ISBN 978-981-14-1684-2
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
(WCSE 2019 SUMMER)
HongKong, *15-17 June*, *2019*, *pp. 65-71*
doi: 10.18178/wcse.2019.06.011

Lost-Min Voting Strategies for Speeding up Multi-SVMs

Shinq-Jen Wu[+, 1], Van-Hung Pham[2]

[1] Da-Yeh University, Department of Electrical Engineering, Chang-Hwa, Taiwan.
[2] Institute information of technology, Vietnam academy of Science and Technology, 18 Hoang Quoc Viet, Hanoi, Vietnam

Abstract. Support vector machines (SVMs) possess good accuracy in big data classification. However, the computational cost in both training and testing stages is a critical issue. The authors recently proposed a two-phase sequential minimal optimization to largely reduce the training cost (tested with 3186–70,000-sample datasets). The authors now focus on speeding up the testing speed of SVMs for multi-class classification. A lost-min strategy is proposed to accelerate the voting algorithm used in multi-SVMs. The number of the used binary classifiers is reduced from an order of n^2 to n (nearly to $n-1$). The proposed lost-min voting strategy was tested with DNA dataset (bioinformatics), Usps datasets (handwritten digits), Letter dataset (English alphabet) and Satimage dataset (satellite imagery of Earth). The time complexity for all of the datasets approaches to $n-1$ algorithm and the accuracy is remained at the same time.

Keywords: Support vector machines, multi-class classification, big data analysis, computational biology.

1. Introduction

SVMs become a standard classification technique in a wide of fields such as visual category reorganization [1], spoof fingerprint detection [2], typhoon rainfall forecast [3] and diabetic retinopathy [4]. The supervised learning algorithm is originally designed to deal with two-target data: For a two-class problem, SVMs are used to classify m-dimentional instances $\mathrm{x_i} \in \mathrm{R^m}, \mathrm{i} = 1, \dots, \mathrm{l}$ into categories $\mathrm{y_i} \in \{1 \quad 1\}$. However, in the real world researchers always have to deal with a large number of classes. A easy way is to divide a multiclassification problem into many binary classification problems. Various approaches were proposed to achieve efficient multi classificaton [5, 6, 7]. The one-aganinst-one (OAO) with a voting strategy [8] and one-against-rest (OAR or one-against-all) with a winer-take-all strategy [9] are two popular methods for doing this dividation. The number of classification for one-against-rest methods is $\mathrm{n} - 1$ which is much less than that of one-against-one approach, $(\mathrm{n}(\mathrm{n} - 1)/2)$. However, in the training stage the data used for OAR is the entire dataset, but only two-class subsets are used in OAO. Additionally, OAR generates a data imbalance issue and further data processing are required [10, 11].

OAO with voting strategies are used in LIBSVM (a SVM software proposed by Chang and Lin [12]). In this paper a lost-min-voting-based strategy is proposed to improve the performance of the one-against-one approach. The authors here name the proposed method as lost-min one-against-one method (lmOAO) which is able to largely accelerate the testing speed of OAO. Four datasets in various fields (DNA, Satimage, USPS and Letter) are used to test the performance of the proposed lmOAO. Simulation results show that the complexity approximates to $n-1$ instead of O(n^2).

[+] Corresponding author. Tel. 886-4-8511888
E-mail address: jen@mail.dyu.edu.tw

This study is organized as follows. In Section 2, a lost-min-voting-based multi-class SVMs is presented. Section 3 is experimental tests for DNA, Usps, Letter and Satimage datasets. Section 5 is the conclusive remark and future works.

2. Lost-Min-Voting-Based Svms

SVMs are essentially a binary classification. Therefore, for data with many classes it is intuitional to divide a multi-classification problem into many binary classification sub-problems. In the training stage SVMs construct a hyperplane as a decision surface in such a way that the margin of the separation between positive and negative examples is maximized [13]. *The optimal hyperplane is denoted by the identified support vector* x^s(the respective class $y^s = 1$ o -1): $g(x^s) = w_o^T x^s + b_o = \pm 1$, where the optimal weight vector w_o and the optimal bias b_o are related to the Lagrange multipliers of the support vector x^s .

2.1 SVMs Training

Thang and coworkers pointed out that SVMs are considerably slower in testing phase than other approaches with similar classification performance because of using a large number of support vectors [14]. Therefore, for a fair comparison the authors do not use previously proposed two-phase sequential minimal optimization [15]. Another sequential minimal optimization is proposed for training SVMs, wherein the number of the estimated support vectors is comparable to that of the general voting strategy used in LibSVM.

The used sequential minimal optimization is based on the following working set selection (named mWSS, a modification of the working set selection in LibSVM). Figure 1 describes the scheme of the proposed working set selection (mWSS), wherein mWSS1, mWSS2 and mWSS3 converge when they reach the following stop condition:

$$-y_i \nabla f(\boldsymbol{\alpha})_i + y_j \nabla f(\boldsymbol{\alpha})_j < \rho, \tag{1}$$

where ρ=0.001 in this study. mWSS is an integration of the modified WSS1 (mWSS1), the modified WSS2 (mWSS2) and the modified WSS3 (mWSS3). The original WSS1, WSS2 and WSS3 choose the working set from $_{\text{up}}(\alpha)$ and $_{\text{ow}}(\alpha)$.

$$\begin{aligned} \mathrm{I}_{\mathbf{up}}(\boldsymbol{\alpha}) &\equiv \{t | \alpha_t < C, y_t = 1 \quad or \quad \alpha_t > 0, y_t = -1\}, \\ \mathrm{I}_{\mathbf{low}}(\boldsymbol{\alpha}) &\equiv \{t | \alpha_t < C, y_t = -1 \quad or \quad \alpha_t > 0, y_t = 1\}. \end{aligned} \tag{2}$$

Their respective modifications choose the i and j from different subsets which are listed as follows.

$$\begin{aligned} \mathrm{I}_{\mathbf{upBound}}(a) &\equiv \{t | \alpha_t = 0, y_t = 1 \quad or \quad \alpha_t = C, y_t = -1\}, \\ \mathrm{I}_{\mathbf{lowBound}}(a) &\equiv \{t | \alpha_t = 0, y_t = -1 \quad or \quad \alpha_t = C, y_t = 1\}, \\ \mathrm{I}_{\mathbf{inter}}(a) &\equiv \{t | 0 < \alpha_t < C\}. \end{aligned} \tag{3}$$

The same mathematical operations are used for the orgianl methods and the modified methods, except the working set selection. The maximal violating pair B={i,j} for these three modifiations (mWSS1, mWSS2 and mWSS3) are chosen as follows.

mWSS1 (modified WSS1[16])

Select i, $i \in \arg\max_t \{-y_t \nabla f(\boldsymbol{\alpha})_t | t \in \mathrm{I}_{\text{up}}(\boldsymbol{\alpha})\}$

Select j, $j \in \arg\min_t \{-y_t \nabla f(\boldsymbol{\alpha})_t | t \in \mathrm{I}_{\text{lowBound}}(\boldsymbol{\alpha})\}$

Return $\boldsymbol{B} = \{i,j\}$

mWSS2 (modified WSS2[17])

Select I, $i \in \arg\max_t \{-y_t \nabla f(\boldsymbol{\alpha})_t | t \in \mathrm{I}_{\text{upBound}}(\boldsymbol{\alpha})\}$

Select j, $j \in \arg\min_t \{-y_t \nabla f(\boldsymbol{\alpha})_t | t \in \mathrm{I}_{\text{low}}(\boldsymbol{\alpha})\}$

Return $\boldsymbol{B}=\{i,j\}$

mWSS3 (modified WSS3[17])

Select I, $i \in \arg\max_t \{-y_t \nabla f(\boldsymbol{\alpha})_t \mid t \in \mathrm{I}_{\mathrm{inter}}(\boldsymbol{\alpha})\}$

Select j, $j \in \arg\min_t \{-y_t \nabla f(\boldsymbol{\alpha})_t \mid t \in \mathrm{I}_{\mathrm{inter}}(\boldsymbol{\alpha})\}$

Return $\boldsymbol{B}=\{i,j\}$

Both WSS2 and WSS3 were widely used in LibSVM. These two methods and their respective modifications (mWSS2 and mWSS3) use the second-order information to hasten the convergence [17].

Algorithms start with mWSS1 and then mWSS2. If the converge criterion is not reached then go back to mWSS1. The repeated process is taken over and over again until mWSS2 converge to a threshold. After that algorithms initiate mWSS3 operation. The training stage stops at the time that mWSS3 converges to the threshold ρ.

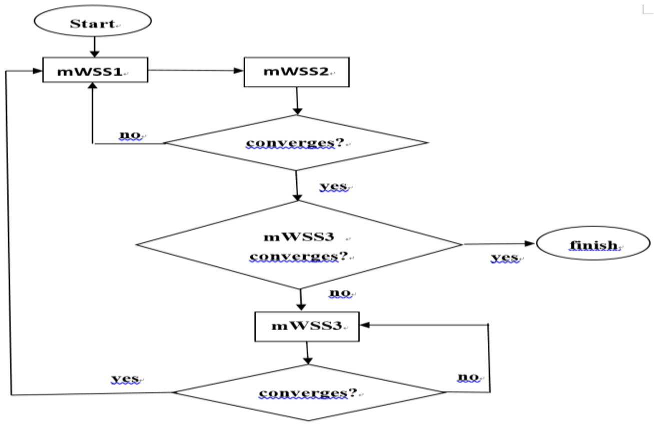

Fig. 1: Modified working set selection for training.

2.2 Lost-min one-against-one SVMs testing

A lost-min strategy is proposed to accelerate the testing speed of OAO. Figure 2 describes the scheme for lost-min one-against-one.

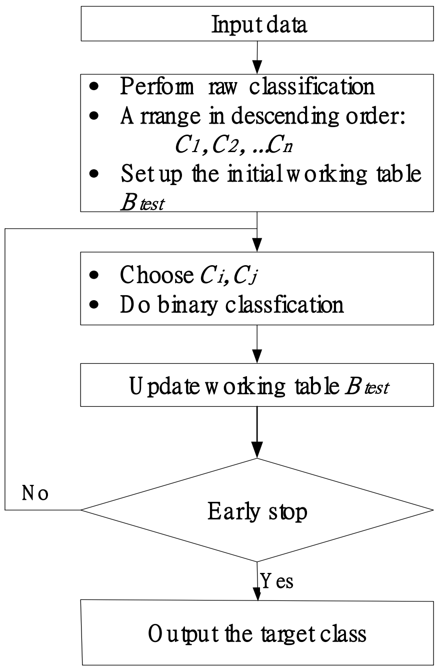

Fig. 2: The lmOAO scheme

The classification speed of SVMs with linear kernel functions is much faster than that of SVMs with nonlinear kernel functions, even the accuracy of the former is less than the latter. In this study linear kernel functions were used for training SVMs and getting a rough classification for testing. The classification function is defined as follows. (the class of $\mathrm{x} = \arg\max_{\mathrm{i}} \mathrm{f}_{\mathrm{c}_{\mathrm{i}}}(\mathrm{x})$ for the max-win voting strategy [18].)

$$f_{c_i}(x) = \sum_{j,j\neq i} sign(f_{ij}(x)), \quad i = 1, \dots n,$$

$$f_{ij}(x) = \sum_{r=1}^{h} \alpha_{ijr} < x, x_{ijr}^{s} > = < x, \sum_{r=1}^{h} \alpha_{ijr} x_{ijr}^{s} >$$

$$= < x, \sum_{r=1}^{h} \alpha_{ijr} x_{ijr}^{s} > = < x, y_{ij}^{s} >, \quad (4)$$

where the hyper function for ij-classifier is $y_{ij}^{s} = \sum_{r=1}^{h} \alpha_{ijr} x_{ijr}^{s}$, $x_{ijr}^{s}$is the estimated support vectors from the training stage and $\alpha_{ijr}$is the respective Lagrange parameter. For each input x the voting function $f_{c_i}(x)$ is estimated and the respective classes are arranged in a descending order:$C_1, C_2, \dots C_3$ according to the value of $f_{c_i}(x)$. The raw classification result is used to set up an initial working table which is denoted as $\boldsymbol{B_{test}}$. Table 1 shows an initial working set table. Black color denotes initial chracters or values. The maximal score max_{c_i} is the maximal possibility of the data x belong to class Ci, which is initially set at n-1. The current score cur_{c_i} is initially set at zero.

The rule for choosing working set $\boldsymbol{B_{test}}$ **is** to choose the largest values of max_{c_i} and cur_{c_i}, and the priority is $max_{c_i} > cur_{c_i}$, (The notation >denotes superior). At first time both maximal score and current score are the same. Therefore, for the data *x* the binary classification pair C_1 and C_2 are chosen firstly because these two possess the first two highest $f_{c_i}(x)$. If the classfication result is C_1 then the respective maximal score of the failed class C_2 is reduced by 1 and the current score of the winner C_1 is incremented by 1, as shown in red characters and digits. Then, C_1 and C_3 classification pair are chosen by the rule. If the result is C_1 then the respective maximal score of the failed class C_3 is reduced by 1 and the current score of the winner C_1 is incremented by 1, as shown in purple characters and digits.

Table 1. Working set table $\boldsymbol{B_{test}}$

	C_1	C_2	C_3	…	C_i	…	C_n
C_1	-	C_1	$\boldsymbol{C_1}$				
C_2	-	-					
⋮	-	-	-				
C_j	-	-	-	-			
⋮	-	-	-	-	-		
C_{n-1}	-	-	-	-	-	-	
C_n	-	-	-	-	-	-	-
max_{c_i}	n-1	n-1-1	n-1-**1**	n-1	n-1	n-1	n-1
cur_{c_i}	0+1+**1**	0	0	0	0	0	0

The process goes over and over again until a target class is reached. A early stop criterion is set at the time that the current score of that class is not less than the maximal score of all of the other classes. Figure 3 describes the conditions for lsOAO stops after (n-1)- or (n-2)-time binary classifications.

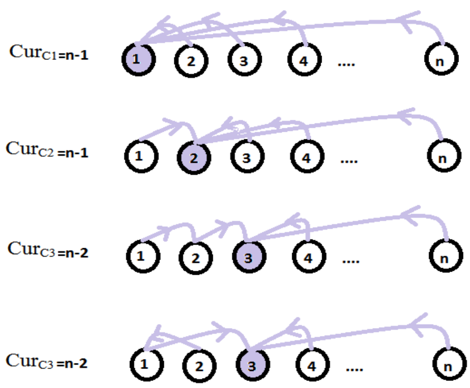

Fig. 3: The case that lmOAO stops at $(n-1)$- or $(n-22)$-time binary classfications

3. Experiments

The proposed method was tested with English alphabet (**Letter** dataset), **Usps** dataset (handwritten ZIP codes extracted from digital images of handwritten addresses), **DNA** dataset in Stalog version (bioinformation) and **Satimage** in Stalog version (satellite images). Table 2 lists the numbers of training and testing samples (instances), the number of features (attributes, genes), the number of classes for those datasets, the training results for kernel parameters and the testing accuracy. All datasets were downloaded from LIBSVM website http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html. The used kernal parameters C and γ, as shown in Table 2, are obtained through grid serching. The accuracy is 95.53% for DNA dataset, 92.15% for Satimage dataset, 95.81% for Usps dataset and 97.58% for Letter dataset.

Table 2 : Data information and experimental results shown in average values. ($\epsilon = 0.001$)

Datasets	features no.	Class no	Training samples no.	Testing samples no.	Kernel parameters		testing accuracy
					C	γ	
DNA	180	3	2,000	1,186	100	0.008	95.53%
Satimage	36	6	4,435	2,000	100	1.7	92.15%
Usps	256	10	7,291	2,007	150	0.15	95.81%
Letter	16	26	15,000	5,000	1000	1.0	97.58%

Figure 4 is a comparion of the used support vectors of these four datasets for the origianl SVMs and the proposed method. Table 3 lists the detailed information for these two approaches. The number of the used support vectors is (488, 1065, 757, 1733) for (DNA, Satimage, Usps, Letter) when lmOAO-based voting strategy is used, and the number is (611, 1463, 1758, 5340) when the origianl voting method is used.

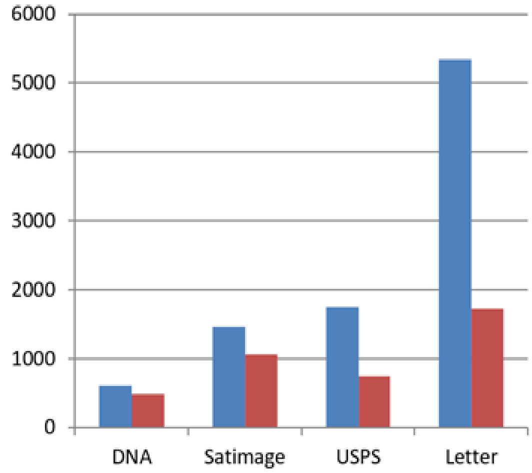

Fig. 4: Comparison of the proposed method (red) to original one (blue) in the number of the used support vectors.

Table 3: Comparison of the used support vectors

Dataset	#Class	The average number of the used support vectors	
		lmOAO-based voting	original voting
DNA	3	488	611
Satimage	6	1065	1463
Usps	10	757	1758
Letter	26	1733	5340

Figure 5 shows the number of kernel estimation in WSS1, WSS3 and the proposed mWSS. A large reduction rate is observed for all of the datasets. (An extra Mnist dataset is used for comparison. The testing result of Mnist is simliar to that of Usps.)

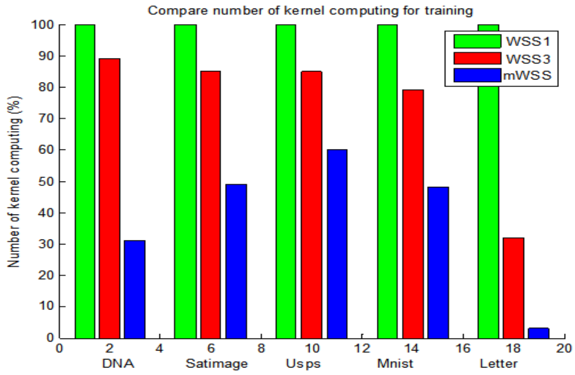

Fig. 5: Comparison of the proposed mWSS to WSS1 [16] and WSS3 [17].

Additionally, the complexity of the proposed method is discussed. Table 4 is a comparison of the proposed method to the voting methods. The used number of binary classifiers of the proposed method is close to that of the n-1 algorithm. A clear comparion is shown in Fig. 6 which demonstrated that the perormance is largely improved as the number of data is increased.

Table 4: Comparsion in the number of binary classifications.

Dataset	*n*	Average number of binary classify performance		
		proposed algorithm	Voting algorithm	(n-1) algorithm
DNA	3	2.00	3	2
Satimage	6	5.01	15	5
USPS	10	9.03	45	9
Letter	26	25.39	325	25

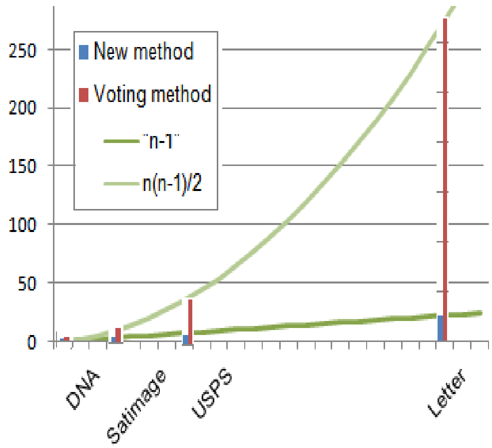

Fig. 6: Comparison in the number of binary classification.

4. Conclusions

SVMs' t aining is to identify a decision bounda y (a hype -plane) that separates a feature space into two halves. The hype p ane is a weighted summation of suppo t vecto s. SVMs' testing is to distinguish the c ass of the input data according to the identified support vectors in the training stage. The authors first use a hyper function y_{ij}^{s} to simplify the estimation of both kernel and voting functions. A lost-min strategy is then proposed to accelerate the voting and get the target class as soon as possible. Experimental results show that the proposed methods largely reduce the complexity of the voting algorithm (from $\frac{n(n-1)}{2}$ to near $n-1$ algorithm). This technology has been embedded in an automatic data entry system (VnHandwritten 1.0). In the future the authors shall focus on developing both training and testing technologies to deal with real imbalance data.

5. Acknowledgments

This research was supported by grant number MOST 107-2221-E-212-013 from the Ministry of Science and Technology of Taiwan, R.O.C.

6. References

[1] Chang, X., Yu, Y. L., and Yang, Y. 2017. Robust top-k multiclass SVM for visual category recognition. In *Proceeding of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canad, August 13 – 17, 2017).

[2] Kho, J. B, Lee, W., Choi H., and Kim, J. (2019). An incremental learning method for spoof fingerprint detection. *Expert Systems with Applications*. 116 (2019), 52-64.

[3] Yang, T. C., Yu, P. S., Lin, K. H., Kuo, C. M., and Tseng, H. W. (2018). Predictor selection method for the construction of support vector machine (SVM)-based typhoon rainfall forecasting models using a non-dominated sorting genetic algorithm. *Meteorological Applications*. 25, 4 (2018), 510-522.

[4] Karthikeyan, R., and Alli, P. 2018. Feature selection and parameters optimization of support vector machines based on hybrid glowworm swarm optimization for classification of diabetic retinopathy. *Journal of Medical Systems*. 42, 10 (2018) 195.

[5] Alber, M., Zimmert, J., Dogan, U., and Kloft, M. 2017. Distributed optimization of multi-class SVMs. *PLoS ONE*. 12, 6 (2017), e0178161.

[6] Xu, J., Liu, X., Huo, Z., Deng, C., Nie, F., and Hunang, H. 2017. Multi-class support vector machine via maximizing multi-class margins. In *Proceedings of the twenty-sixth International Joint Conference on Artificial Intelligence*.

[7] Kumar, M. K., and Gopal, M. 2010. Fast multiclass SVM classification using decision tree based one-against-all method. *Neural Processing Lett*ers. 32, 3 (December 2010), 311–323. https://doi.org/10.1007/s11063-010-9160-y.

[8] Chmielnicki, W., and Stapor, K. 2016. Using the one-versus-rest strategy with samples balancing to improve pairwise coupling classification. *Int. J. Appl. Math. Comput. Sci.* 26, 1 (2016), 191–201.

[9] Kreßel, U. 1999. Pai wise c assification and suppo t vecto machines. In *Advances in Kernel Methods: Support Vector Learnings.* Cambridge, MA, MIT Press, 255-268.

[10] Chmielnicki, W., and Stapor, K. 2016. Using the one-versus-rest strategy with samples balancing to improve pairwise coupling classification. *Int. J. Appl. Math. Comput. Sci.* 26, 1 (2016), 191–201.

[11] Beyan, C., and Fisher, R., Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition*. 48, 5 (2015), 1653–1672

[12] Chang, C. C., and Lin, C. J. 2011. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol*. 2 (2011), 27:1--27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[13] Vapnik, V. 1998. Statistical Learning Theory. Wiley, New York.

[14] Thang, P. Q., Lam, H. T., and Thur, N. T. (2018) Improving simplification of support vector machine for classification. *International Journal of Machine Learning and Computing*, 8, 4 (2018),372-376.

[15] Wu, S. J., Pham, V. H., and Nguyen, T. N. (2017). Two-phase optimization for support vectors and parameter selection of support vector machines: two-class classification. *Applied Soft Computing.* 59 (2017), 129-142.

[16] Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., and Murthy, K. R. K. 2001. mp ovements to p att's SMO algorithm for SVM classifier design. *Neural Comput*. 13, 3 (2001), 637–649.

[17] Fan, R. E., Chen, P. H., and Lin, C. J. 2005. Working set selection using second order information for training support vector machines. *J Mach Learn Res*. 6 (2005), 1889-1918.

[18] Mustaqeem, A., Anwar, S. M., and Majid, M. 2018. Multiclass classification of cardiac arrhythmia using improved feature selection and SVM invariants. *Computational and Mathematical Methods in Medicine*. Volume 2018, Article ID 7310496, 10 pages.