Data classification using Sparse and Robust model: least squares support vector machine with L1 norm

Liwei Wei[1]*, Hao Yu[2], Junhua Liu[1]

[1]*China National Institute of Standardization, No.4 Zhichun Road, Beijing, China*

[2]*China Agricultural University Library, No.2 Yuanmingyuan West Road, Beijing, China*

Received 23 November 2014, www.cmnt.lv

Abstract

Least squares support vector machine (LS-SVM) has an outstanding advantage of lower computational complexity than that of standard support vector machines. Its shortcomings are the loss of sparseness and robustness. Thus it usually results in slow testing speed and poor generalization performance. In this paper, a least squares support vector machine with L1 norm (LS-SVM-L1) is proposed to deal with above shortcomings. This method is equivalent to solve a linear equation set with deficient rank just like the over complete problem in independent component analysis (ICA). A minimum of 1-norm based object function is chosen to get the sparse and robust solution based on the idea of basis pursuit (BP) in the whole feasibility region. Some UCI datasets are used to demonstrate the effectiveness of this model. The experimental results show that LS-SVM-L1 can obtain a small number of support vector and improve the generalization ability of LS-SVM.

Keywords: data classification, LSSVM, data analysis

1 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 da ta 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 ε -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.

Focusing on the above-mentioned questions, we propose a new method to improve the sparseness and robustness of the LS-SVM. In this method, a L_1 norm representation is used as the object function. And LS-SVM is used to characterize the system as a set of linear equations with deficient rank just like the overcomplete problem in independent component analysis (ICA) [10]. So the solution with the minimum L_1. Norm is got based on the idea of basis pursuit (BP) in the whole feasibility region [11, 12]. BP is closely connected with linear programming. So the proposed method is called least squares support vector machine with linear programming formulation (LS-SVM-L_1 Above contents are introduced in chapter 2. Then the performance of this method is examined by three examples.

This paper is organized as follows. In section 2, we give the LS-SVM- L_1 classifier and regression formulations and then set up the corresponding solutions. Numerical test results represent in Section 3 shows that our LS-SVM- L_1 is of good sparse and robustness performance. Section 4 concludes the paper and introduces some future research directions.

[*] ***Corresponding author's*** e-mail: weilw@cnis.gov.cn

2 LS-SVM with *L₁* norm

2.1 LS-SVM-L1 CLASSIFIER

Like least squares support vector machine, the object function for the LS-SVM-L1 is defined as:

$$\min\ J(\vec{w},e)=\frac{1}{2}\|w\|^2+\frac{1}{2}\gamma\sum_{i=1}^{n}e_i^2 \tag{1}$$

For classification problems, it subjects to:

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

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 $\alpha_{i,k}$, the corresponding Lagrangian is given by:

$$L(w,b,e,\alpha)=J(w,e)- \sum_{i=1}^{n}\sum_{k=1}^{m}\alpha_{i,k}\left\{y_i\left(\vec{w}^T(k)\phi\left(x_{i,k}\right)+b\right)+e_i-1\right\} \tag{3}$$

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

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

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

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

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

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

$$y_i\left(\sum_{j=1}^{n}\sum_{k=1}^{m}\alpha_{j,k}y_jk(x_{i,k},x_{j,k})+b\right)+ \sum_{k=1}^{m}\alpha_{j,k}/\gamma=1, i=1,...,n \tag{7}$$

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

$$A_1\cdot\begin{bmatrix}b\\ \vec{\alpha}\end{bmatrix}=\begin{bmatrix}0\\ \vec{l}\end{bmatrix}, \tag{8}$$

where

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

$\vec{l}^T=[1,\cdots,1]_{1\times n}, \vec{\alpha}=[\alpha_{1,1},...,\alpha_{n,1},\alpha_{1,2},...,\alpha_{nm}]$ and

$$K_d=\begin{bmatrix}y_1y_1k(x_{1,d},x_{1,d})+\frac{1}{\gamma} & \cdots & y_1y_nk(x_{1,d},x_{n,d})\\ & \vdots & \\ y_ny_1k(x_{n,d},x_{1,d}) & \cdots & y_ny_nk(x_{n,d},x_{n,d})+\frac{1}{\gamma}\end{bmatrix} \tag{9}$$

$d=1,\cdots,m$

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

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

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 $\alpha_{i,k}$ 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:

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

Above function is equivalent to the sum of the sub-function in different elements:

$$f(x)=\mathrm{sgn}\left(\sum_{k=1}^{m}f_k(x)+b\right)= \mathrm{sgn}\left(\sum_{k=1}^{m}(\sum_{i=1}^{n}y_i\alpha_{i,k}k(x_{i,k},x_k))+b\right) \tag{12}$$

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

2.2 LS-SVM-L1 REGRESSION

Similarly, LS-SVM-L1 for regression problem can be described as the following mathematical programming:

$$\min\ J(\vec{w},e)=\frac{1}{2}\|w\|^2+\frac{1}{2}\gamma\sum_{i=1}^{n}e_i^2 \tag{13}$$

$$y_i=w^T(k)\varphi(x_{i,k})+b+e_i, i=1,...,n,k=1,...,m \tag{14}$$

where the constant γ is called the regularization parameter and $\varphi(x_{i,k})$ is a nonlinear mapping function.

By introducing the Lagrange multipliers $\alpha_{i,k}$, we obtain:

$$L(w,b,e,\alpha)=J(w,e)- \sum_{i=1}^{n}\sum_{k=1}^{m}\alpha_{i,k}\left\{\vec{w}^T(k)\phi(x_{i,k})+b+e_i-y_i\right\} \tag{15}$$

Similarly, Equation (14) is transformed to the following form:

$$y_i = \sum_{j=1}^{n}\sum_{k=1}^{m}\alpha_{j,k}k(x_{i,k}, x_{j,k}) + b + \sum_{k=1}^{m}\alpha_{j,k} / \gamma, \tag{16}$$
$$i = 1,...,n$$

So it is equivalent to solve the following equation set:

$$A_2 \cdot \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix}, \tag{17}$$

where $A_2 = \begin{bmatrix} 0 & \vec{l}^T & \cdots & \vec{l}^T \\ \vec{l} & K_1 & \cdots & K_m \end{bmatrix}_{(n+1)\times(mn+1)}$ and

$$K_d = \begin{bmatrix} k(x_{1,d}, x_{1,d}) + \frac{1}{\gamma} & \cdots & k(x_{1,d}, x_{n,d}) \\ & \vdots & \\ k(x_{n,d}, x_{1,d}) & \cdots & k(x_{n,d}, x_{n,d}) + \frac{1}{\gamma} \end{bmatrix}, \tag{18}$$
$$d = 1,...,m$$

Then the output is obtained:

$$y_i = \sum_{j=1}^{n}\sum_{k=1}^{m}\alpha_{i,k}k(x_{i,k}, x_{j,k}) + b\,. \tag{19}$$

Above function is equivalent to the sum of the sub-function in different elements:

$$f(x) = \sum_{k=1}^{m} f_k(x) + b = \sum_{k=1}^{m}(\sum_{i=1}^{n}\alpha_{i,k}k(x_{i,k}, x_k)) + b \tag{20}$$

2.3 FINDING SOLUTIONS

From Equations (8) or (17), we can find that the new LS-SVM is equivalent to solve a deficient rank linear equation set just like the over complete problem in ICA. Because the matrix A is $n \times nm$, there are infinite solutions to Equations (8) or (17). 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:

$$\min \left\|\vec{\beta}\right\|_1 . \tag{21}$$

Subject to

$$A \cdot \vec{\beta} = c\,, \tag{22}$$

where $\vec{\beta} = \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix}$,

$$\begin{cases} A = A_1, c = \begin{bmatrix} 0 \\ \vec{l} \end{bmatrix}, & \textit{for} \quad \textit{classifier} \\ A = A_2, c = \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix}, & \textit{for} \quad \textit{regression} \end{cases}.$$

It is a very important character that $e_i = \sum_{k=1}^{m}\alpha_{i,k} / \gamma$. Because b is a constant, the minimum of $\left\|\vec{\beta}\right\|_1$ is equivalent to that of $\left\|\vec{\alpha}\right\|_1$. From Equation (6), we can conclude that:

$$\left\|\vec{e}\right\|_1 = \sum_{i=1}^{n}\left|e_i\right| = \frac{\sum_{i=1}^{n}\left|\sum_{k=1}^{m}\alpha_{i,k}\right|}{\gamma} \le \frac{\left|\sum_{i=1}^{n}\sum_{k=1}^{m}\alpha_{i,k}\right|}{\gamma} = \frac{\left\|\vec{\alpha}\right\|_1}{\gamma}$$

So the minimum of $\left\|\vec{\alpha}\right\|_1$ can guarantee $\left\|\vec{e}\right\|_1$ in a lower level. 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 [13].

2.4 ALGORITHM

The procedures to implement the LS-SVM-L1 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 over complete forms.

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

3) Construct formulation: the new model based on the LS-SVM is constructed according to the Equation (8) or (17).

4) Solving: a linear programming with equality constrains ((21) and (22)) is solved to get the Lagrange coefficients α and b .

5) Output: then the output is got according to Equation (12) or (19).

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

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

3 Experiments analysis

To illustrate the effectiveness of our approach, we report results on three datasets taken from the UCI Machine learning Repository. These datasets are frequently used as benchmarks to compare the performance of different classification methods in the literature. The three disease datasets are the Wisconsin breast cancer dataset (WBCD), the heart disease dataset (HD) and the PIMA dataset. The first dataset is Wisconsin Diagnostic Breast Cancer dataset. It contains 699 records. Nine variables are used as the patients' characteristics. The second dataset is heart disease dataset, which contains 270 records. Thirteen variables are used as the patients' characteristics. The last dataset is PIMA dataset, which contains 768 records. Eight variables are used as the patients' characteristics.

The classification performance is measured by its specificity (T1), sensitivity (T2) and overall hit rate (T), 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.

$$overall \quad hit \quad rate(T) = \frac{TP+TN}{n}, \tag{23}$$

$$specificity(T1) = \frac{TP}{TP+FP}, \tag{24}$$

$$sensitivity(T2) = \frac{TN}{TN+FN}. \tag{25}$$

3.1 EXPERIMENTAL RESULTS ON THREE UCI DISEASE DATASETS

Firstly, the data is normalized. For the LS-SVM- L_1 classifier, the Gaussian kernel is used. So the kernel parameter and regularization parameter γ need to be chosen. Table 1 shows test set correctness, using the LS-SVM- L_1 with various parameters γ when σ^2 is equal to 1000, under ten-fold cross validation for the above mentioned dataset.

The column titled NSV is the number of support vectors selected from the training samples.

From Table 1, it can be seen that the number of selected support vectors is gradually increasing with the increase of γ. A bigger parameter γ makes the value of coefficient matrix decrease in constraint Equation (22) so that the constraint can be satisfied. As a result, the sparse of Lagrange $\alpha_{i,k}$ becomes bad. It tells us that only eight support vectors can achieve the best result: the average specificity is 96.57%, the average sensitivity is 99.48% and the average overall accuracy is 97.59%. This demonstrates that the proposed method is of good generalization ability.

The ten-fold experimental results of LS-SVM- L_1 using various kernel parameter are illustrated in Table 2 when γ is set to $\gamma = 2^5$. The change of kernel parameter has minor effect on degrading the classification accuracy when the parameter γ is fixed. This shows that our proposed method is very robust.

The Specificity (T1), Sensitivity (T2), overall hit rate (T), number of selected features for three datasets experimental results using LS-SVM- L_1 approach are shown in Table 3.

From Table 3, we can see that this model achieve good classification results using several features. It is shown that a reasonable feature extraction can improve the performance of the learning algorithm greatly. So this model is very simple and has inexpensive computation cost.

3.2 COMPARISON WITH OTHER CLASSIFIER MODELS

In order to further evaluate the effectiveness of the LS-SVM- L_1 model, the classification results are compared with some other methods using the same dataset, such as SVM and GA-based approach, in which the former methods cannot select features while the later can select features subset. The results of the GA-based models quoted from the reference [14]. Table 4 summarizes the T1, T2 and T accuracy of the three models. From Table 4, we can draw the conclusions as follows:

For the breast cancer dataset and PIMA, LS-SVM- L_1 model has the best accuracy classification capability in comparison with other three models. It correctly classifies 99.48% of ill instances and 96.57% of total ones for the WBCD. These results indicate that our method is very efficient in binary classification problem. But the GA-based approach only selects one feature, which decreases its classification accuracy. The GA-based model achieves the best classification accuracy and LS-SVM- L_1 achieves the second best results for the heart disease datasets.

In general, the LS-SVM- L_1 can provide efficient alternatives in conducting classification tasks.

TABLE 1 Ten-fold experimental results of LS-SVM- L_1 using various parameter γ ($\sigma^2 = 5000$)

Methods	NSV	T1	T2	Overall
$2^{0.1}$	2	96.32	99.48	97.32
2^{1}	2	96.32	99.48	97.32
2^{5}	3	96.57	99.48	97.5
2^{10}	5	96.08	97.38	96.49
2^{50}	6	45.84	88.48	73.29
2^{100}	6	50.49	62.31	54.26
2^{500}	6	50.49	62.31	54.26
2^{100}	7	50.49	62.31	54.26

TABLE 2 Ten-fold experimental results of LS-SVM- L_1 using different kernel parameter ($\gamma = 2^5$)

σ^2	NSV	Type1	Type2	Overall
1	5	93.63	67.54	85.31
10	6	89.22	95.29	91.15
50	6	92.89	98.95	94.82
100	7	96.08	97.38	96.49
500	7	96.32	98.95	97.16
1000	8	97.3	98.43	97.66
5000	2	96.57	99.48	97.59
10000	2	96.32	98.95	97.16

TABLE 3 experimental results using LS-SVM- L_1

Database	NSV	T1	T2	T
WBCD	2	99.48	96.57	97.59
HD	3	87.98	83.64	86.93
PIMA	3	82.51	75.166	78.02

TABLE 4 comparison with experimental results of three models

Method	Accuracy	WBCD	HD	PIMA
SVM	T1	30.72	33.33	5.33
	T2	100	80.52	95.34
	T	78.01	54.71	35.92
GA-based approach	T1	97.87	94.47	73.35
	T2	89.96	95.11	87.04
	T	96.19	95.3	74.2
	Number of selection features	1	5.4	3.7
LS-SVM-L_1	T1	99.48	87.98	82.51
	T2	96.57	83.64	75.16
	T	97.59	86.93	78.02
	Number of selection features	2	3	3

5 Conclusion

Unlike SVM and weighted LS-SVM, the LS-SVM- L_1 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- L_1 can be viewed as a weighted sum for different components. This makes the output more understandable.

Through the practical data experiment, we have obtained good classification results using selected features. And these show that LS-SVM- L_1 is of good performance in data classification. Thus the LS-SVM- L_1 provides efficient alternatives in conducting data classification tasks. Future studies will aim at finding the law existing in the parameters' setting. Generalizing the rules by the features that have been selected is another further work.

Acknowledgments

This research has been supported by a public benefit special fund from Quality inspection industry of China (#201210011).

References

[1] Vapnik V 1995 The Nature of Statistic Learning Theory *Springer-Verlag: New York* 69-71

[2] Vapnik V 1998 Statistic Learning Theory *Willey: New York* 125-9

[3] Scholkopf B, Smola A 2002 Learning with Kernels *MIT press: Cambridge* 21-30

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

[5] Suykens J A K, Gestel T V, Branbater J D, Moor B D, Wandewalle J 2002 Least squares support vector machines *World Scientific: Singapore* 167-70

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

[7] Suykens J A K, Lukas L, Wandewalle J 2000 Sparseness approximation using least squares support vector machines *IEEE international symposium on Circuits and System* **2** 57-760

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

[9] Jozsef V and Gabor H 2004 A sparse least squares support vector machine classifiers *IEEE international conference on neural network* **1** 543-8

[10] Hyvarinen A, Karhunen J and Oja E 2001 Independent Component Analysis *Willey: New York*131-5

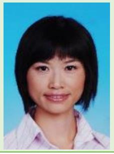
[11] Chen S S, Donoho D L, Sauders M A 2001 Atomic decomposition by basis pursuit *SIAM review* **43** 129-59

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

[13] Wei L W, Chen Z Y, Li J P 2011 Evolution Strategies Based Adaptive Lp LS-SVM *Information science* **181**(14) 3000-16

[14] Huang C L, Wang C J 2006 A GA-based feature selection and parameters optimization for support vector machines *Expert Systems with Applications* 31231-40

Authors



Liwei Wei, 23 January, 1981, Beijing, China.

Current position, grades: laboratory director of China National Institute of Standardization.
University studies: data mining, knowledge mining, big data analysis models.
Scientific interest: knowledge mining, big data analysis models.
Publications: 50 journal articles.

Hao Yu, 11 December, 1975, Beijing, China.

Current position, grades: IT center of China Agricultural University Library.
University studies: information technology.
Scientific interest: information technology and programming.
Publications: 20 journal articles.