J Glob Optim (2008) 41:15–30
DOI 10.1007/s10898-007-9162-0

Quadratic kernel-free non-linear support vector machine

Issam Dagher

Received: 20 October 2006 / Accepted: 5 April 2007 / Published online: 13 June 2007

Abstract A new quadratic kernel-free non-linear support vector machine (which is called QSVM) is introduced. The SVM optimization problem can be stated as follows: Maximize the geometrical margin subject to all the training data with a functional margin greater than a constant. The functional margin is equal to $W^T X + b$ which is the equation of the hyper-plane used for linear separation. The geometrical margin is equal to $\frac{1}{||W||}$. And the constant in this case is equal to one. To separate the data non-linearly, a dual optimization form and the Kernel trick must be used. In this paper, a quadratic decision function that is capable of separating non-linearly the data is used. The geometrical margin is proved to be equal to the inverse of the norm of the gradient of the decision function. The functional margin is the equation of the quadratic function. QSVM is proved to be put in a quadratic optimization setting. This setting does not require the use of a dual form or the use of the Kernel trick. Comparisons between the QSVM and the SVM using the Gaussian and the polynomial kernels on databases from the UCI repository are shown.

Keywords Support Vector Machine (SVM) · Geometrical margin · Functional margin · QSVM · Quadratic function · Dual optimization form · Kernel trick.

1 Introduction

Support Vector Machine (SVM) was first proposed by Vapnik [13]. It is based on finding an optimal hyper-plane which separates the data into two classes with the largest margin. Some applications of the SVM are: Histogram-based Image Classification [2], Spam Categorization [6], Financial Time Series Forecasting [1], Face Membership Authentication [9] and data analysis and classification [4,8].

I. Dagher (✉)
Department of Computer Engineering, University of Balamand,
P.O. Box 100, Tripoli, Elkoura, Lebanon
e-mail: dagheri@balamand.edu.lb

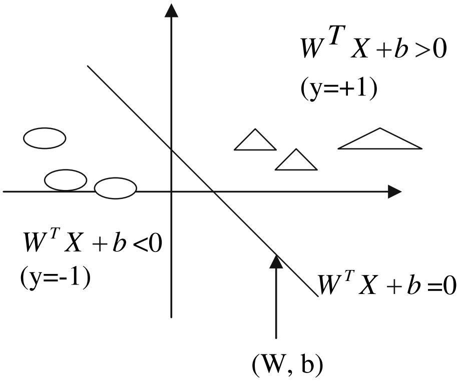


Fig. 1 Linear separation of the data

Given the n training data pairs (X_i, y_i). Where X_i is an m-dimensional vector and yi $\in \{-1, +1\}$ is its target. It is assumed that all the data of Class 1 (Class 2) are labeled by +1 (-1), respectively. The data can be put in matrix form as follows:

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nm} \end{bmatrix}; \; y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \tag{1}$$

A hyper-plane (W, b) [6] that is capable of separating the data into two classes is given by:

$$\begin{aligned} &W^T X + b = 0, \\ &\text{where } W = [w_1 \quad w_2 \quad \cdots \quad w_m]^T \text{ and } b \text{ is a scalar.} \end{aligned} \tag{2}$$

The training objective is to find the $m + 1$ unknowns (W, b) which will linearly separate the data. Any data X_i can be labeled as follows:

$$y_i = \begin{cases} +1 & \text{if } W^T X_i + b \geq 0 \\ -1 & \text{if } W^T X_i + b < 0 \end{cases} \Leftrightarrow y_i(W^T X_i + b) \geq 1 \Leftrightarrow \overset{\Lambda}{\gamma}(i) \geq 1 \tag{3}$$

$\overset{\Lambda}{\gamma}(i)$ is called the functional margin. Bigger values of $\overset{\Lambda}{\gamma}(i)$ implies more confident classification. For example, the triangle biggest in size in Fig. 1 has the biggest value of the functional margin for Class1.

It should be noted that the functional margin of a data is not equal to the Euclidian distance of that data to the hyper-plane. This distance is given by the geometrical margin γ (Fig. 2).

The geometrical margin can be derived as follows:

$$\begin{aligned} &\overset{\rightarrow}{OX} = \overset{\rightarrow}{OX_i} - \overset{\rightarrow}{X_i X} \\ &X = X_i - \gamma(i)\frac{W}{||W||} \text{ and } W^T X + b = 0 \Rightarrow \gamma(i) = \frac{W^T X_i + b}{||W||} = \frac{\overset{\Lambda}{\gamma}(i)}{||W||} \end{aligned} \tag{4}$$

The SVM optimization problem can be stated as follows: Maximize the geometrical margin subject to all the training data have a functional margin greater than a constant $\overset{\Lambda}{\gamma}$.

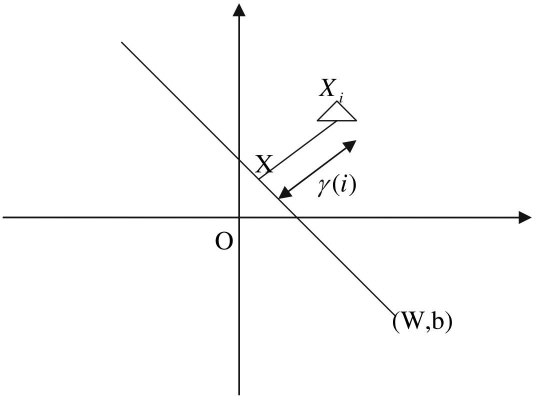


Fig. 2 Illustration of the geometrical margin

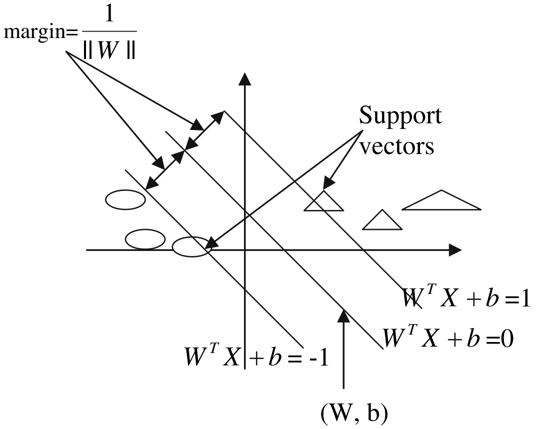


Fig. 3 SVM optimization output

Let $\overset{\Lambda}{\gamma} = 1$, the optimization problem can be written as:

$$\begin{aligned} &\text{Maximize} \quad \gamma = \frac{1}{\|W\|} \\ &\text{subject} \quad y_i(W^T X_i + b) \geq 1; \quad i = 1, \ldots, n. \end{aligned} \tag{5}$$

Figure 3 shows the output of the SVM optimization problem: the margin, the support vectors and the equations of the two lines passing by the support vectors.

The SVM optimization problem can be stated as follows:

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\|W\|^2 = \frac{1}{2} W^T W \\ &\text{subject} \quad -\mathrm{y_i}(W^T X_i + b) \leq -1; \quad i = 1, \ldots, n. \end{aligned} \tag{6}$$

Now we convert the above formulation into the following quadratic optimization problem:

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2} z^T H z + f^T z \\ &\text{subject} \quad \mathrm{Az} \leq \mathrm{c}, \end{aligned} \tag{7}$$

where:

$$z = [\mathrm{W}b]^T;\ H = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix};\ f = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$A = -\begin{bmatrix} y_1 & 0 & 0 \cdots & 0 \\ 0 & y_2 & 0 \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 \cdots & y_n \end{bmatrix} \begin{bmatrix} X_1^T & 1 \\ X_2^T & 1 \\ \vdots & \vdots \\ X_n^T & 1 \end{bmatrix};\ c = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}. \tag{8}$$

The dual representation of the SVM optimization problem is given by using the following Lagrangian function:

$$\begin{aligned} & L(W, b) = \frac{1}{2}||W||^2 - \sum_{i=1}^{n} \alpha(i)[y_i(W^T X_i + b) - 1] \text{ with } \alpha(\mathrm{i}) \geq 0, \\ & \frac{\partial L}{\partial w} = 0 \Rightarrow W = \sum_{i=1}^{n} \alpha(i) y_i X_i, \\ & \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha(i) y_i = 0, \\ & L(\alpha) = \sum_{i=1}^{n} \alpha(i) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha(i)\alpha(j) y_i y_j X_i^T X_j. \end{aligned} \tag{9}$$

The dual problem is stated as:

$$\text{Maximize} \quad L(\alpha) \Leftrightarrow \text{Minimize } \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha(i)\alpha(j) y_i y_j X_i^T X_j - \sum_{i=1}^{n} \alpha(i) \tag{10}$$

$$\text{subject} \quad \sum_{\mathrm{i}=1}^{\mathrm{n}} \alpha(\mathrm{i}) y_{\mathrm{i}} = 0 \text{ and } \alpha(\mathrm{i}) \geq 0.$$

This dual problem can be put in a form identical to the following quadratic optimization problem:

$$\begin{aligned} & \text{Minimize} \quad \frac{1}{2} z^T H z + f^T z \\ & \text{subject} \quad \mathrm{Az} \leq \mathrm{c} \text{ and } Bz = d \text{ and } z_l \leq z \leq z_u, \\ & \text{where} \\ & z = \begin{bmatrix} \alpha(1) \ \alpha(2) \cdots \alpha(n) \end{bmatrix}^T; \quad f = \begin{bmatrix} -1 \ -1 \cdots -1 \end{bmatrix}; \quad A = \begin{bmatrix} 0 \cdots 0 \end{bmatrix}; \quad c = 0; \\ & B = \begin{bmatrix} y_1 \ y_2 \cdots y_n \end{bmatrix}; \quad d = 0; \quad z_l = \begin{bmatrix} 0 \cdots 0 \end{bmatrix}; \quad z_u = \begin{bmatrix} \infty \cdots \infty \end{bmatrix} \\ & H = \begin{bmatrix} y_1 y_1 X_1^T X_1 & y_1 y_2 X_1^T X_2 & \cdots & y_1 y_n X_1^T X_n \\ y_2 y_1 X_2^T X_1 & y_2 y_2 X_2^T X_2 & \cdots & y_2 y_n X_2^T X_n \\ \vdots & \vdots & & \vdots \\ y_n y_1 X_n^T X_1 & y_n y_2 X_n^T X_2 & \cdots & y_n y_n X_n^T X_n \end{bmatrix}. \end{aligned} \tag{11}$$

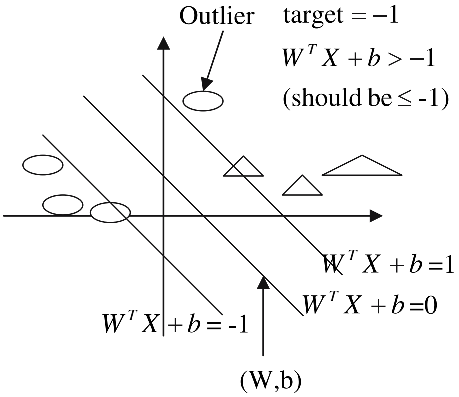


Fig. 4 Presence of an outlier in the data

Figure 4 show that the presence of some outliers will make the data non-linearly separable [10].

There are two kinds of misclassification errors:

$$\begin{aligned}&\text{target} = -1 \text{ and } W^T X + b > -1 \Rightarrow E_1 = W^T X + b + 1\\ &and\\ &\text{target} = 1 \text{ and } W^T X + b < 1 \Rightarrow E_2 = 1 - (W^T X + b).\end{aligned} \tag{12}$$

Minimizing the errors can be done by introducing two slack variables as follows:

$$\begin{aligned}&W^T X + b \le -1 + \xi_1 \text{ and minimize } \xi_1\\ &and\\ &W^T X + b \ge 1 - \xi_2 \text{ and minimize } \xi_2.\end{aligned} \tag{13}$$

The solution for misclassification errors can be addressed by the following optimization problem called soft SVM.

$$\begin{aligned}&\text{Minimize} \quad \frac{1}{2}||W||^2 + C\sum_{i=1}^{n} \zeta_i\\ &\text{subject} \quad y_{\rm i}(W^T X_i + b) \le 1 - \zeta_i; \quad i = 1, \ldots, n;\ \zeta_{\rm i} \ge 0\\ &C: \ \text{Weighting constant.}\end{aligned} \tag{14}$$

The dual SVM problem is given by:

$$\begin{aligned}&\text{Minimize } L(\alpha) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha(i)\alpha(j) y_i y_j X_i^T X_j - \sum_{i=1}^{n} \alpha(i)\\ &\text{subject} \quad \sum_{\rm i=1}^{\rm n} \alpha({\rm i}) y_{\rm i} = 0 \text{ and } 0 \le \alpha({\rm i}) \le C.\end{aligned} \tag{15}$$

This dual problem can be put as a quadratic optimization similar to (12) with the exception of $z_u = [C \cdots C]$ instead of a vector with infinite (or very big values) components.

For non-linearly separable data X, a transformation $\phi(X)$ called feature space is used [6,11].

This transformation is illustrated below:

X	$\phi(X)$
m-dimensions	p-dimensions (p>m)
Non-linearly separable data	Linearly separable data

In the feature space, the SVM solution is to find (W, b) such that:

$$\begin{aligned} W^T\phi(X) + b &\geq 1 \Rightarrow Class1 \\ W^T\phi(X) + b &\leq -1 \Rightarrow Class2. \end{aligned} \tag{16}$$

The optimization problem for the non-linearly separable data is similar to the one for the linearly separable data (dual forms) except that every dot product is replaced by a non linear kernel function K (kernel trick).

$$\phi^T(X1).\phi(X2) = K(X1, X2). \tag{17}$$

For example, consider the following problem applied to the linearly separable data:

$$\text{Minimize} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha(i)\alpha(j)y_i y_j X_i^T X_j - \sum_{i=1}^{n}\alpha(i) \tag{18}$$

$$\text{subject} \quad \sum_{\mathrm{i=1}}^{\mathrm{n}}\alpha(\mathrm{i})y_{\mathrm{i}} = 0 \text{ and } \alpha(\mathrm{i}) \geq 0.$$

For the non-linearly separable data and using the kernel trick the above problem is replaced by:

$$\text{Minimize} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha(i)\alpha(j)y_i y_j K(X_i, X_j) - \sum_{i=1}^{n}\alpha(i) \tag{19}$$

$$\text{subject} \quad \sum_{\mathrm{i=1}}^{\mathrm{n}}\alpha(\mathrm{i})\mathrm{y}_{\mathrm{i}} = 0 \text{ and } \alpha(\mathrm{i}) \geq 0.$$

There is usually no automatic way to choose a kernel. Some examples of a Kernel are:

$$\begin{aligned} &\text{Gaussian: } K(X1, X2) = \exp\left(-\frac{(X_1 - X_2)^2}{2\sigma^2}\right) \\ &\text{Polynomial: } K(X1, X2) = \left(1 + X_1^T X_2\right)^p \end{aligned} \tag{20}$$

2 Quadratic kernel-free SVM (QSVM)

A quadratic function (W, b, c) that is capable of separating non-linearly the data into two classes is given by:

$$f(X) = \frac{1}{2}X^T W X + b^T X + c$$

$$W = W^T = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{12} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & & \vdots \\ w_{1m} & w_{2m} & \cdots & w_{mm} \end{bmatrix}; \ b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}. \tag{21}$$

The following two important comments should be stated:

1. The decision surfaces $f(X) = ct$ can assume any of the general forms of hyper-planes, hyper-spheres, hyper-ellipsoids, hyper-paraboloids, hyper-hyperboloids of various types [7].
2. $f(X)$can be considered as the sum of two terms: the non-linear term ($f_{\text{non-linear}}(X) = \frac{1}{2}X^T WX$) and the linear term ($f_{\text{linear}}(X) = b^T X + c$).

Using Fig. 5, the derivation of the QSVM optimization problem is done as follows:

$$\overrightarrow{OX_B} = \overrightarrow{OX_i} - \overrightarrow{X_i X_B} \Rightarrow X_B = X_i - \gamma(i)\frac{\Delta f(X_B)}{||\Delta f(X_B)||}. \tag{22}$$

Applying the first order Taylor Series expansion [3]:

$$f(X_i) \approx f(X_B) + \Delta^T f(X_B)(X_i - X_B) \tag{23}$$

X_B is on the decision function:

$$f(X_B) = 0. \tag{24}$$

And the functional margin is given by:

$$f(X_i) = \overset{\wedge}{\gamma}(i). \tag{25}$$

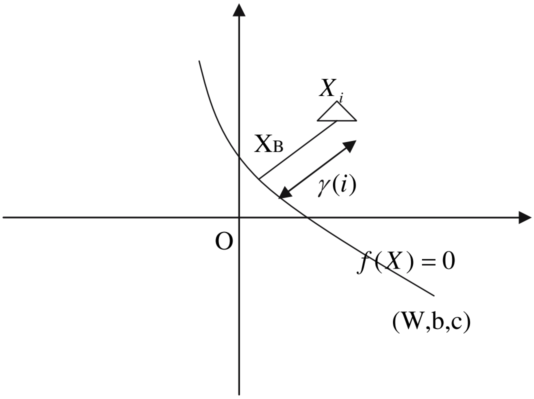
Equations (23–25) will give:

$$\overset{\wedge}{\gamma}(i) \approx \gamma(i)||\Delta f(X_B)|| \Rightarrow \gamma(i) \approx \frac{\overset{\wedge}{\gamma}(i)}{||\Delta f(X_B)||}. \tag{26}$$

It should be noted that this equation is valid for the hyper-plane decision function:

$$\Delta f(X) = W \Rightarrow \gamma(i) = \frac{\overset{\wedge}{\gamma}(i)}{||W||} \tag{27}$$

Fig. 5 Quadratic decision surface



The QSVM optimization problem is given by:

$$\begin{aligned} &\text{Maximize} \quad \gamma(i) \approx \frac{\overset{\wedge}{\gamma}}{||\Delta f(X_B)||} \\ &\text{subject to:} \quad \gamma(i) \geq \overset{\wedge}{\gamma} \end{aligned} \tag{28}$$

Letting $\overset{\wedge}{\gamma} = 1$, the QSVM optimization problem can be restated as follows:

$$\begin{gathered} \text{Minimize} \quad ||\Delta f(X)|| \Leftrightarrow \text{Minimize } ||\Delta f(X)||^2 \\ \text{subject to: } \gamma(i) \geq 1. \end{gathered} \tag{29}$$

For n training data, the QSVM optimization problem becomes:

$$\begin{aligned} &\text{Minimize} \quad \sum_{i=1}^{\mathrm{n}} ||\Delta f(X_i)||^2 \\ &\text{Subject to:} \quad \gamma(\mathrm{i}) \geq 1 \text{ for } i = 1, \mathrm{K}, n. \end{aligned} \tag{30}$$

The gradient of the quadratic objective function is given by:

$$\Delta f(X) = WX + b \tag{31}$$

The norm of the gradient is given by:

$$||\Delta f(X)||^2 = (WX + b)^T (WX + b) = X^T W^T WX + b^T b + 2b^T W. \tag{32}$$

In this paper, the following approximation is done:

$$\text{Minimize } ||\Delta f(X)||^2 \approx \text{Minimize} X^T W^T WX + b^T b. \tag{33}$$

This approximation can be justified as follows:

$$\begin{aligned} &||\Delta f(X)||^2 \geq 0 \Rightarrow X^T W^T WX + b^T b \geq -2b^T W \\ &\Rightarrow \text{Minimize } X^T W^T WX + b^T b \Leftrightarrow \text{Maximize } -2b^T W \Leftrightarrow \text{Minimize } 2b^T W \Rightarrow \\ &\approx \text{Minimize} ||\Delta f(X)||^2. \end{aligned}$$

This approximation made it easy for the QSVM optimization problem to be put as a quadratic optimization problem. It contains two terms: One term related to the linear part of the quadratic function and another term related to the non-linear term.

For the linear part:

$$\Delta f_{\text{linear}}(X) = b \Rightarrow ||\Delta f(X)||^2 = b^T b. \tag{34}$$

The quadratic form of the linear part is:

$$||\Delta f_{\text{linear}}(X)||^2 = \frac{1}{2} b^T H_{\text{linear}} b, \tag{35}$$

$$\text{where } \mathrm{z}_{\text{linear}} = \mathrm{b};\ H_{\text{linear}} = \begin{bmatrix} 2 & 0 & \cdots & 0 \\ 0 & 2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 2 \end{bmatrix}.$$

The constraints for the linear part are given by:

$$-y_{\mathrm{i}}(b^T X_i + c) \leq -1; \quad i = 1, \ldots, n.$$

For the non-linear part:

$$\Delta f_{\mathrm{non-linear}}(X) = WX \Rightarrow ||\Delta f_{\mathrm{non-linear}}(X)||^2 = (WX)^T (WX). \tag{36}$$

Equation (36) can be put in a quadratic form as follows:

$$||\Delta f_{\mathrm{non-linear}}(X)||^2 = \frac{1}{2} W_1^T M_1^T M_1 W_1 \tag{37}$$
$$z_{\mathrm{non-linear}} = W_1; \;\; H_{\mathrm{non-linear}} = M_1^T M_1,$$

where W_1 is the vector formed by taking the $\frac{m^2+m}{2}$parameters of the upper triangle of the matrix W (eq. 38).

$$W_1 = [\mathrm{uppertriangle}(W)] = [w_{11} \quad \cdots \quad w_{1m} \quad w_{22} \quad \cdots \quad w_{2m} \quad \cdots \quad w_{mm}]. \tag{38}$$

And M_1 is a $\left(\frac{m^2+m}{2}\right) \times m$ matrix formed as follows:

Assuming that the data X is given by $\left[x_1 \; x_2 \; \cdots \; x_m \right]^T$ then in each ith column of M_1, find the positions of all the components of W_1 which have the form w_{id} or w_{di} (where d can be any value) and set those positions in the *ith* column of M_1 to $\left[x_1 \; x_2 \; \cdots \; x_m \right]$and set the other positions to zero.

This can be illustrated for $m = 3$:

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{12} & w_{22} & w_{23} \\ w_{13} & w_{23} & w_{33} \end{bmatrix} \Rightarrow W_1 = \left[w_{11} \; w_{12} \; w_{13} \; w_{22} \; w_{23} \; w_{33} \right]$$

$$M_1 = \begin{bmatrix} x_1 & 0 & 0 \\ x_2 & x_1 & 0 \\ x_3 & 0 & x_1 \\ 0 & x_2 & 0 \\ 0 & x_3 & x_2 \\ 0 & 0 & x_3 \end{bmatrix}.$$

The constraint equations for the non-linear part can be obtained by looking at the components of the W_1 vector and replacing every w_{ij} by $\begin{cases} \mathrm{x_i}.x_j & i \neq j \\ \frac{1}{2} x_i.x_i & i = j \end{cases}$

The constraint equations (n equations corresponding to n data) of the previous example are (assuming $m = 3$):

$$-y_i \left[\tfrac{1}{2} x_1.x_1 \;\; x_1.x_2 \;\; x_1.x_3 \;\; \tfrac{1}{2} x_2.x_2 \;\; x_2.x_3 \;\; \tfrac{1}{2} x_3.x_3 \right]_i \leq -1; \;\; i = 1, \cdots, n.$$

The sum of the two parts (the QSVM optimization problem) can be put as a one quadratic optimization problem as follows:

$$\begin{array}{ll} \text{Minimize } \; ||\Delta f(X)||^2 \Leftrightarrow & \text{Minimize } \frac{1}{2} z^T H z \\ \text{subject to: } \gamma(i) \geq 1 & \text{subject } to \;\, Az \leq k, \end{array}$$

$$\text{where } H = \begin{bmatrix} H_{\text{non-linear}} & \mathbf{0} & 0 \\ \mathbf{0} & H_{\text{linear}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } z = \begin{bmatrix} W_1 & b & c \end{bmatrix}^T$$

$$-y_i \begin{bmatrix} \frac{1}{2}x_1.x_1 & \cdots & x_1.x_m & \frac{1}{2}x_2.x_2 & \cdots & x_2.x_m & \cdots & \frac{1}{2}x_m.x_m & \boxed{x_1 \quad x_2 \quad \cdots \quad x_m} & 1 \end{bmatrix}_i \le -1 \tag{39}$$

$$i = 1, \cdots, n$$

3 $2-d$ Simulations

To illustrate the non-linear and the linear capabilities of the QSVM, three different simulations with different $2-d$ data sets were done. The first data set is given below. It contains ten data with their corresponding target y.

X = [1 1;1 2;3 0.5; 4 2;5 3; 2 2; 2 3;3 3;2.5 2.5;3 3.5];
y = [1 1 1 1 1 –1 –1 –1 –1 –1]′;

Applying QSVM, the following results were obtained. Figure 6 shows the data and the decision surfaces (contour plots) $f(X) = 0$, $f(X) = 1$, and $f(X) = -1$. The support vectors are on the last two contour plots. All the data with $f(X) \geq 1(\leq -1)$ belong to Class1 (Class 2). And Fig. 7 shows the $3-d$ plot of the quadratic objective function.

The second data set is the following:
X = [0 0;1 0;0 1;–1 0;0 –1;2 0;0 2;–2 0;0 –2;2 2];
y = [1 1 1 1 1 –1 –1 –1 –1 –1]′;
The following results (Figs. 8, 9) were obtained:
The third data set (linear set) is:
X = [–1.5 –5;–2 –4;–3 –3;–4 –4;–5 –5;1 3;2 2;3 2;4 2;5 3];
y = [1 1 1 1 1 –1 –1 –1 –1 –1]′;
And the following results (Figs. 10, 11) were obtained:

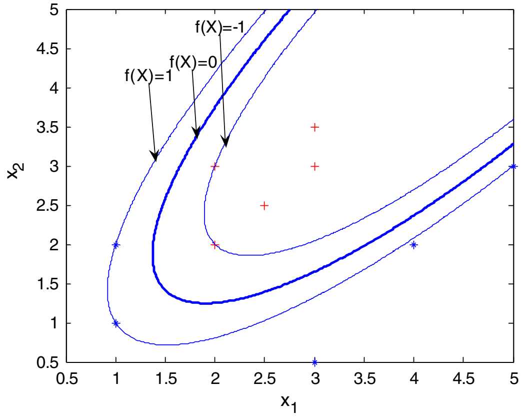


Fig. 6 Decision surface for the first $2-d$ data set

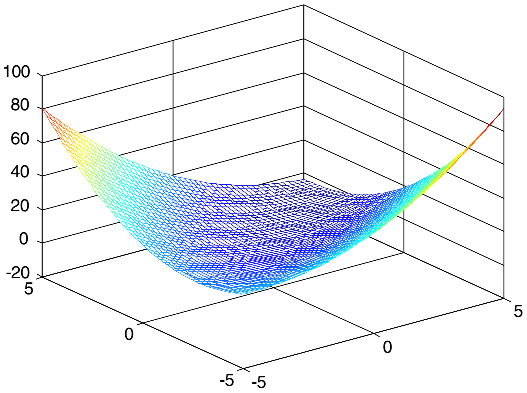


Fig. 7 $3-d$ Plot of the quadratic function for the first $2-d$ data set

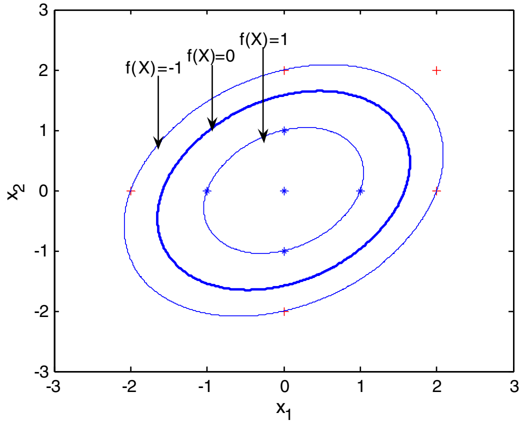


Fig. 8 Decision surface for the second $2-d$ data set

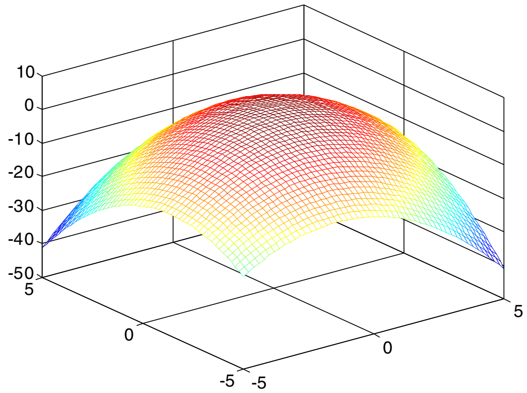


Fig. 9 $3-d$ Plot of the quadratic function for the second $2-d$ data set

4 Simulations on databases from the UCI repository

The databases, obtained from the UCI repository [12], used in the simulations are:

a. **Iris** database: 4 features, 3 classes, and 150 data.
b. **Balance** database: 4 features, 3 classes, and 625 data.

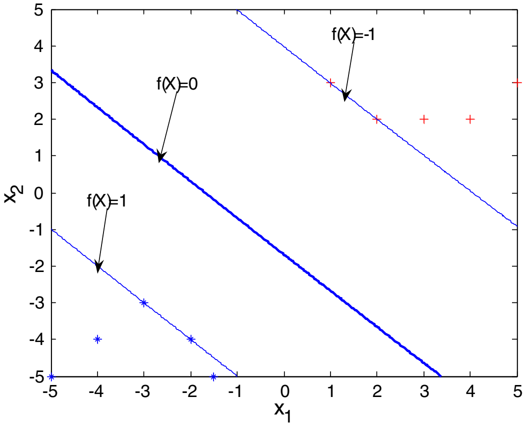


Fig. 10 Decision surface for the third $2 - d$ data set

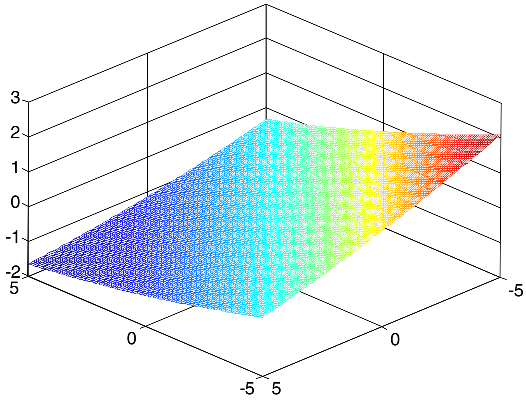


Fig. 11 $3 - d$ Plot of the quadratic function for the third $2 - d$ data set

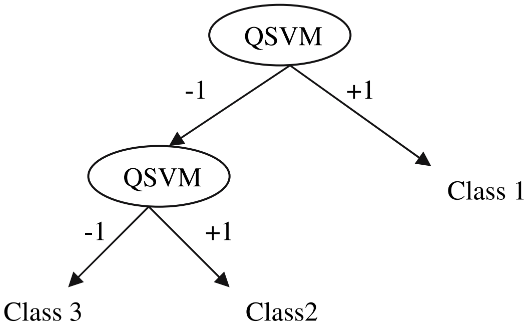


Fig. 12 Illustration of the use of QSVM for more than two classes

c. **Breast-Cancer** database: 9 features, 2 classes, 683 data.
d. **Glass** database: 9 features, 6 classes, 214 data.
e. **Wine** database: 13 features, 3 classes, 178 data.

For more than two classes, Fig. 12 illustrates the use of the QSVM. For example the Iris database is divided first into two sets. The first set contains all the data labeled Class 1 ($y = +1$) and the second set contains the remaining mixed labeled (Class 2 and Class 3) data ($y = -1$). QSVM is applied to these two sets and the first set of parameters (W_1, b_1, c_1)

Table 1 Results of the QSVM and the Kernel SVM for the **Iris** database

Iris	%train		Mean	Var	Min	Max
	20%	QSVM	96.67	4.16	93.33	98.33
		PSVM p=2	95.83	8.68	90.83	98.33
		SVM $\sigma = 0.5$	95.00	7.98	91.67	98.33
	40%	QSVM	98.22	0.37	97.77	98.88
		PSVM p=2	96.22	12.71	90.00	98.88
		SVM $\sigma = 1$	98.22	0.98	97.77	100
	60%	QSVM	100	0	100	100
		PSVM p=2	98.22	2.22	96.66	100
		SVM $\sigma = 0.2$	99.00	2.22	96.67	100
	LOO	QSVM	96	80	80	100
		PSVM p=2	94	10	80	100
		SVM $\sigma = 0.1$	94	100	80	100

corresponding to the first quadratic function $f_1(X)$ is obtained. Then the second set is divided into two subsets: All the data labeled Class 2 ($y = +1$) and the data labeled Class 3 ($y = -1$). QSVM is applied and the second set of parameters (W_2, b_2, c_2) corresponding to $f_2(X)$ is obtained.

In the test phase and for a given unlabeled data X:

$$\text{if } f_1(X) \geq 1 \Rightarrow \text{ Class1}$$

$$\text{else if } f_2(X) \geq 1 \Rightarrow Class2$$

$$\text{else Class3}$$

QSVM is compared to the SVM using the Gaussian kernel with different standard of deviations σ and different percentages of training including the five fold Leave-one-out (LOO) strategy. The values of σ are varied from 0.1 to1 with a step size of 0.1. It is also compared to the SVM using the polynomial kernel with $p = 1$ and 2 (PSVM). The tables show the values of the parameters which gave the best performances. The Matlab quadratic optimization function *quadprog*() is used with the default maximum number of iterations for both algorithms. Five different orders of pattern presentations are used. The mean, variance, minimum, maximum generalization performances on the remaining test sets are shown in the following tables (Tables 1, 2, 3, 4, 5).

From the results obtained, the following comments can be made:

- QSVM gave better performance than the SVM with polynomial kernel (PSVM).
- The best performances of the SVM with Gaussian kernel depend on the value of σ and on the percentage of training data. The performance of the QSVM does not depend on any tuning parameter.
- Irrespective of the σ used, QSVM gave better performance for the **Breast-Cancer** database, the **Glass** database and the **Balance** database using the 40% training data.

Table 2 Results of the QSVM and the Kernel SVM for the **Wine** database

Wine	%train		Mean	Var	Min	Max
	20%	QSVM	96.38	2.26	94.44	97.91
		PSVM p=2	95.83	2.41	93.75	97.91
		SVM $\sigma = 1$	96.66	1.06	95.13	97.91
	40%	QSVM	98.22	0.37	97.77	98.88
		PSVM p=2	98.14	1.71	97.22	100
		SVM $\sigma = 1$	98.22	0.98	97.77	100
	60%	QSVM	100	0	100	100
		PSVM p=2	100	0	100	100
		SVM $\sigma = 1$	100	0	100	100
	LOO	QSVM	98	80	80	100
		PSVM p=2	96	120	80	100
		SVM $\sigma = 0.3$	98	80	80	100

Table 3 Results of the QSVM and the Kernel SVM for the **Glass** database

Glass	%train		Mean	Var	Min	Max
	20%	QSVM	66.20	11.46	62.06	69.54
		PSVM p=2	64.82	18.39	59.77	68.96
		SVM $\sigma = 1$	65.63	5.68	62.64	68.96
	40%	QSVM	87.48	2.21	85.49	89.31
		PSVM p=2	72.21	77.09	62.59	83.96
		SVM $\sigma = 0.5$	85.64	3.32	83.96	88.54
	60%	QSVM	100	0	100	100
		PSVM p=2	80.22	39.77	71.59	88.63
		SVM $\sigma = 0.1$	100	0	100	100
	LOO	QSVM	60	400	40	80
		PSVM p=2	54	680	20	80
		SVM $\sigma = 0.1$	58	480	40	80

5 Conclusions

In this paper, a new quadratic kernel-free non-linear support vector machine (which is called QSVM) is introduced. Geometrical and functional margins are derived. QSVM was put as a quadratic optimization problem. It does not require the use of a dual optimization form or the use of the Kernel trick. Simulations on $2 - d$ data sets show that QSVM can separate linearly and non-linearly the data. The decision surfaces can assume any of the general forms of hyper-planes, hyper-spheres, hyper-ellipsoids, hyper-paraboloids, hyper-hyperboloids of various types. Simulations on the UCI repository databases show the good performance capabilities of this algorithm compared to the SVM using the Gaussian and the polynomial kernels without the need to tune any parameter.

Table 4 Results of the QSVM and the Kernel SVM for the **Balance** database

Balance	%train		Mean	Var	Min	Max
	20%	QSVM	94.98	3.28	92.82	96.61
		PSVM p=2	89.20	4.29	86.45	91.63
		SVM $\sigma = 0.8$	94.98	2.05	93.82	96.81
	40%	QSVM	99.57	0.16	98.93	100
		PSVM p=2	78.13	106.14	67.55	90.95
		SVM $\sigma = 0.3$	92.02	5.97	89.14	96.80
	60%	QSVM	100	0	100	100
		PSVM p=2	58.80	128.80	50.38	78.17
		SVM $\sigma = 0.1$	100	0	100	100
	LOO	QSVM	90	80	84	100
		PSVM p=2	84	680	40	100
		SVM $\sigma = 0.2$	86	80	80	100

Table 5 Results of the QSVM and the Kernel SVM for the **Breast-Cancer** database

Breast	%train		Mean	Var	Min	Max
	20%	QSVM	95.76	4.41	92.33	97.62
		PSVM p=2	83.83	728.42	35.58	97.44
		SVM $\sigma = 1$	95.83	4.30	92.33	97.44
	40%	QSVM	97.95	0.16	97.56	98.54
		PSVM p=2	88.66	129.98	70.31	97.08
		SVM $\sigma = 0.5$	89.87	35.56	83.21	96.35
	60%	QSVM	100	0	100	100
		PSVM p=2	85.8394	90.86	75.54	96.71
		SVM $\sigma = 0.1$	99.56	0.95	97.81	100
	LOO	QSVM	92	100	80	100
		PSVM p=2	88	140	80	100
		SVM $\sigma = 0.1$	90	120	80	100

References

1. Cao, L.J., Tay, F.E.H.: Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Transactions on Neural Networks **14**(6), 1506–1518 (2003)
2. Chapelle, O.l., Haffner, P., Vapnik, V.N.: Support Vector Machines for Histogram-Based Image Classification. IEEE Transactions on Neural Networks **10**(5), 1055–1064 (1999)
3. Chong, E.K.P., Zak, S.H.: An Introduction to Optimization. Wiley Inter-Science (1996)
4. Cifarelli, C., Nieddu, L., Seref, O., Pardalos, P.M.: K-T.R.A.C.E: A kernel k-means procedure for classification. Journal Of Computer Operational Research **34**(10), 3154–3161 (2007)
5. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines (and other kernel-based learning methods). Cambridge University Press (2000)
6. Drucker, H., Wu, D., Vapnik, V.N.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks **10**(5), 1048–1054 (1999)
7. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. Wiley- InterScience (1973)

8. Abello, J., Pardalos, P.M., Resende, M.G.C.: Handbook of massive data sets. Kluwer (2002)
9. Pang, S., Kim, D., Bang, S.Y.: Face Membership Authentication Using SVM Classification Tree Generated by Membership-Based LLE Data Partition. IEEE Transactions on Neural Networks **16**(2), 436–446 (2005)
10. Scholkopf B.: Support Vector Learning. PhD thesis, Technical University of Berlin (1997)
11. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press Cambridge MA (2002)
12. UCI Machine Learning repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html
13. Vapnik, V: The Nature of Statistical Learning Theory. Springer-Verlag (1995)