Pattern Recognition Letters 30 (2009) 1507–1515

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

IP-LSSVM: A two-step sparse classifier

B.P.R. Carvalho *, A.P. Braga

Depto. Engenharia Eletrônica, Campus da UFMG, Pampulha, 31.270-901 Belo Horizonte, MG, Brazil

ARTICLE INFO

Article history:
Received 25 February 2008
Received in revised form 5 February 2009
Available online 7 August 2009

Communicated by P. Sarkar

Keywords:
Sparse classifier
Least squares support vector machine
Support vector automatic detection

ABSTRACT

We present in this work a two-step sparse classifier called $IP-LSSVM$ which is based on Least Squares Support Vector Machine (LS-SVM). The formulation of LS-SVM aims at solving the learning problem with a system of linear equations. Although this solution is simpler, there is a loss of sparseness in the feature vectors. Many works on LS-SVM are focused on improving support vectors representation in the least squares approach, since they correspond to the only vectors that must be stored for further usage of the machine, which can also be directly used as a reduced subset that represents the initial one. The proposed classifier incorporates the advantages of either SVM and LS-SVM: automatic detection of support vectors and a solution obtained simply by the solution of systems of linear equations. $IP-LSSVM$ was compared with other sparse LS-SVM classifiers from literature, LS^2-SVM, $Pruning$, $Ada-Pinv$ and $RRS+LS-SVM$. The experiments were performed on four important benchmark databases in Machine Learning and on two artificial databases created to show visually the support vectors detected. The results show that $IP-LSSVM$ represents a viable alternative to SVMs, since both have similar features, supported by literature results and yet $IP-LSSVM$ has a simpler and more understandable formulation.

1. Introduction

The success of Support Vector Machine (SVM) (Vapnik, 1995) is mainly due to its solid formal basis and elegant approach in margin maximization and support vectors selection. Maximum margin hyperplane can be obtained thanks to the quadratic programming (QP) approach to the learning problem, while support vectors are outlined by the sensitivity of the corresponding Lagrange multipliers (Vapnik, 1995), which are non-zero in the separation margin. Nevertheless, alternatives to the quadratic programming approach, such as the Least Squares Support Vector Machine (LS-SVM) (Suykens and Vandewalle, 1999) are found in the literature. LS-SVM yields simplicity by solving the primal problem as a system of linear equations. The least squares (LS) solution is less computationally intensive than the quadratic programming one, but it also results on loss of sparseness of the Lagrange multipliers vector. Therefore, selecting LS-SVM support vectors by the non-zero criterion usually results on all training patterns being considered as support vectors, what is sometimes regarded as a drawback of the LS approach.

The importance of an optimal number of support vectors can not be neglected in a classification problem, since they represent the most relevant samples for outlining the separation boundary. Support vectors are useful for representing large static and dynamic data sets for classification purposes and can also help in problem analysis by pointing out to the most relevant cases (Tax and Duin, 1999; Ganapathiraju and Picone, 2000). As a consequence of this trade-off between sparseness and complexity, many works on LS-SVM are focused on improving support vectors representation of the LS approach (Suykens et al., 2000; Valyon and Horváth, 2004; Carvalho and Braga, 2005; Carvalho et al., 2007). The motivation behind these works are that LS-SVM may still provide a reduced set of support vectors, by simply observing the proper features from the LS solution.

SVM's constrained optimization problem is formalized in the LS-SVM approach as a least squares problem in the form $\mathbf{AX}=\mathbf{B}$, where $\mathbf{A}$ contains mainly kernel mapping information, $\mathbf{X}$ contains the optimization parameters (Lagrange multipliers α and bias b) and $\mathbf{B}$ is a vector of equality constrains. The problem of support vectors identification in this approach can be regarded as the one of solving the optimization problem with the smallest possible vector $\mathbf{X}$. This would result on a maximum margin with minimum number of support vectors. The problem is therefore on selecting rows of $\mathbf{X}$ without changing the separating hyperplane and yet maintaining the original LS-SVM formulation.

In order to avoid kernel mapping information loss due to dimensionality reduction of $\mathbf{A}$ as a consequence of eliminating rows of $\mathbf{X}$, the $IP-LSSVM$ approach presented in this paper maintains labeling information in $\mathbf{A}$ for all patterns in the data set, including those that had their corresponding rows eliminated

* Corresponding author. Fax: +55 3132416175.
E-mail addresses: bpenna@gmail.com (B.P.R. Carvalho), apbraga@cpdee.ufmg.br (A.P. Braga).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.07.022

in **X**. The problem is solved in two steps. The first one corresponds to a feed-forward LS-SVM phase with the objective of obtaining the Lagrange multipliers. In the second one, vector elimination is followed by feed-forwarding the inputs with support vectors only. The mapping obtained in both phases should match, despite of the dimensionality reduction in the last one. In spite of the LS Lagrange multipliers vectors not being sparse, their magnitude do contain boundary information. *IP – LSSVM* takes advantage of this by selecting support vectors according to the magnitudes of their Lagrange multipliers. The new criterion, that is based on the support of two parallel hyperplanes, is more consistent with the concept of a SVM classifier and has yielded better results than those obtained with current approaches (Section 2).

Some of the most recent sparse LS-SVM classifiers found in the literature (Section 3) were evaluated on four benchmark classification databases (Blake and Merz, 1998) in the experiments of Section 5: Ionosphere, Pima Indian Diabetes, Bupa Liver Disorder and Tic Tac Toe. A high rank of similarity among the support vectors obtained with *IP – LSSVM* and those generated with QP SVMs was achieved for different real and synthetic data sets (Section 5). This suggests that the proposed approach can take advantage of least squares simplicity and still detect quadratic programming support vectors.

2. Least Squares Support Vector Machine

Given the training set $\{\mathbf{x}_i, y_i\}_{i=1}^N$ with $\mathbf{x}_i \in \Re^n$ and $y_i \in \{-1, +1\}$, the basic principle of SVMs is to map the input data into a high dimensional feature space by means of kernel functions. Kernel mapping results on a linearly separable problem in the feature space that can be solved with a hyperplane in the form $\boldsymbol{\omega}^T\varphi(\mathbf{x}) + b = 0$ where $\boldsymbol{\omega}$ is the parameter's vector, b is the bias term and $\varphi(\cdot)$ is the mapping function. Margin maximization is obtained by minimizing the squared norm of $\boldsymbol{\omega}$ while also minimizing the error of the training set. The resulting optimization problem is usually formulated within constrained optimization principles. The primal LS-SVMs expression for solving this problem is presented in Eq. (1). The slack variable e_i that appears in both the cost function and in the constrain of the equation has the function of controlling the margin width or, in other words, the distance between the separating hyperplane and the two parallel hyperplanes that encapsulate the margin. The error of the training data is optimized by

$$min_{\omega,b,\mathbf{e}}\ J_p(\boldsymbol{\omega}, b, \mathbf{e}) = \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 \tag{1}$$

subject to

$$y_i[\boldsymbol{\omega}^T\varphi(\mathbf{x}_i) + b] = 1 - e_i, \quad i = 1, \ldots, N$$

where γ is a margin parameter, analogous to SVM's C.

After deriving the Lagrangean of Eq. (1) in relation to its primal and dual variables, Eq. (2) is obtained.

$$\begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \alpha \sum_{i=1}^{N}\sum_{j=1}^{N}(y_i y_j \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) + \frac{1}{\gamma}) + yb = 1 \end{cases} \tag{2}$$

Eq. (2) can be written as a linear system $\mathbf{AX} = \mathbf{B}$ where

$$\mathbf{A} = \begin{bmatrix} 0 & -\mathbf{Y}^T \\ \mathbf{Y} & \mathbf{H} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \tag{3}$$

$$\text{with} \quad \mathbf{H} = \mathbf{Z}\mathbf{Z}^T + \frac{\mathbf{I}}{\gamma} \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} \varphi(x_1)y_1 & \cdots & \varphi(x_1)y_1 \\ \vdots & \ddots & \vdots \\ \varphi(x_N)y_N & \cdots & \varphi(x_N)y_N \end{bmatrix} \tag{4}$$

Once the Lagrange multipliers and bias term are obtained from Eq. (3), the output of the LS-SVM can be calculated by simply applying the expression $f(\mathbf{x}) = sign\left[\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right]$ where $K(\mathbf{x}, \mathbf{x}_i) = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{x}_i)$.

Considering that $\alpha_i = \gamma e_i$ (Eq. (2)) it is possible to assume that rarely a Lagrange multiplier α will be zero in the solution of a LS-SVM (Suykens and Vandewalle, 1999), what makes the range of values of α different from those obtained by the quadratic programming solution. This happens because γ does not impose a range constraint in α like the parameter C does in QP SVMs where $0 \leqslant \alpha \leqslant C$. Nevertheless, it will be shown in the next sections that *IP – LSSVM* is able to express support vectors that are very close to those obtained by QP SVMs.

3. Sparse methods for LS-SVM

The most relevant methods for enhancing sparseness in LS-SVM Lagrange multiplier vector are described in this section. The sparse methods presented are $LS^2 - SVM, Pruning, Ada - Pinv$ and $RRS + LS - SVM$. These methods are compared with our proposed classifier *IP – LSSVM* in the results and discussions section.

3.1. $LS^2 - SVM$

This method was proposed on Valyon and Horváth (2004), using some ideas from RSVM (Lee, 2001), such as the elimination of columns of **A** without eliminating the corresponding rows. Likewise our *IP – LSSVM* approach, this is a two-phase method that attempts to reduce **A** in order to detect the support vectors. The first phase is carried out by reducing the dimension of matrix **A** in Eq. (3) with a column elimination algorithm only (Valyon and Horváth, 2004). The objective is to perform elementary operations in matrix **A** with the aim of obtaining its echelon reduced form matrix **A**′. A threshold function is applied to **A**′ so that its elements that are smaller than a threshold $\epsilon \in \Re$ are set to zero. After obtaining the reduced matrix **A**′, the corresponding columns that have only zero elements are eliminated. Since all rows of **A**′ were maintained, while some columns were removed, the new matrix **A**′ is not square. Therefore, in the second step, the reduced linear system $\mathbf{A}'\mathbf{X} = \mathbf{B}$ becomes over-determined and the pseudo-inverse $(\mathbf{A}')^+$ needs to be calculated in order to find a solution for **X**. This method has an extra training parameter ϵ that corresponds to a numeric tolerance used by the process of reduction to the echelon form, described above.

3.2. Pruning

In this method (Suykens et al., 2000), training vectors $\mathbf{x}_i$ are eliminated according to the absolute value of their Lagrange multipliers $|\alpha_i|$. The process is accomplished recursively, with gradual vector elimination at each iteration, until a stop criterion is reached, which is usually associated with decrease in performance on a validation set. Vectors are eliminated by setting the corresponding Lagrange multipliers to zero, without any change in matrix dimensions. The resolution of the current linear system, for each new reduced set, is needed at each iteration, and the reduced set is selected from the best iteration. This is a multi-step method, since the linear system needs to be solved many times until the convergence criterion is reached.