A New Features Selection model: Least Squares Support Vector Machine with Mixture of Kernel

Liwei Wei[1, a *], Wenwu Li[1,b] and Qiang Xiao[2,c]

[1]Dept. of laboratory management, China national institute of standardization, Beijing, China

[2]Dept. of business management, Beijing International Electric Engineering Ltd. Co. Beijing, China

[a]weilw@cnis.gov.cn, [b]liww@cnis.gov.cn, [c]xiaoqiang@snpdri.com

Keywords: data classification;LS-SVM-MK;mixture kernel;SVM;LS-SVM.

Abstract. In this paper, a least squares support vector machine with mixture kernel (LS-SVM-MK) is proposed to solve the problem of the traditional LS-SVM model, such as the loss of sparseness and robustness. Thus that will result in slow testing speed and poor generalization performance. The revision model LS-SVM-MK is equivalent to solve a linear equation set with deficient rank just like the over complete problem in independent component analysis. A minimum of 1-penalty based object function is chosen to get the sparse and robust solution. Some UCI datasets are used to demonstrate the effectiveness of this model. The experimental results show that LS-SVM-MK can obtain a small number of features and improve the generalization ability of LS-SVM.

Introduction

Support vector machines (SVM) [1]-[2] is powerful new tools for data classification and function estimation. Recently SVM have received a lot of attention in the machine learning community because of their remarkable generalization performance. The SVM typically follows from the solution to a quadratic programming. Despite its many advantages, one problem is that the size of the matrix of the quadratic programming is directly proportional to the number of training points. Thus this greatly increases the computational complexity [3], especially for the problems which deal with mass data or need on-line computation. Least squares support vector machine just makes up for that shortcoming.

Least squares support vector machine (LS-SVM) [4]-[5] is equivalent to solve a set of linear equations instead of a quadratic programming. Because the e –insensitive loss function used in SVM is replaced by a sum square error loss function, the inequality restriction is replaced by the equation restriction. Thus this makes the least squares support vector machine achieve lower computational complexity. But there are some potential drawbacks for LS-SVM [6]. The first drawback is that the usage of the sum square error may lead to less robust estimates. Reference [6] presents a weighted LS-SVM to solve this issue. This method needs an interactive procedure to get optimal cost function and robust estimation gradually. The second drawback is that the sparseness of the data points is lost. The pruning method [7] is used to get the sparse solution by omitting a relative small amount of the least meaningful data points. It also needs a series of steps for LS-SVM to retrain. A more sophisticated pruning method [8] introduces a procedure that the training samples be selected from a data set, and these training samples will introduce the smallest approximation error that can be omitted. Another method [9] deletes some columns of the coefficient matrix through a certain measure. When the final model is used to represent the original system, the performance would be hurt.

Recently, how to learn the kernel from data draws many researchers' attention. The reference [10] draws the conclusion that the optimal kernel can always be obtained as a convex combinations of many finitely basic kernels. And some formulations [11] [12] have been proposed to perform the optimization in manner of convex combinations of basic kernels.

Motivated by above questions and ideas, we propose a new method named least squares support vector machines with mixture of kernel (LS-SVM-MK) to classify the data. In this method the kernel is a convex combination of many finitely basic kernels. Each basic kernel has a kernel coefficient and is provided with a single feature. The 1-norm is utilized in LS-SVM-MK. As a result, its objective function turns into a linear programming parameter iterative learning procedure and greatly reduces the

computational complexity. Furthermore, we can select the optimal feature subset automatically and get an interpretable model.

This paper is organized as follows. In section 2, we give the LS-SVM-MK formulations and then set up the corresponding solutions. Numerical test results represent in Section 3 shows that our LS-SVM-MK is of good sparse and robustness performance. Section 4 concludes the paper and introduces some future research directions.

LS-SVM with mixture of kerneL

A.LS-SVM-MK

Like least squares support vector machine, the object function for the LS-SVM-LP is defined as:

$$\min J(\mathbf{r}_{w,e}) = \frac{1}{2}\|w\|^2 + \frac{1}{2}g\sum_{i=1}^{n} e_i^2 \tag{1}$$

For classification problems, it subjects to:

$$y_i(w^T(k)j(x_{i,k}) + b) = 1 - e_i, i = 1, \mathbf{L}, n, k = 1, \mathbf{L}, m \tag{2}$$

where $x_{i,k}$ denotes the k^{th} component of the input vector x_i. It can be overcomplete dictionaries such as wavelet.

By introducing Lagrange multipliers $a_{i,k}$, the corresponding Lagrangian is given by:

$$L(w,b,e,a) = J(w,e) - \sum_{i=1}^{n}\sum_{k=1}^{m} a_{i,k}\{y_i(\mathbf{r}_{w^T}(k)j(x_{i,k}) + b) + e_i - 1\} \tag{3}$$

where $a_{i,k}$ is the Lagrange multiplier for the k^{th} component of sample i.

According to Kuhn-Tucker conditions, the following functions can be got:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\sum_{k=1}^{m} a_{i,k} y_i j(x_{i,k}) \tag{4}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\sum_{k=1}^{m} a_{i,k} y_i = 0 \tag{5}$$

$$\frac{\partial L}{\partial e_k} = 0 \Rightarrow e_i = \sum_{k=1}^{m} a_{i,k} / g \tag{6}$$

Substitute equations (4) and (6) into equation (2), then equation (2) is transformed to the following form:

$$y_i\left(\sum_{j=1}^{n}\sum_{k=1}^{m} a_{j,k} y_j k(x_{i,k}, x_{j,k}) + b\right) + \sum_{k=1}^{m} a_{j,k} / g = 1, i = 1, \mathbf{L}, n \tag{7}$$

Equations (5) and (7) can be written as the following matrix form:

$$A_1 * \begin{bmatrix} b \\ \mathbf{r}_a \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{r}_l \end{bmatrix} \tag{8}$$

Where

$$A_1 = \begin{bmatrix} 0 & \mathbf{r}_{y^T} & \mathbf{L} & \mathbf{r}_{y^T} \\ \mathbf{r}_y & K_1 & \mathbf{L} & K_m \end{bmatrix}_{(n+1)\times(mn+1)}$$

$\mathbf{r}_{l^T} = [1, \mathbf{L}, 1]_{1\times n}$, $\mathbf{r}_a = [a_{1,1}, \mathbf{L}, a_{n,1}, a_{1,2}, \mathbf{L}, a_{nm}]$ and

$$K_d = \begin{bmatrix} y_1 y_1 k(x_{1,d}, x_{1,d}) + \frac{1}{g} & \mathbf{L} & y_1 y_n k(x_{1,d}, x_{n,d}) \\ & \mathbf{M} & \\ y_n y_1 k(x_{n,d}, x_{1,d}) & \mathbf{L} & y_n y_n k(x_{n,d}, x_{n,d}) + \frac{1}{g} \end{bmatrix}, d = 1, \mathbf{L}, m \tag{9}$$

The following equation is the standard form of LS-SVM:

$$\begin{bmatrix} 0 & \vec{y}^T \\ \vec{y} & K+\frac{I}{g} \end{bmatrix}\begin{bmatrix} b \\ \vec{a} \end{bmatrix}=\begin{bmatrix} 0 \\ \vec{l} \end{bmatrix} \tag{10}$$

Compared equation (8) with the standard form of LS-SVM in equation (10), we can find that the kernel mapping is executed in each component and the Lagrange multiplier $a_{i,k}$ can be seen as the weight for each component and sample other than only for each sample in other methods.

Then the output is obtained:

$$f(x)=\operatorname{sgn}\left(\sum_{j=1}^{n}\sum_{k=1}^{m} y_i a_{i,k} k(x_{i,k}, x_{j,k})+b\right) \tag{11}$$

In practice, a simple and efficient method is that the kernel function being illustrated as the convex of combinations of the basic kernel:

$$k(\vec{x}_i, \vec{x}_j)=\sum_{i=1}^{n} b_d k(x_{i,d}, x_{j,d}) \tag{12}$$

where $x_{i,d}$ denotes the d^{th} component of the input vector $\vec{x}_i$.

Substituting Equation (12) into Equation (7), and multiplying Equation (7) and (8) by b_d, suppose $h_{i,d}=a_i \cdot b_d$, then the Lagrange dual problem change into equation (13). Function (11) is equivalent to the sum of the sub-function in different elements:

$$f(x)=\operatorname{sgn}\left(\sum_{k=1}^{m} f_k(x)+b\right)=\operatorname{sgn}\left(\sum_{k=1}^{m}(\sum_{i=1}^{n} y_i h_{i,k} k(x_{i,k}, x_k))+b\right) \tag{13}$$

where $f_k(x)$ represents the contribution for the output by each element.

B.Finding Solutions

From equations (8), we can find that the new LS-SVM is equivalent to solve a deficient rank linear equation set just like the overcomplete problem in ICA. Because the matrix A is $n \times nm$, there are infinite solutions to equations (8). It brings us a chance and challenge to get sparse solutions. There are many approaches presented to resolve this problem, including the method of Frames (MOF) and basis pursuit (BP) [11]-[12].

Unlike MOF, BP replaces the l^2 norm with the l^1 norm:

$$\min \left\|\vec{q}\right\|_1 \tag{14}$$

Subject to $A*\vec{q}=c$ (15)

where $\vec{q}=\begin{bmatrix} b \\ \vec{h} \end{bmatrix}$.

It is a very important character that $e_i=\sum_{k=1}^{m} a_{i,k}/h$. Because b is a constant, the minimum of $\left\|\vec{q}\right\|_1$ is equivalent to that of $\left\|\vec{a}\right\|_1$. And from equation (6), we can conclude that:

$$\left\|\vec{e}\right\|_1=\sum_{i=1}^{n}|e_i|=\frac{\sum_{i=1}^{n}\left|\sum_{k=1}^{m} a_{i,k}\right|}{h} \le \frac{\left|\sum_{i=1}^{n}\sum_{k=1}^{m} a_{i,k}\right|}{h}=\frac{\left\|\vec{a}\right\|_1}{h}$$

So the minimum of $\left\|\vec{a}\right\|_1$ can guarantee $\left\|\vec{e}\right\|_1$ in a lower level. And it improves the robustness for the final solution. Of course, we can use other optimization forms or algorithms according to the requirements of the problems. The flexibility is just the most advantages for this method. So the new LS-SVM method is called least squares support vector machine with linear programming formulation.

C.Algorithm

The procedures to implement the MK-LS-SVM can be summarized as the following steps:

1) Data Preprocess: firstly, the data is normalized for convenience. Then the input data is presented in the overcomplete forms.

2) Initialization Parameters: according to some criteria or experience, the coefficient h must be given some initial value.

3) Construct Formulation: the new model based on the LS-SVM is constructed according to the equation (8).

4) Solving: a linear programming with equality constrains (10) is solved to get the Lagrange coefficients a and b.

5) Output: then the output is got according to equation (13).

6) Calculate Errors: some defined measures are calculated according to the coefficients a and b solved by step 4.

If we aren't satisfied with the results, change the coefficients and go back to step 2.

Experiments Analysis

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

This section reports the results of our empirical analysis on the above presented MK-LS-SVM algorithm. And we discuss the comparisons between MK-LS-SVM and other popular models, such as LS-SFM, GA-based SVM, wavelet neural network, new class of wavelet neural network, LS-SVM and ridge regression.

The classification performance is measured by its specificity accuracy, sensitivity accuracy and overall accuracy, which are the percent of correctly classified of healthy records, percent of correctly classified of patients and the percent of correctly classified in total, respectively.

$$Sensitivity = \frac{TN}{TN + FP} \tag{16}$$

$$Specificity = \frac{TP}{TP + FN} \tag{17}$$

$$\text{Overall} = \frac{TN + TP}{TN + FP + TP + FN} \tag{18}$$

Where TN is a sample which represents the number of samples for the sick sample itself is sick, it was also correctly classified as sick. FP represents the number of samples is the sample itself is sick, but it was mistakenly classified as healthy. Similarly, TP represents a sample of the sample itself is healthy, it is correctly classified as a healthy number of samples. FP is a healthy sample representative of the sample itself was classified as the number of sick samples.

To decrease the bias arising from the choice of split between training sample and test sample, we randomly select the training samples from the dataset and make use of five-fold cross validation.

Here we will test three UCI database to verify the validity of our proposed model. These databases are often used as a reference database to verify the validity of the classification model. These three databases[13] are Wisconsin breast cancer dataset (WBCD), Heart disease dataset (HD) and PIMA database, their basic information shown in the chart I below.

Table 1. There UCI databases

Database	Class number	Records number	Feature number
WBCD	2	699	9
HD	2	270	13
PIMA	2	768	8

For the LS-SVM-MK classifier, the Gaussian kernel is used. So the kernel parameter and regularization parameter g need to be chosen. Table I shows test set correctness, using the

LS-SVM-MK with various parameters g when s^2 is equal to 1000, under five-fold cross validation for the above mentioned dataset. The column titled number of selected features (NSF) is the number of support vectors selected from the training samples.

LS-SVM-MK average test results on three databases are shown in Table II. The kernel parameter and regularization parameters are based on the results of cross-validation selection. According to experience, our proposed classification accuracy of the model is good, but also in ensuring good sensitivity and specificity in the lower level, the more difficult points PIMA database we used only two-dimensional features, while the total classification accuracy was 77.92%. For the other two databases, the selected feature numbers were 2 and 5, the classification results using these features to train the classifier was also very good. As can be seen, least squares support vector machines with mixture of kernels to get the final result only a few iterative steps, the algorithm speed is very fast.

Table 2. Experimental results of MK-LS-SVM using three database

Database	**WBCD**	**HD**	**PIMA**
Feature number	9	13	8
Number of selection features(NSF)	2	5	2
Interactive iteration number	3	3	1
Overall	97.51	95.90	77.92
sensitivity	98.04	98.48	59.59
specificity	97.26	92.79	87.40

In order to further evaluate the effectiveness of the proposed LS-SVM-MK, the classification results are compared with some other methods using the same dataset, including LS-SFM, GA-based SVM and LS-SVM. The results of the LS-SFM model and GA-based SVM model are quoted from the reference [14] and [15]. We summarized the results of the best model compared to the overall classification accuracy. And the results list as shown in Table III. As can be seen, our model for WBCD and HD database has the best classification results. PIMA database for our model is in the second pla

Table 3. Comparison of classification accuracy for three methods

Model	**WBCD**	**HD**	**PIMA**
MK-LS-SVM	97.51	95.90	77.92
GA-based SVM	96.19	94.80	81.50
Best other	96.80	84.00	76.50

ce,behind the GA-based SVM model. In comparison, mixture kernel least squares support vector machine performance is still very good. In addition, the model presented in this chapter can automatically select a subset of features. These results indicate that our method is very efficient in binary classification problem. The results show that there is only a small part of feature subset selected, so these results will greatly help the doctor for making a medical diagnosis. It should indeed be a better alternative as it can identify important independent variables which may provide valuable information for further managerial and related decision-makings.

Conclusions

Unlike SVM and weighted LS-SVM, the LS-SVM-MK is equivalent to get the minimum of a sum absolute error in the feasibility region. So this method can improve the robustness and get the sparseness for the solution simultaneously. Another advantage is that it is equivalent to solve a linear programming and do not increase the computational burden that much. In addition, the output of the LS-SVM-MK can be viewed as a weighted sum for different components. This makes the output more understandable. Furthermore, empirical results show that the LS-SVM-MK is very efficient in both classification and regression.

Further work includes: making this method more suitable for large-scale datasets by modifying the algorithm and extending this idea to other kernel methods. In addition, how to make use of feature coefficients to reduce some features to make the model more simple and interpretable is another future work.

Acknowledgements

This research has been supported by the Basal Research Funds for central public research institutes (#222015Y-4006), and a public benefit special fund from Quality inspection industry of China (#201510048), and also supported by the science and technology support program.(2014BAK07B00).

References

[1] V. Vapnik, The Nature of Statistic Learning Theory. Springer-Verlag New York, 1995.

[2] V. Vapnik, Statistic Learning Theory. Willey, New York, 1998.

[3] B. Scholkopf, and A. Smola, Learning with Kernels. MIT press, Cambridge, USA, 2002.

[4] J.A.K. Suykens, and J. Wandewalle, Least squares support vector machine classifiers, Neural Processing Letters. 9 (1999) 293-300.

[5] J.A.K. Suykens, T.V. Gestel, J.D. Branbater, B.D. Moor, and J. Wandewalle, Least squares support vector machines. World Scientific, Singapore, 2002.

[6] J.A.K. Suykens, J.D. Branbater, L. Lukas, and J. Wandewalle, Weighted least squares support vector machine: robustness and sparseness approximation, Neurocomputing. 48(2002) 85-105.

[7] J.A.K. Suykens, L. Lukas, and J. Wandewalle, Sparseness approximation using least squares support vector machines, IEEE international symposium on Circuits and System. 2(2000) 757-760.

[8] Y.G. Li, C. Lin, and W.D. Zhang, Improved sparse least-squares support vector machine classifiers, Neurocomputing. 69(2006)1655-1658.

[9] V. Jozsef, and H. Gabor, A sparse least squares support vector machine classifiers,IEEE international conference on neural network. 1(2004) 543-548.

[10] A. Hyvarinen, J. Karhunen, and E. Oja, Independent Component Analysis. Willey, New York, 2001.

[11] S.S. Chen, D.L. Donoho, and M.A. Sauders, Atomic decomposition by basis pursuit, SIAM review. 43(2001)129-159.

[12] P. Georqiev, and A. Cichocki, Sparse component analysis of overcomplete mixture by improved basis pursuit method, IEEE international symposium on Circuits and System. 5(2004)37-40.

[13] Information on http://www.ics.uci.edu/~mlearn/databases/

[14] L.W. wei, H. Yu, and J. H. Liu, Data classification using Sparse and Robust model: least squares support vector machine with L1 norm, computer modeling & new technologies. 18, (12B) (2014)686-691.

[15] C.L. Huang, C.J. wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Systems with Applications. 31(2006)231-240.