Least Squares Twin Support Vector Machines Based on Sample Reduction for Hyperspectral Image Classification

Li-guo Wang[a], Ting-ting Lu[b] and Yue-shuang Yang[c]

College of Information and Communication Engineering, Harbin Engineering University, Harbin, 150001, China

[a]email: wangliguo@hrbeu.edu.cn, [b]email: a10260a@sina.com, [c]email: yangyueshuang2010@126.com.

Keywords: Least Squares Twin Support Vector Machine (LSTSVM); sample reduction; hyperspectral image.

Abstract. To overcome the low efficiency of Least Squares Twin Support Vector Machine (LSTSVM) in classifying, a new method called Sample Reduction LSTSVM (SR-LSTSVM) is proposed. The method greatly reduces the training samples and so improves the speed of LSTSVM, while the ability of LSTSVM to classify is unaffected. Our experiment results show remarkable improvement of the speed of LSTSVM on hyperspectral image, supporting our idea.

Introduction

The hyperspectral remote sensing technology, which appeared early in 1980s, has become an important technique in map cartography, vegetation investigation, ocean remote sensing, agriculture remote sensing, atmosphere research, environment monitoring and military information acquiring [1]. Support Vector Machine (SVM) [2] is an excellent kernel-based tool for Hyperspectral Image Classification [3] [4]. The central idea of SVM is to find the optimal separating hyperplane between the positive and negative examples. The optimal hyperplane is defined as the one giving maximum margin between the training examples that are closest to the hyperplane. In 2007, Jayadeva et al proposed Twin SVM (TSVM) [5]. TSVM generates two nonparallel hyperplanes by solving two smaller-sized QPPs such that each hyperplane is close to one class and as far as possible from the other. The strategy of solving two smaller-sized QPPs, rather than a single large-sized QPP, makes the TSVM is approximately four times faster than the usual SVM. TSVM has become one of the popular methods because of its low computational complexity. LSTSVM [6] is one of the variants of TSVM, it has better classification performance than TSVM and SVM [6] [7] [8]. Many variants of LSTSVM have been proposed, such as Knowledge based LSTSVM [9], Laplacian LSTSVM for semi-supervised classification [10], Weighted LSTSVM [11]. However, the low classification efficiency of LSTSVM has seldom been mentioned.

In this paper, a novel LSTSVM algorithm based on sample reduction is proposed. The experimental results on two data sets confirm the feasibility and effectiveness of the proposed method.

The remainder of this paper is organized as follows. In Section 2, we explain the method of LSTSVM and SR-LSTSVM, we discuss linear and nonlinear sample reduction methods respectively corresponding to linear and nonlinear LSTSVM. The experiments and their results are given in Section 3 and discussed in Section 4.

Methods

A. LSTSVM

LSTSVM is a useful extended of TSVM. It modified the primal QPPs of TSVM in least squares sense and solved them with equality constraints instead of inequalities of TSVM. As a result, the solution of LSTSVM follows directly from solving two systems of linear equations as opposed to

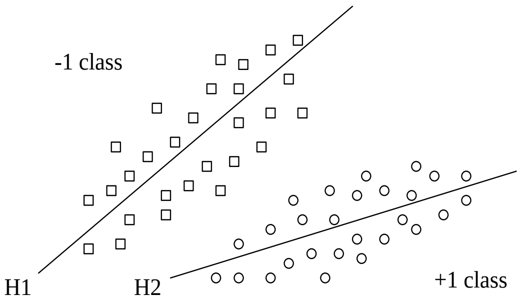


Fig.1 Geometric interpretation of LSTSVM

soloving two QPPs and two systems of linear equations in TSVM. Fig.1 shows the geometric interpretation of LSTSVM, where squares represent the -1 class samples, circles represent the +1 class samples, H1 represents the negative hyperplane and H2 represents the positive hyperplane.

Given m_1 datapoints in $\mathbf{R}^n$ of class +1 represented by matrix $\boldsymbol{A}$, and m_2 datapoints in $\mathbf{R}^n$ of class -1 represented by matrix $\boldsymbol{B}$**,** the two non-parallel hyperplanes of LSTSVM are:

$$\begin{aligned} &K(\boldsymbol{x}',\boldsymbol{C}')\boldsymbol{u}^1 + r^1 = 0 \\ &K(\boldsymbol{x}',\boldsymbol{C}')\boldsymbol{u}^2 + r^2 = 0 \end{aligned} \qquad (1)$$

where $\boldsymbol{x} \in \mathbf{R}^n$ is the testing sample, $\boldsymbol{x}'$ is the transformed of $\boldsymbol{x}$, $\boldsymbol{C} = [\boldsymbol{A} \;\; \boldsymbol{B}]$ and K is any arbitrary kernel. In this paper, Gauss radial basis kernel function is used, it’s form is shown in (2).

$$K(\boldsymbol{x},\boldsymbol{y}) = \exp[-\|\boldsymbol{x}-\boldsymbol{y}\|^2 / 2\sigma^2] \qquad (2)$$

The primal QPPs of nonlinear TSVM can be modified in the same way with 2-norm of slack variables and inequality constraints replaced by equality constraints:

$$\begin{aligned} &\underset{\boldsymbol{u}^1,r^1}{Min}\frac{1}{2}\|(K(\boldsymbol{A},\boldsymbol{C}')\boldsymbol{u}^1 + \boldsymbol{e}r^1)\|^2 + \frac{C_1}{2}\boldsymbol{y}'\boldsymbol{y} \\ &\text{s.t } \text{-}(K(\boldsymbol{B},\boldsymbol{C}')\boldsymbol{u}^1 + \boldsymbol{e}r^1)\text{+}\boldsymbol{y}\text{=}\boldsymbol{e} \\ &\underset{\boldsymbol{u}^2,r^2}{Min}\frac{1}{2}\|(K(\boldsymbol{B},\boldsymbol{C}')\boldsymbol{u}^2 + \boldsymbol{e}r^2)\|^2 + \frac{C_2}{2}\boldsymbol{y}'\boldsymbol{y} \\ &\text{s.t } (K(\boldsymbol{A},\boldsymbol{C}')\boldsymbol{u}^2 + \boldsymbol{e}r^2)\text{+}\boldsymbol{y}\text{=}\boldsymbol{e} \end{aligned} \qquad (3)$$

By substituting the constraints into objective function, these QPPs become:

$$\begin{aligned} &\underset{\boldsymbol{u}^1,r^1}{Min}\frac{1}{2}\|(K(\boldsymbol{A},\boldsymbol{C}')\boldsymbol{u}^1 + \boldsymbol{e}r^1)\|^2 + \frac{C_1}{2}\|K(\boldsymbol{B},\boldsymbol{C}')\boldsymbol{u}^1 + \boldsymbol{e}r^1 + \boldsymbol{e}\|^2 \\ &\underset{\boldsymbol{u}^2,r^2}{Min}\frac{1}{2}\|(K(\boldsymbol{B},\boldsymbol{C}')\boldsymbol{u}^2 + \boldsymbol{e}r^2)\|^2 + \frac{C_2}{2}\|-(K(\boldsymbol{A},\boldsymbol{C}')\boldsymbol{u}^2 - \boldsymbol{e}r^2 + \boldsymbol{e})\|^2 \end{aligned} \qquad (4)$$

The solution of QPPs (4) can be derived to be:

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{u}^1 \\ r^1 \end{bmatrix} = -(\boldsymbol{H}'\boldsymbol{H} + \frac{1}{C_1}\boldsymbol{G}'\boldsymbol{G})^{-1}\boldsymbol{H}'\boldsymbol{e} \\ &\begin{bmatrix} \boldsymbol{u}^2 \\ r^2 \end{bmatrix} = (\boldsymbol{G}'\boldsymbol{G} + \frac{1}{C_2}\boldsymbol{H}'\boldsymbol{H})^{-1}\boldsymbol{G}'\boldsymbol{e} \end{aligned} \qquad (5)$$

Where $\boldsymbol{G}\text{=}[K(\boldsymbol{A},\boldsymbol{C}') \;\; \boldsymbol{e}], \boldsymbol{H}\text{=}[K(\boldsymbol{B},\boldsymbol{C}') \;\; \boldsymbol{e}]$.

Once the parameters of two hyperplanes (5): $\boldsymbol{u}^1$, r^1, $\boldsymbol{u}^2$, r^2 were obtained, a new datapoint $\boldsymbol{x}$ to +1 or -1 is classified based on which perpendicular distance is minimum:

Class $i=\underset{i=1,2}{Min}\left|K(\boldsymbol{x}',\boldsymbol{C}')\boldsymbol{u}^i+r^i\right|$

(6)

B. Sample Reduction LSTSVM

As mentioned earlier, LSTSVM generates two nonparallel hyperplanes such that each hyperplane is close to one class and as far as possible from the other. Take the positive hyperplane for example, it's determined by only a small subset of the whole negative training data which is close to positive samples. Thus LSTSVM classification time can be considerably reduced by removing the central regional's negative training samples. The training sample reduction algorithm will not have any impact on LSTSVM accuracy but it will speed up the LSTSVM testing process. For this purpose, a new sample reduction LSTSVM based on 1-against-rest [12] multi classifier is proposed.

In terms of linear and nonlinear separable cases, our method falls into linear and nonlinear sample reduction method respectively.

a. Linear sample reduction method

Definition Center: a center is defined as the average features of examples for certain class. Given training examples $\{x_1, x_2, ..., x_n\}$, then the center is expressed [13]:

$$\boldsymbol{m}=\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i \qquad (7)$$

Definition Distance: Distance is meant the difference of feature between two examples. Given two examples $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, the distance can be expressed:

$$d(\boldsymbol{x}_1,\boldsymbol{x}_2)=\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2=\sqrt{\sum_{i=1}^{N}(\boldsymbol{x}_1^{\ i}-\boldsymbol{x}_2^{\ i})^2} \qquad (8)$$

Definition Center-Distance: the distance from $\boldsymbol{x}$ to $\boldsymbol{m}_x$:

$$d_{\boldsymbol{xx}}(\boldsymbol{x},m_x)=\sqrt{\sum_{i=1}^{N}(\boldsymbol{x}^i-m_x^{\ i})^2} \qquad (9)$$

$\{x_1, x_2, ..., x_n\}$ represent the +1 class training samples set, and $\{\boldsymbol{x}'_1, \boldsymbol{x}'_2, ..., \boldsymbol{x}'_n\}$ represent the -1 class training samples set, only the negative training samples are reduced considering the number of negative training samples is much more than the positive, our algorithm proceeds as follows:

Step1: give the pre reduction rate *r*, calculate the center-distance of the negative samples according to equation (9);

Step2: sort the negative training samples in ascending order according to the distance calculated in step1;

Step3: remove the negative training samples that have small center-distance according the pre reduction rate;

Step4: set the remaining negative training samples and the original positive training samples as LSTSVM new training set.

b. Nonlinear sample reduction method:

In the nonlinear case, we map the training samples into some other feature space by using a nonlinear mapping ϕ. The distance between two points $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ in feature space *H* is:

$$d^H(\boldsymbol{z}_1,\boldsymbol{z}_2)=\sqrt{K(\boldsymbol{z}_1,\boldsymbol{z}_1)-2K(\boldsymbol{z}_1,\boldsymbol{z}_2)+K(\boldsymbol{z}_2,\boldsymbol{z}_2)} \qquad (10)$$

Note that the center in feature space $\boldsymbol{m}_\phi$ becomes:

$$\boldsymbol{m}_\phi=\frac{1}{n}\sum_{i=1}^{n}\phi(\boldsymbol{x}_i) \qquad (11)$$

Where *n* is the number of examples.

The center-distance in feature space is:

$$d^{H}{}_{xx}(\boldsymbol{x},\boldsymbol{m}_{\phi})=\sqrt{K(\boldsymbol{x},\boldsymbol{x})-\frac{2}{n}\sum_{i=1}^{n}K(\boldsymbol{x},\boldsymbol{x}_{i})+\frac{1}{n^{2}}\sum_{i=1}^{n}\sum_{j=1}^{n}K(\boldsymbol{x}_{i},\boldsymbol{x}_{j})} \tag{12}$$

Since the mapped examples are linear separable in feature space, therefore the sample reduction method in the feature space is the same as the linear case mentioned in a.

C. Algorithm Complement

Now we briefly describe our algorithm SR-LSTSVM as follows:

(1). Pre-reducing negative training samples by calculation distance;

(2). Optimizing by LSTSVM, and completing classification.

Test results

A. Data

1) Indian Pine: The Indian Pine hyperspectral dataset covers a forest/agricultural site in Indiana and was obtained by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor. The hyperspectral image has a spatial size of 144 × 144 pixels. Twenty spectral bands were removed due to the noise and water absorption, and the remaining 200 spectral dimensions were used. Five classes with large numbers of samples in the original reference map were selected for experiments, including corn-min, grass/pasture, grass/trees, soybeans-notill, and woods. We randomly split them into the training set and testing set (5 percent for training, 95 percent for testing).

2) Pavia University: The University of Pavia dataset was obtained by the reflective optics system imaging spectrometer (ROSIS) sensor during a flight campaign over the Engineering School at University of Pavia. Water absorption bands affected by the atmosphere were removed, and the original 115 bands were reduced to 103 bands. An area with a size of 144×144 pixels and five classes is used for test, including metal sheets, soil, bitumen, bricks, and shadows. We randomly split them into the training set and testing set (2 percent for training, 98 percent for testing).

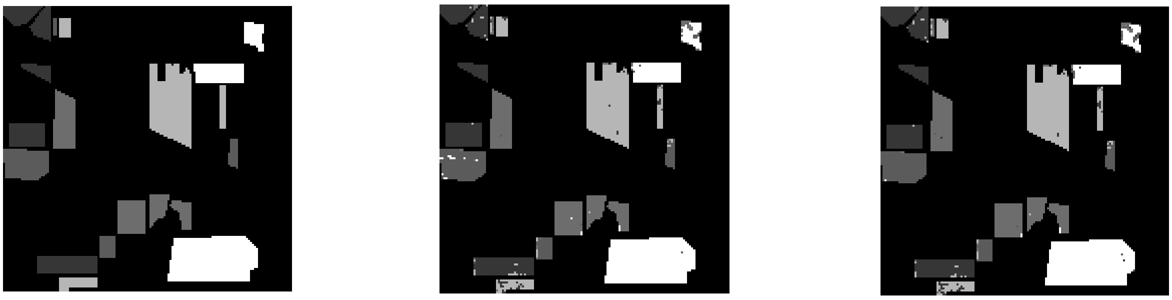
B. Parameter Settings

A grid search was adopted to find proper *C* and σ within sets of {0.1, 1, 10, 20, 60, 100, 160} and {10^{-2}, 10^{-1}, 10^{0}, 10} respectively, and pre-reducing rate *r* is 50%. We repeat the experiment 10 times, and the classification time is the average of ten values.

C. Results

The experimental results on Indian Pine dataset are shown in Figure 2 and Table 1, and the Pavia University dataset results are shown in Figure 3 and Table2.

As shown in Table 1 and Table 2, the classification time of SR-LSTSVM on Indian Pine dataset and Pavia University dataset are substantially shortened by 42% and 46% compared to LSTSVM. While the overall classification accuracy are not affected as shown in Figure 2 and Figure 3.



(a) Reference land-cover map (b) LSTSVM (c) SR-LSTSVM

Fig.2 Indian Pine dataset results.

Table 1 Comparison on India Pine dataset

Algorithm	C1	C2	C3	C4	C5	OA	Kappa	Time(s)
LSTSVM	96.28%	91.35%	98.80%	95.45%	97.06%	96.20%	0.9514	56.7868
SR-LSTSVM	96.88%	93.56%	98.80%	95.14%	96.06%	96.20%	0.9514	**33.1871**

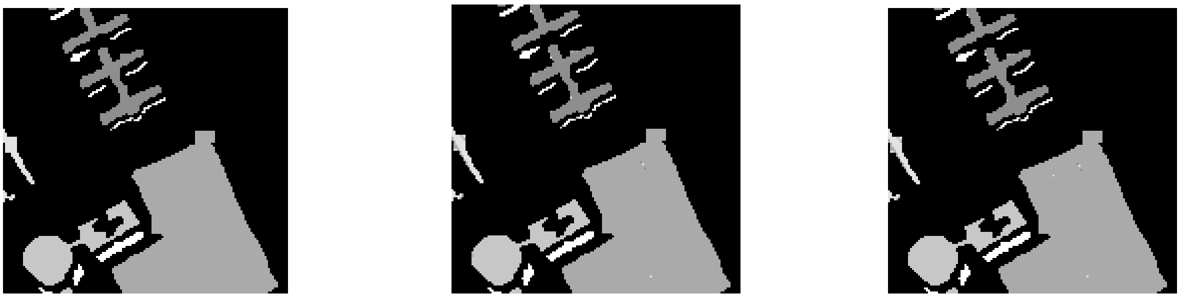

(a) Reference land-cover map (b) LSTSVM (c) SR-LSTSVM

Fig.3 Pavia University dataset results.

Table 2 Comparison on Pavia University dataset

Algorithm	C1	C2	C3	C4	C5	OA	Kappa	Time(s)
LSTSVM	98.47%	99.90%	99.75%	79.84%	99.65%	99.26%	0.9853	34.3838
SR-LSTSVM	98.61%	99.88%	99.87%	80.62%	99.65%	99.29%	0.9860	**18.6853**

Figure 4 represents the influence of pre-reducing rate on classification time in Indian Pine dataset, we can know that, when *r* is in the range of 90%~70%, the classification accuracy increased from 74.59% to 91.66% dramatically; the classification accuracy reaches the maximum when *r* is 50%; when the pre reduction ratio is less than 50%, the classification accuracy is almost the same. When the *r* is in the range of 0%~90%, the classification time decrease approximately linearly

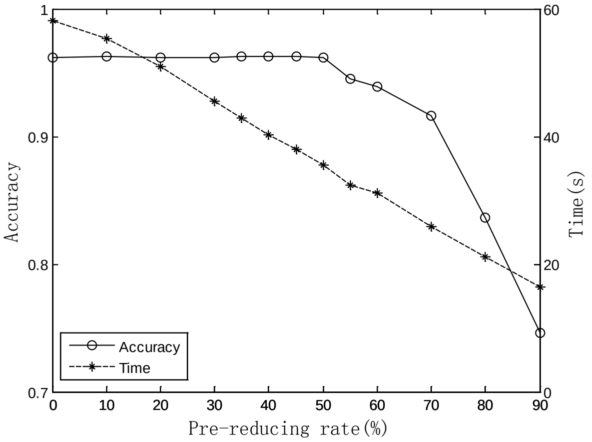


Fig.4 The influence of pre-reducing rate(*r*) on classification time in Indian Pine dataset

These results lead us to conclude that it is no difference between the classification accuracy of SR-LSTSVM and LSTSVM. But it is important to stress that the classification speed of SR-LSTSVM is much faster than that of LSTSVM.

Conclusion

In this paper we present a new method for pre-deducting training samples to improve the speed of LSTSVM. The method can extract the negative samples which are important for the positive decision surface. We can not only assure the ability of LSTSVM to classify patterns, but also enormously improve the speed of LSTSVM by using this method. The results of experiments show the validity and feasibility of the method.

Acknowledgement

In this paper, the research was sponsored by the National Natural Science Foundation of China (Project No. 61275010), the Ph.D. Programs Foundation of Ministry of Education of China (Project No. 20132304110007), the Nature Science Foundation of Heilongjiang Province (Project No. F201409) and the Central University Basic Scientific Research Business Expenses Major Project Financing Plan (Project No. HEUCFD1410).

References

[1] Wang Li-guo, Zhao Chun-hui. Processing Techniques of Hyperspectral Imagery[M]. Beijing: National Defense Industry Press, 2013, 6-7.

[2] Cortes C, Vapnik V. Support Vector Networks[J], Machine Learning, 1995, 20(3): 273-297.

[3] Tan Kun, DU Pei-jun, Hyperspectral Remote Sensing Image Classification Based on Support Vector Machine[J], Journal of Infrared and Millimeter Waves, 2008, 27(2): 123-128.

[4] WANG Yang-ting, Remote Sensing Image Automatic Classification with Support Vector Machine[J], Computer Simulation, 2013, 30(6): 378-385.

[5] Jayadeva, Khemchandani R, Twin Support Vector Machines for Pattern Classification[J], IEEE Transactions on Pattern Analysis and machine intelligence, 2007, 29(5): 905-910.

[6] Arun Kumar M, Gopal M , Least squares twin support vector machines for pattern classification[J], Expert Systems with Applications, 2009, 36(4): 7535–7543.

[7] PENG Xin-jun, Least squares twin support vector hypersphere (LS-TSVH) for pattern recognition[J], Expert Systems with Applications, 2010, 37: 8371–8378.

[8] CHU Mao-xiang, WANG An-na, GONG Rong-fen. Improvement on Least Squares Twin Support Vector Machine for Pattern Classification[J]. Acta Electronica Sinica, 2014, 42(5): 998-1003.

[9] M. Arun Kumar, Reshma Khemchandani, M. Gopal , et al, Knowledge based Least Squares Twin support vector machines[J], Information Sciences, 2010, 180: 4606–4618.

[10] Wei-Jie Chen, Yuan-Hai Shao, Nai-Yang Deng, et al, Laplacian least squares twin support vector machine for semi-supervised classification[J], Neurocomputing, 2014, 145: 465–476.

[11] MU Xiao-xia, LI Jun-tao, and CHEN Liu-yuan, Classification with Noise via Weighted Least Squares Twin Support Vector Machine[J], Computer Simulation, 2014, 31(5): 288-292.

[12] WANG Hai-yan, LI Jian-hui, YANG Feng-lei. Overview of support vector machine analysis and algorithm[J]. Application Research of Computers, 2014, 31(5): 1281-1286.

[13] Zhang Li, Zhou Weida and Jiao Licheng, Pre- extracting Support Vectors for Support Vector Machine[C], In 2000 5th international conference on Signal Processing Proceedings, Beijing, 2000, 1432-1435.