Applied Soft Computing 24 (2014) 1105–1111

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Training of support vector machine with the use of multivariate normalization

F.J. Martínez López [a], S. Martínez Puertas [b,*], J.A. Torres Arriaza [a]

[a] *Computer Department, University of Almeria, Almeria, Spain*
[b] *Math Department, University of Almeria, Almeria, Spain*

ARTICLE INFO

Article history:
Received 31 October 2011
Received in revised form 4 July 2014
Accepted 11 August 2014
Available online 27 August 2014

Keywords:
Support vector machines
Normalization
Mahalanobis distance
Classification algorithms

ABSTRACT

SVM (support vector machines) techniques have recently arrived to complete the wide range of classification methods for complex systems. These classification systems offer similar performances to other classifiers (such as the neuronal networks or classic statistical classifiers) and they are becoming a valuable tool in industry for the resolution of real problems. One of the fundamental elements of this type of classifier is the metric used for determining the distance between samples of the population to be classified. Although the Euclidean distance measure is the most natural metric for solving problems, it presents certain disadvantages when trying to develop classification systems that can be adapted as the characteristics of the sample space change. Our study proposes a means of avoiding this problem using the *multivariate normalization* of the inputs (both during the training and classification processes). Using experimental results produced from a significant number of populations, the study confirms the improvement achieved in the classification processes. Lastly, the study demonstrates that the *multivariate normalization* applied to a real SVM is equivalent to the use of a SVM that uses the Mahalanobis distance measure, for non-normalized data.

1. Introduction

1.1. *Support vector machines*

SVMs have rapidly become tools for general use in the field of pattern recognition. The simplest application of this technique is the problem of binary classification (where only two classes are defined). The underlying idea [1] is to find a hypothesis H that minimizes the probability of empirical error (the probability that H contains an error in a test set selected at random). In Ref. [2], it is demonstrated that minimizing the empirical error is equivalent to finding the hyperplane (Figs. 1 and 2) that lies at the maximum distance from the closest training samples for the two classes.

1.2. *Mathematical basis*

Let there be n samples, independent and identically distributes, taken from an unknown probability distribution $P(x, y)$, consisting of pairs conformed by a vector $x_i \in \mathcal{R}^n$, i.e

$$x_i^T = (x_{i1}, x_{i2}, \ldots, x_{in})_{1\times n}$$

and a class label $(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)$.

We wish to construct a SVM that knows how to relate the values of the vectors x_i with the corresponding values of the labels y_i, by means of a decision $y : \mathcal{R}^n \longrightarrow \{-1, 1\}$, which lies between two hyperplanes that define a margin of maximum size.

1.2.1. *The linearly separable case*

The simplest case [3], is where the data can be classified by a *separating hyperplane*, with the equation

$$\omega^T \cdot x + b = 0 \tag{1}$$

where ω is a vector normal to the hyperplane, called the weight vector, given by

$$\omega^T = (\omega_1, \omega_2, \ldots, \omega_n)_{1\times n}$$

and b is known as the *bias*.

This hyperplane must be optimum, and so the values of ω and b must fulfil

$$\min_{(\omega,b)} \frac{1}{2}\|\omega\|^2 = \min_{(\omega,b)} \frac{1}{2}\omega^T \cdot \omega \tag{2}$$

subject to the restrictions

$$-y_i(\omega^T \cdot x_i + b) + 1 \le 0\,\text{fori} = 1, 2, \ldots, l \tag{3}$$

The solution to this optimization problem [4] is found at the saddle point of the Lagrangian function. In order to find this, we

[*] Corresponding author. Tel.: +34 950015672.

http://dx.doi.org/10.1016/j.asoc.2014.08.020
1568-4946/© 2014 Elsevier B.V. All rights reserved.

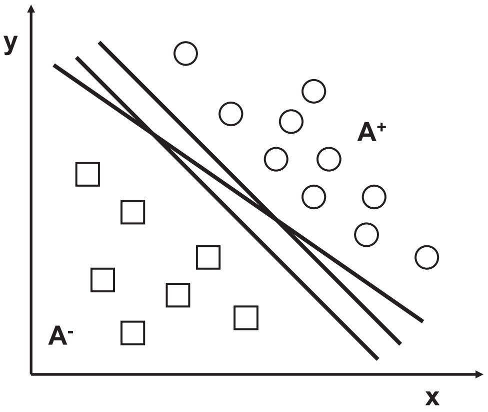

Fig. 1. Possible separating hyperplanes.

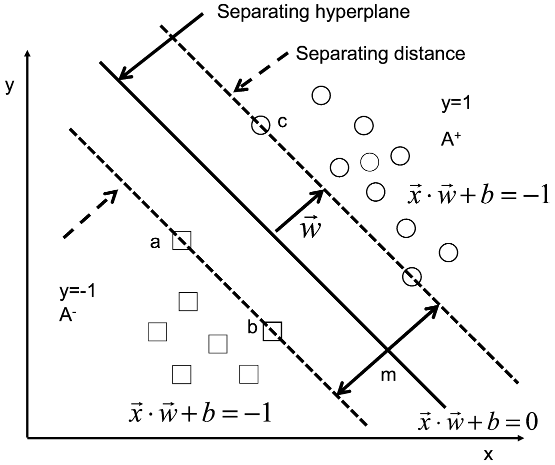
Separating hyperplane
Separating distance
y
c
y=1
A+
$\vec{x}\cdot\vec{w}+b=-1$
$\vec{w}$
a
y=-1
A-
b
m
$\vec{x}\cdot\vec{w}+b=-1$
$\vec{x}\cdot\vec{w}+b=0$
x

Fig. 2. Optimum separating hyperplane.

must minimize the Lagrangian with respect to ω and b, at the same time maximizing it with respect to α_i.

In this way, the optimum weight vector and the optimum bias will be defined by

$$\omega^* = \sum_{i=1}^{l} \alpha_i^* \cdot x_i \cdot y_i \tag{4}$$

$$b^* = \frac{1}{k}\left[\sum_{s=1}^{k} (y_s - (\omega^*)^T \cdot x_s)\right] \tag{5}$$

where k is the number of support vectors (with $\alpha_i^* > 0$ with $i = 1, 2, \ldots, l$ are the Lagrange multipliers).

1.2.2. The non-linear case

As expressed in Ref. [5], in the majority of cases it is not possible to separate using a linear frontier in the space dimension of the data. In this case, the SVM can project the vector of input data into non-linear regions, by means of mapping these points x_i. Using a suitable non-linear projection, it will be possible to separate the support vectors in that hyperspace.

To do this, we need to take the points and, using the function $\varphi : \mathcal{R}^n \longrightarrow \mathcal{R}^m$, project them into the feature space $\mathcal{R}^m$. In this way, if we obtain an optimum separation hyperplane in the space $\mathcal{R}^m$, we can state that a region of non-linear separation also exists in $\mathcal{R}^n$.

According to Ref. [5], the principal problem of this projection, known as the *curse of dimensionality*, is the potential increase in computation time. For this reason, a symmetrical function is used, known as a *kernel* [6], calculated from the points in the input space as $K(u, v) = \varphi(u) \cdot \varphi(v)$, which enables the operations to be executed in the input space. Thus, the scalar product does not necessarily need to be evaluated in the feature space (provided the kernels comply with the *Mercer Conditions* [7]).

The region of non-linear separation can be found as the solution to the linear problem, but with $\varphi(x_i)$ instead of x_i

$$\min_{(\varphi(\omega),b)} \frac{1}{2}\|\varphi(\omega)\|^2 = \min_{(\varphi(\omega),b)} \frac{1}{2}\varphi(\omega)^T \cdot \varphi(\omega) \tag{6}$$

and subject to the restrictions

$$-y_i(\varphi(\omega)^T \varphi(x_i) + b) + 1 \leq 0 \,\text{for}\, \alpha_i \geq 0, i = 1, 2, \ldots, l \tag{7}$$

The optimum solution, as in the linear case, is found at the Lagrangian saddle point, and so the Lagrangian must be minimized with respect to $\varphi(\omega)$ and b, at the same time as maximizing it with respect to α_i.

So, the optimum bias is obtained as follows

$$b^* = \frac{1}{k}\left[\sum_{s=1}^{k}\left(y_s - \sum_{i=1}^{l} \alpha_i \cdot y_i \cdot k(x_s^T \cdot x_i)\right)\right] \tag{8}$$

where k is the number of support vectors.

The classification is made using the same function as in the linearly separable case, but in this case, it is not possible to explicitly calculate the weight vector (for which we would need to know φ). Therefore, we employ the following expression [8]:

$$u(x) = \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot k(x_i^T \cdot x) + b^* \tag{9}$$

1.3. Data normalization to improve the generalization performance with SVM

The normalization of the input data of the SVM is an option that is increasingly used in the classification process of SVMs. There are many articles dealing with this, suggesting a wide combination of proposals. Among the numerous proporsals for normalization we have reviewed in the literature, we can cite [9–12] to improve the accuracy or [13] to improve the speed up of the learning phase.

We have studied some of these normalizations [14,9] of which we can highlight those proposed by Ref. [11]. These are the *Min–Max Normalization* and the *Zero-Mean Normalization*.

1.3.1. Min–Max normalization

The formulation of the *Min–Max normalization* is:

$$D\prime(i) = \frac{D(i) - \min(D)}{\max(D) - \min(D)} \cdot (U - L) + L \tag{10}$$

where D' is the normalized data matrix, D is the natural data matrix and U and L are the upper and lower normalization bounds.

This type of normalization method is used to normalize a data matrix into a desired bound. The most popular bound is between 0 and 1. We also change bound values to between 0 and −1 or 1 and −1.

1.3.2. Zero-Mean normalization

The formulation of the *Zero-Mean normalization* is as follows:

$$D\prime = \frac{D - \overline{D}}{\sigma} \tag{11}$$

where $\overline{D}$ is the mean of the data matrix D and σ is the standard deviation of the same data matrix. In this normalization method, the mean of the normalized data points is reduced to zero. As a result, the mean and standard deviation of the natural data matrix are required.