés-Rodríguez, "A new training algorithm for support vector machines," in *Proc. of the Fifth Bayona Workshop on Emerging Technologies in Telecommunications*, Baiona, Spain, Sept. 1999, pp. 116–120.

[13] A. Navia-Vázquez, F. Pérez-Cruz, A. Artés-Rodríguez, and A. R. Figueiras-Vidal, "Weighted least squares training of support vectors classifiers which leads to compact and adaptive schemes," *IEEE transactions on Neural Networks*, no. 5, pp. 1047–1059, 9 2001.

[14] F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez, "An IRWLS procedure for SVR," in *Proc. of the EUSIPCO'00*, Tampere, Finland, Sept. 2000.

[15] F. Pérez-Cruz and A. Artés-Rodríguez, "An irwls procedure for nu-svr," in *Proc. of the ICASSP'01*, Salt Lake City, (Utah U.S.A.), May 2001,

[16] C. J. C. Burges, "Simplified support vector decision rules," in *Proc. 13th Int. Conference on Machine Learning*, L. Saitta, Ed., San Mateo, CA, 1996, pp. 71–77, Morgan Kaufmann.

[17] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Knowledge Discovery and Data Mining*, vol. 2, no. 2, pp. 121–167, 1998.

[18] R. Fletcher, *Practical Methods of Optimization*, Wiley, Chichester, second edition, 1987.

[19] P. W. Holland and R. E. Welch, "Robust regression using iterative re-weighted least squares," *Communications of Statistics Theory Methods*, vol. A6, no. 9, pp. 813–27, 1977.

[20] F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez, "SVC-based equalizer for burst TDMA transmissions," *Signal Processing*, vol. 81, no. 8, pp. 1681–1693, Aug. 2001.

[21] F. Pérez-Cruz, P. L. Alarcón-Diana, A. Navia-Vázquez, and A. Artés-Rodríguez, "Fast training of support vector classifiers," in *Advances in Neural Information Processing Systems 13*, Cambridge, MA, Nov. 2000, M.I.T. Press.

[22] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley, 1992.

[23] O. L. Mangasarian and D. R. Musicant, "Lagrangian support vector machines," *Journal of Machine Learning Research*, pp. 161–177, 2000.

loss function	th	Tr. Err.	Prob. Error	Nsv
"Vapnik" $s = 1$	0.3763	0.2446	0.2526	197.28
	(0.0181)	(0.0241)	(0.0116)	(10.02)
"Vapnik" $s = 2$	0.3272	0.2779	0.2832	292.0
	(0.0234)	(0.0305)	(0.0150)	(10.43)
LS-SVM	0.3197	0.2794	0.2868	400
	(0.0247)	(0.0373)	(0.0164)	(0)
Sigmoid $\eta = 1$	0.4984	0.1779	0.1922	13.61
	(0.0044)	(0.0143)	(7.98×10^{-4})	(4.23)
Sigmoid $\eta = 2$	0.5000	0.1778	0.1918	10.00
	(0.0036)	(0.0141)	(7.31×10^{-4})	(1.89)
Sigmoid $\eta = 3$	0.4999	0.1774	0.1918	9.44
	(0.0036)	(0.0146)	(8.10×10^{-4})	(1.83)
Poly $s = 2$	0.5004	0.1778	0.1919	9.94
	(0.0047)	(0.0180)	(0.001)	(1.68)
Poly $s = 4$	0.5000	0.1772	0.1918	24.26
	(0.0047)	(0.0185)	(9.99×10^{-4})	(20.82)
Optimal linear Cl.	0.5		0.1906	

Table 1: Results for the synthetic data set with linear kernel. We show threshold (th), training error rate (Tr. Err.), probability of error (prob. Error) computed according to (33), and number of support vectors (Nsv) for the proposed loss functions. We show mean and standard deviation, in brackets, for 20 sets.

data set name	loss function	C	Tr. Err.	Ts. Err.
Balance	"Vapnik" $s = 1$	100	4.65% (0.56)	5.77% (0.60)
	Sigmoid $\eta = 2$	100	3.25% (0.66)	4.98% (0.72)
Car	"Vapnik" $s = 1$	35	13.42% (0.74)	13.42% (0.99)
	Sigmoid $\eta = 2$	35	10.95% (0.68)	11.53% (0.89)
Contraceptive	"Vapnik" $s = 1$	35	31.00% (0.41)	32.81% (0.61)
	Sigmoid $\eta = 2$	1000	27.27% (0.44)	31.86% (0.58)
Ecoli	"Vapnik" $s = 1$	35	2.56% (0.05)	2.61% (0.19)
	Sigmoid $\eta = 2$	35	2.14% (0.03)	2.80% (0.20)
Haberman	"Vapnik" $s = 1$	100	26.24% (0.51)	25.78% (0.92)
	Sigmoid $\eta = 2$	100	22.84% (0.73)	24.31% (0.85)
Liver	"Vapnik" $s = 1$	3.5	29.24% (1.2)	31.52% (1.3)
	Sigmoid $\eta = 2$	3.5	25.89% (1.1)	30.51% (1.2)
Tic-Tac-Toe	"Vapnik" $s = 1$	0.1-1000	1.60% (0.22)	1.83%(0.50)
	Sigmoid $\eta = 2$	0.1-1000	1.60% (0.22)	1.83% (0.50)

Table 2: Results for the UCI Machine Learning repository data sets with linear kernel. We show the train and test error rate for each data set, loss function, and corresponding hyperparameter. The values in brackets indicates the standard deviation over 20 trials.

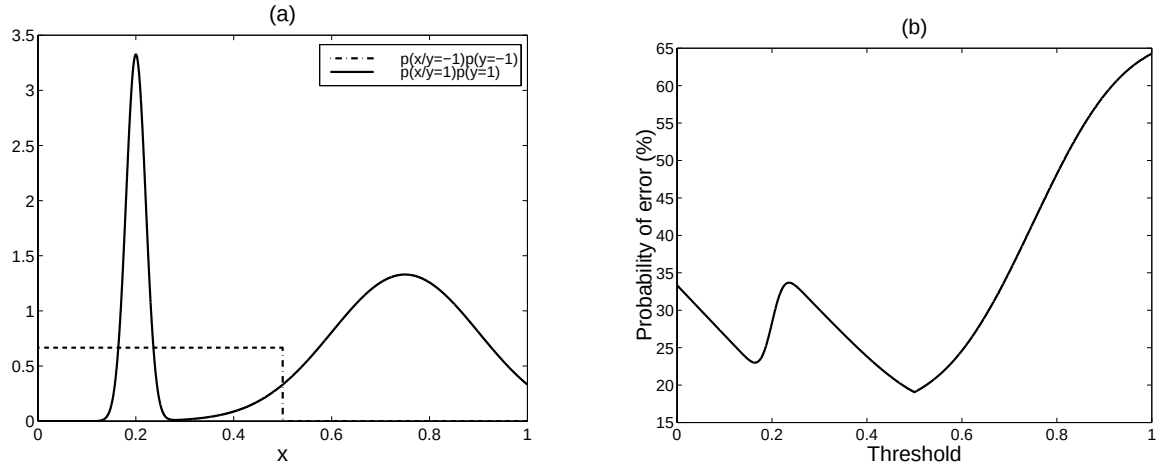

Figure 1: (a) The p.d.f. for class +1 (solid line) and class −1 (dash-dotted line). (b) Probability of error according to (33) for each threshold.

data set name	loss function	σ	C	Tr. Err.	Ts. Err.
Balance	"Vapnik" $s = 1$	$2\sqrt{d}$	35	0% (0)	0% (0)
	Sigmoid $\eta = 2$	$2\sqrt{d}$	35	0%(0)	0% (0)
Car	"Vapnik" $s = 1$	$\sqrt{d}/2$	3.5	0.02% (0.01)	0.73% (0.02)
	Sigmoid $\eta = 2$	$\sqrt{d}/2$	10	0.00% (0.00)	0.69% (0.01)
Contraceptive	"Vapnik" $s = 2$	$2\sqrt{d}$	35	24.23% (0.80)	28.58% (0.83)
	Sigmoid $\eta = 2$	$2\sqrt{d}$	35	22.38% (0.80)	28.34% (0.85)
Ecoli	"Vapnik" $s = 1$	$\sqrt{d}$	3.5	2.09% (0.45)	2.80% (0.80)
	Sigmoid $\eta = 2$	$\sqrt{d}$	35	2.09% (0.31)	2.99% (0.95)
Haberman	"Vapnik" $s = 1$	$2\sqrt{d}$	10	23.91% (1.2)	24.65% (3.2)
	Sigmoid $\eta = 2$	$2\sqrt{d}$	1	21.83% (1.4)	24.93% (3.3)
Liver	"Vapnik" $s = 1$	$2\sqrt{d}$	10	23.55% (1.5)	27.31% (3.5)
	Sigmoid $\eta = 2$	$2\sqrt{d}$	3.5	24.08% (1.3)	27.09% (3.8)
Tic-Tac-Toe	"Vapnik" $s = 1$	$\sqrt{d}/2$	3.5	0% (0)	0.3% (0.34)
	Sigmoid $\eta = 2$	$\sqrt{d}/2$	3.5	0% (0)	0.3% (0.34)

Table 3: Results for the UCI Machine Learning repository data sets with RBF kernel. We show the train and test error rate for each data set, loss function, and corresponding hyperparameters. The values in brackets indicates the standard deviation over 20 trials.

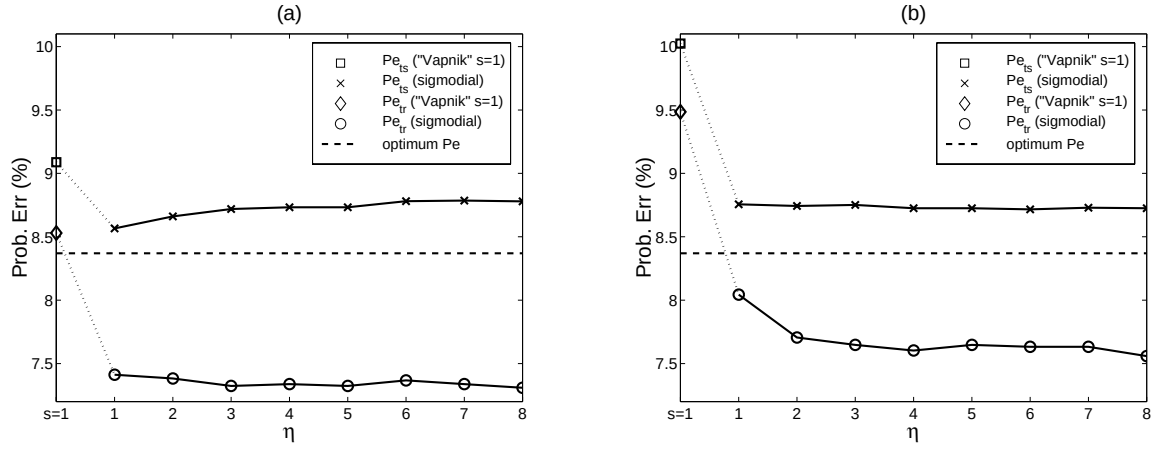

Figure 2: (a) Train and test error rate for the synthetic data set with RBF kernel for "Vapnik" and Sigmoid loss function. (b) Train and test error rate for the synthetic data set with polynomial kernel for "Vapnik" and Sigmoid loss function. The dashed line represents the probability of error of the optimal bayesian classifier in both plots.

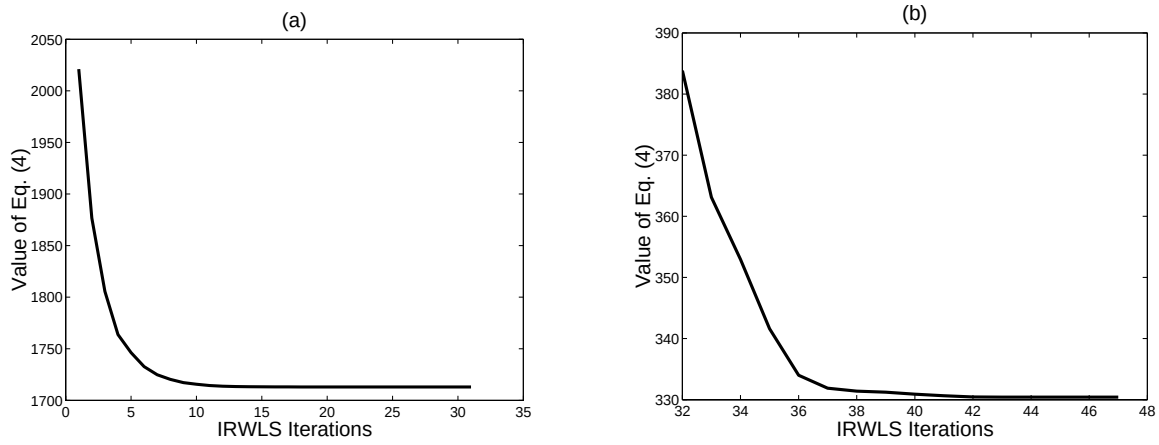

Figure 3: Two-step procedure for solving the SVC with non-convex loss function: (a) convergence of the IRWLS procedure to the SVC solution with loss function (7); (b) convergence of the IRWLS procedure to the SVC solution with loss function (31) with $\eta = 2$, using as starting point the solution at the final iteration in (a).