

98-725



Least Squares Support Vector Machine Classifiers

J.A.K. SUYKENS and J. VANDEWALLE

Katholieke Universiteit Leuven, Department of Electrical Engineering, ESAT-SISTA Kardinaal Mercierlaan 94, B-3001 Leuven (Heverlee), Belgium, e-mail: johan.suykens@esat.kuleuven.ac.be

Abstract. In this letter we discuss a least squares version for support vector machine (SVM) classifiers. Due to equality type constraints in the formulation, the solution follows from solving a set of linear equations, instead of quadratic programming for classical SVM's. The approach is illustrated on a two-spiral benchmark classification problem.

Key words: classification, support vector machines, linear least squares, radial basis function kernel

Abbreviations: SVM – Support Vector Machines; VC – Vapnik-Chervonenkis; RBF – Radial Basis Function

1. Introduction

Recently, support vector machines (Vapnik, 1995; Vapnik, 1998a; Vapnik, 1998b) have been introduced for solving pattern recognition problems. In this method one maps the data into a higher dimensional input space and one constructs an optimal separating hyperplane in this space. This basically involves solving a quadratic programming problem, while gradient based training methods for neural network architectures on the other hand suffer from the existence of many local minima (Bishop, 1995; Cherkassky & Mulier, 1998; Haykin, 1994; Zurada, 1992). Kernel functions and parameters are chosen such that a bound on the VC dimension is minimized. Later, the support vector method was extended for solving function estimation problems. For this purpose Vapnik's epsilon insensitive loss function and Huber's loss function have been employed. Besides the linear case, SVM's based on polynomials, splines, radial basis function networks and multilayer perceptrons have been successfully applied. Being based on the structural risk minimization principle and capacity concept with pure combinatorial definitions, the quality and complexity of the SVM solution does not depend directly on the dimensionality of the input space (Vapnik, 1995; Vapnik, 1998a; Vapnik, 1998b).

In this paper we formulate a least squares version of SVM's for classification problems with two classes. For the function estimation problem a support vector interpretation of ridge regression (Golub & Van Loan, 1989) has been given in (Saunders et al., 1998), which considers equality type constraints instead of inequalities from the classical SVM approach. Here, we also consider equality

constraints for the classification problem with a formulation in least squares sense. As a result the solution follows directly from solving a set of linear equations, instead of quadratic programming. While in classical SVM's many support values are zero (nonzero values correspond to support vectors), in least squares SVM's the support values are proportional to the errors.

This paper is organized as follows. In Section 2 we review some basic work about support vector machine classifiers. In Section 3 we discuss the least squares support vector machine classifiers. In Section 4 examples are given to illustrate the support values and on a two-spiral benchmark problem.

2. Support Vector Machines for Classification

In this Section we shortly review some basic work on support vector machines (SVM) for classification problems. For all further details we refer to (Vapnik, 1995; Vapnik, 1998a; Vapnik, 1998b).

Given a training set of N data points $\{y_k, x_k\}_{k=1}^N$, where $x_k \in \mathbb{R}^n$ is the k th input pattern and $y_k \in \mathbb{R}$ is the k th output pattern, the support vector method approach aims at constructing a classifier of the form:

$$y(x) = \text{sign} \left[\sum_{k=1}^N \alpha_k y_k \psi(x, x_k) + b \right], \quad (1)$$

where α_k are positive real constants and b is a real constant. For $\psi(\cdot, \cdot)$ one typically has the following choices: $\psi(x, x_k) = x_k^T x$ (linear SVM); $\psi(x, x_k) = (x_k^T x + 1)^d$ (polynomial SVM of degree d); $\psi(x, x_k) = \exp\{-\|x - x_k\|_2^2 / \sigma^2\}$ (RBF SVM); $\psi(x, x_k) = \tanh[\kappa x_k^T x + \theta]$ (two layer neural SVM), where σ , κ and θ are constants.

The classifier is constructed as follows. One assumes that

$$\begin{aligned} w^T \varphi(x_k) + b &\geq 1, & \text{if } y_k = +1, \\ w^T \varphi(x_k) + b &\leq -1, & \text{if } y_k = -1, \end{aligned} \quad (2)$$

which is equivalent to

$$y_k [w^T \varphi(x_k) + b] \geq 1, \quad k = 1, \dots, N, \quad (3)$$

where $\varphi(\cdot)$ is a nonlinear function which maps the input space into a higher dimensional space. However, this function is not explicitly constructed. In order to have the possibility to violate (3), in case a separating hyperplane in this high dimensional space does not exist, variables ξ_k are introduced such that

$$\begin{aligned} y_k [w^T \varphi(x_k) + b] &\geq 1 - \xi_k, & k = 1, \dots, N, \\ \xi_k &\geq 0, & k = 1, \dots, N. \end{aligned} \quad (4)$$

According to the structural risk minimization principle, the risk bound is minimized by formulating the optimization problem

$$\min_{w, \xi_k} \mathcal{J}_1(w, \xi_k) = \frac{1}{2} w^T w + c \sum_{k=1}^N \xi_k \quad (5)$$

subject to (4). Therefore, one constructs the Lagrangian

$$\mathcal{L}_1(w, b, \xi_k; \alpha_k, \nu_k) = \mathcal{J}_1(w, \xi_k) - \sum_{k=1}^N \alpha_k \{y_k [w^T \varphi(x_k) + b] - 1 + \xi_k\} - \sum_{k=1}^N \nu_k \xi_k \tag{6}$$

by introducing Lagrange multipliers $\alpha_k \geq 0, \nu_k \geq 0$ ($k = 1, \dots, N$). The solution is given by the saddle point of the Lagrangian by computing

$$\max_{\alpha_k, \nu_k} \min_{w, b, \xi_k} \mathcal{L}_1(w, b, \xi_k; \alpha_k, \nu_k). \tag{7}$$

One obtains

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial w} = 0 &\rightarrow w = \sum_{k=1}^N \alpha_k y_k \varphi(x_k), \\ \frac{\partial \mathcal{L}_1}{\partial b} = 0 &\rightarrow \sum_{k=1}^N \alpha_k y_k = 0, \\ \frac{\partial \mathcal{L}_1}{\partial \xi_k} = 0 &\rightarrow 0 \leq \alpha_k \leq c, \quad k = 1, \dots, N, \end{aligned} \tag{8}$$

which leads to the solution of the following quadratic programming problem

$$\max_{\alpha_k} \mathcal{Q}_1(\alpha_k; \varphi(x_k)) = -\frac{1}{2} \sum_{k,l=1}^N y_k y_l \varphi(x_k)^T \varphi(x_l) \alpha_k \alpha_l + \sum_{k=1}^N \alpha_k, \tag{9}$$

such that

$$\sum_{k=1}^N \alpha_k y_k = 0, \quad 0 \leq \alpha_k \leq c, \quad k = 1, \dots, N.$$

The function $\varphi(x_k)$ in (9) is related then to $\psi(x, x_k)$ by imposing

$$\varphi(x)^T \varphi(x_k) = \psi(x, x_k), \tag{10}$$

which is motivated by Mercer’s Theorem. Note that for the two layer neural SVM, Mercer’s condition only holds for certain parameter values of κ and θ .

The classifier (1) is designed by solving

$$\max_{\alpha_k} \mathcal{Q}_1(\alpha_k; \psi(x_k, x_l)) = -\frac{1}{2} \sum_{k,l=1}^N y_k y_l \psi(x_k, x_l) \alpha_k \alpha_l + \sum_{k=1}^N \alpha_k, \tag{11}$$

subject to the constraints in (9). One does not have to calculate w nor $\varphi(x_k)$ in order to determine the decision surface. Because the matrix associated with this quadratic programming problem is not indefinite, the solution to (11) will be global (Fletcher, 1987).

Furthermore, one can show that hyperplanes (3) satisfying the constraint $\|w\|_2 \leq a$ have a VC-dimension h which is bounded by

$$h \leq \min([r^2 a^2], n) + 1, \tag{12}$$

where $[.]$ denotes the integer part and r is the radius of the smallest ball containing the points $\varphi(x_1), \dots, \varphi(x_N)$. Finding this ball is done by defining the Lagrangian

$$\mathcal{L}_2(r, q, \lambda_k) = r^2 - \sum_{k=1}^N \lambda_k (r^2 - \|\varphi(x_k) - q\|_2^2), \tag{13}$$

where q is the center of the ball and λ_k are positive Lagrange multipliers. In a similar way as for (5) one finds that the center is equal to $q = \sum_k \lambda_k \varphi(x_k)$, where the Lagrange multipliers follow from

$$\max_{\lambda_k} \mathcal{Q}_2(\lambda_k; \varphi(x_k)) = - \sum_{k,l=1}^N \varphi(x_k)^T \varphi(x_l) \lambda_k \lambda_l + \sum_{k=1}^N \lambda_k \varphi(x_k)^T \varphi(x_k), \tag{14}$$

such that

$$\sum_{k=1}^N \lambda_k = 1, \quad \lambda_k \geq 0, \quad k = 1, \dots, N.$$

Based on (10), \mathcal{Q}_2 can also be expressed in terms of $\psi(x_k, x_l)$. Finally, one selects a support vector machine with minimal VC dimension by solving (11) and computing (12) from (14).

3. Least Squares Support Vector Machines

Here we introduce a least squares version to the SVM classifier by formulating the classification problem as

$$\min_{w,b,e} \mathcal{J}_3(w, b, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2, \tag{15}$$

subject to the equality constraints

$$y_k [w^T \varphi(x_k) + b] = 1 - e_k, \quad k = 1, \dots, N. \tag{16}$$

One defines the Lagrangian

$$\mathcal{L}_3(w, b, e; \alpha) = \mathcal{J}_3(w, b, e) - \sum_{k=1}^N \alpha_k \{y_k [w^T \varphi(x_k) + b] - 1 + e_k\}, \tag{17}$$

where α_k are Lagrange multipliers (which can be either positive or negative now due to the equality constraints as follows from the Kuhn-Tucker conditions (Fletcher, 1987)).

The conditions for optimality

$$\begin{aligned} \frac{\partial \mathcal{L}_3}{\partial w} = 0 &\rightarrow w = \sum_{k=1}^N \alpha_k y_k \varphi(x_k), \\ \frac{\partial \mathcal{L}_3}{\partial b} = 0 &\rightarrow \sum_{k=1}^N \alpha_k y_k = 0, \\ \frac{\partial \mathcal{L}_3}{\partial e_k} = 0 &\rightarrow \alpha_k = \gamma e_k, \quad k = 1, \dots, N, \\ \frac{\partial \mathcal{L}_3}{\partial \alpha_k} = 0 &\rightarrow y_k [w^T \varphi(x_k) + b] - 1 + e_k = 0, \quad k = 1, \dots, N \end{aligned} \tag{18}$$

can be written immediately as the solution to the following set of linear equations (Fletcher, 1987)

$$\left[\begin{array}{ccc|c} I & 0 & 0 & -Z^T \\ 0 & 0 & 0 & -Y^T \\ 0 & 0 & \gamma I & -I \\ \hline Z & Y & I & 0 \end{array} \right] \begin{bmatrix} w \\ b \\ e \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \bar{1} \end{bmatrix}, \quad (19)$$

where $Z = [\varphi(x_1)^T y_1; \dots; \varphi(x_N)^T y_N]$, $Y = [y_1; \dots; y_N]$, $\bar{1} = [1; \dots; 1]$, $e = [e_1; \dots; e_N]$, $\alpha = [\alpha_1; \dots; \alpha_N]$. The solution is also given by

$$\left[\begin{array}{c|c} 0 & -Y^T \\ \hline Y & ZZ^T + \gamma^{-1}I \end{array} \right] \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{1} \end{bmatrix}. \quad (20)$$

Mercer's condition can be applied again to the matrix $\Omega = ZZ^T$, where

$$\begin{aligned} \Omega_{kl} &= y_k y_l \varphi(x_k)^T \varphi(x_l) \\ &= y_k y_l \psi(x_k, x_l). \end{aligned} \quad (21)$$

Hence, the classifier (1) is found by solving the linear set of Equations (20)–(21) instead of quadratic programming. The parameters of the kernels such as σ for the RBF kernel can be optimally chosen according to (12). The support values α_k are proportional to the errors at the data points (18), while in the case of (14) most values are equal to zero. Hence, one could rather speak of a support value spectrum in the least squares case.

4. Examples

In a first example (Figure 1) we illustrate the support values for a linearly separable problem of two classes in a two dimensional space. The size of the circles indicated at the training data is chosen proportionally to the absolute values of the support values. A linear SVM has been taken with $\gamma = 1$. Clearly, points located close and far from the decision line have the largest support values. This is different from SVM's based on inequality constraints, where only points that are near the decision line have nonzero support values. This can be understood from the fact that the signed distance from a point x_k to the decision line is equal to $(w^T x_k + b)/\|w\| = (1 - e_k)/(y_k \|w\|)$ and $\alpha_k = \gamma e_k$ in the least squares SVM case.

In a second example (Figure 2) we illustrate a least squares support vector machine RBF classifier on a two-spiral benchmark problem. The training data are shown on Figure 2 with two classes indicated by 'o' and '*' (360 points with 180 for each class) in a two dimensional input space. Points in between the training data located on the two spirals are often considered as test data for this problem but are not shown on the figure. The excellent generalization performance is clear from the decision boundaries shown on the figures. In this case $\sigma = 1$ and $\gamma = 1$ were chosen as parameters. Other methods which have been applied to the two-spiral

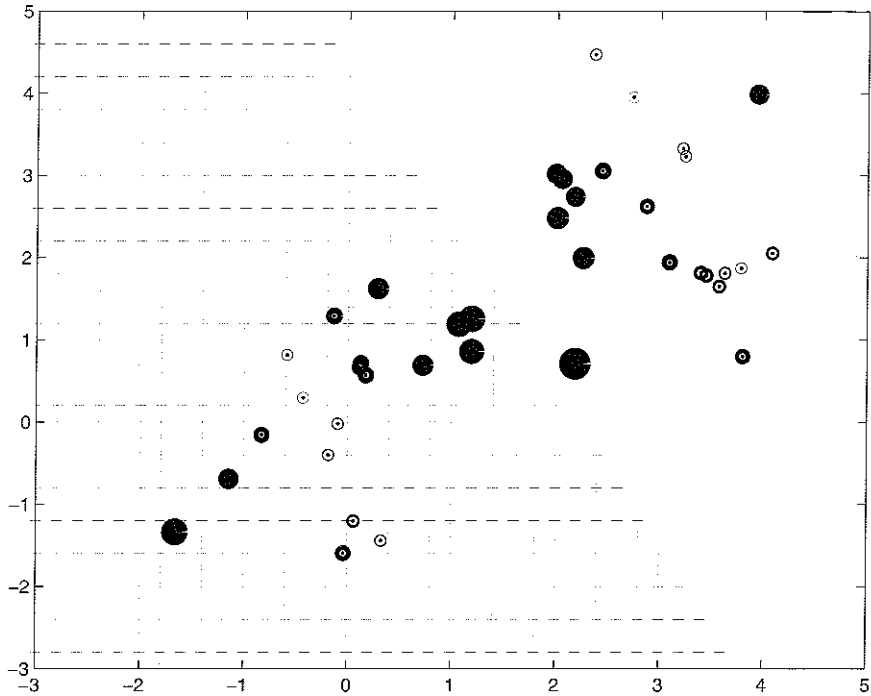


Figure 1. Example of two linearly separable classes in a two-dimensional input space. The size of the circles indicated at the training data is chosen proportionally to the absolute value of the support value.

benchmark problem, such as the use of circular units (Ridella et al., 1997), have shown good performance as well. The least squares SVM solution on the other hand can be found with low computational cost and is free of many local minima, being the solution to a convex optimization problem. For two-spiral classification problems the method gives good results over a wide parameter range of σ and γ values.

5. Conclusions

We discussed a least squares version of support vector machine classifiers. Due to the equality constraints in the formulation, a set of linear equations has to be solved instead of a quadratic programming problem. Mercer's condition is applied as in other SVM's. For a complicated two-spiral classification problem it is illustrated that a least squares SVM with RBF kernel is readily found with excellent generalization performance and low computational cost.