Zhang Chen[1], Xia Shixiong[1], Liu Bing[1]

[1]School of Computer Science and Technology, China University of Mining and Technology, Xu Zhou, 221116, China

Maximum Margin Clustering Using Extreme Learning Machine

Abstract. *Maximum margin clustering (MMC) is a newly proposed clustering method, which extends large margin computation of support vector machine (SVM) to unsupervised learning. But in nonlinear cases, time complexity is still high. Since extreme learning machine (ELM) has achieved similar generalization performance at much faster learning speed than traditional SVM and LS-SVM, we propose an extreme maximum margin clustering (EMMC) algorithm based on ELM. It can perform well in nonlinear cases. Moreover, the kernel parameters of EMMC need not be tuned by means of random feature mappings. Experimental results on several real-world data sets show that EMMC performs better than traditional MMC methods, especially in handling large scale data sets.*

Streszczenie. *Opisano nową metodę klastrowania* ***„maximum margin clusterung MMC" która rozszerza wielkość marginesu obliczeń numerycznych w systemie SVM z uczeniem bez nadzoru. Nowa metoda EMMC (extreme maximum margin clustering) zapewnia szybsze uczenie, szczególnie w warunkach nieliniowości. (Nowa metoda klastrowania – extreme margin clustering EMC w systemach extreme learning machine ELM)***

Keywords: Maximum margin clustering, unsupervised learning, extreme learning machine (ELM), random feature mapping
Słowa kluczowe: SVM – support vector machine, klastrowanie, uczenie bez nadzoru

Introduction

Our goal is to design a fast maximum margin clustering algorithm used to improve the conventional high computational complexity of MMC. This approach is formulated as a sequence of efficient ELM training. Firstly, we reformulate the MMC problem based on ELM as a nonconvex optimization problem, and then perform alternating optimization directly on the constructed nonconvex problem instead of relaxing it. Our key modification is to replace SVM or SVR by ELM with the square loss, which can not only speed up the MMC algorithm, but discourage premature convergence. Thus, compared to existing approaches, the proposed EMMC in fact involves only a sequence of ELM training and the resultant implementation is fast and scales well. Experimental evaluations on several real-world data sets show that EMMC performs better than existing MMC methods.

Organization of the paper is as follows. Section 2 reviews prior research. Section 3 outlines the proposed method .Experimental results on several real-world data sets are provided in Section 4.Section 5 concludes the paper.

Review of previous research

Maximum margin clustering (MMC) is a newly proposed clustering method by means of supervised learning method. The key idea of MMC is to extend the maximum margin principle of support vector machines (SVM) to the unsupervised learning scenario. Hence the MMC technique often obtains more accurate results than conventional clustering methods. As the labels of samples are unknown, optimization over all the possible labeling leads to a hard, non-convex integer optimization problem. Consequently, different optimization techniques have been used to relax the original problem. Xu et al. [1] reformulate it as a semidefinite programming (SDP) problem, which could be efficiently solved using standard SDP solvers such as SeDuMi [2] and SDPT3[3]. Valizadegan and Jin [4] further proposed the generalized MMC (GMMC) algorithm which reduces the number of parameters in the SDP formulation form n^2 to n, where n is the number of samples. Unfortunately, due to the fact that solving SDP is still computationally expensive, the worst-case time complexity of MMC and GMMC is $O(n^{6.5})$ and $O(n^{4.5})$ respectively. Zhang et al. [5] utilized the alternative optimization techniques to solve the MMC problem. But how to make MMC applicable to a large scale data set is a very challenging and valuable research topic.

Recently, ELM has been attracting considerable interests from more and more researchers [6-9]. The idea of ELM is actually the same to that of the random vector functional-link (RVFL) network [10, 11]where the hidden neurons are randomly selected and only the weights of the output layer need to be trained. Hence, ELM can be regarded as the single-hidden-layer RVFL network. The relatively fast convergence rate and small approximation error can be guaranteed if the number of hidden nodes is large enough, which is meaningful to large scale data sets. ELM provides a unified solution to different practical applications (e.g., regression, binary, and multiclass classifications), while different variants of LS-SVM and SVM are required for different types of applications, so the application of ELM is much easier.

Approach

In this section, we will firstly reformulate the MMC problem based on ELM with single output, and then solve the constructed nonconvex MMC problem by means of alternating optimization. Computationally, this allows the nonconvex problem to be formulated as a sequence of ELM training, so the proposed algorithm is fast and effective. Finally, we extend EMMC with single output to the multi-outputs scenario.

Since ELM can approximate any target continuous functions, the output of the ELM classifier $h(x)\boldsymbol{\beta}$ can be close to the class labels in the corresponding regions as possible. Thus the classification problem for the ELM with a single-output node can be formulated as [5]:

$$\min_{\beta,\xi_i} \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{n}\xi_i^2, s.t. h(x_i)\beta = t_i - \xi_i, i = 1,...,n \quad (1)$$

where $h(x) = [h_1(x),...,h_L(x)]$ is the output (row) vector of the hidden layer with respect to the input $\boldsymbol{x}$. Thus the corresponding MMC problem is

$$\min_{t,\beta,\xi_i} \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{n}\xi_i^2, s.t. h(x_i)\beta = t_i - \xi_i, i = 1,...,n \quad (2)$$

where $t_i \in \{\pm 1\}$ for two-class clustering with the class balance constraint $-l \le \sum_{i=1}^{n} t_i \le l$ or $t_i \in \{1,...,m\}$ for m-classes clustering with the class balance constraint $-l \le N_p - N_q \le l$, where m is the number of classes, $p, q \in \{1,...,m\}$, N_p and N_q are the number of samples in the pth and qth class, respectively.

A natural way to solve (2) is to use a simple iterative approach based on alternating optimization [5]. This is similar to the Iterative SVR proposed in [12]. First, we fix *t* and minimize (2) w.r.t. β, which is just a standard ELM training. Then, we fix β and minimize (2) w.r.t. *t*. Specifically, we discuss the following problem without the class balance constraint.

(3) $$\min \sum_{i=1}^{n} (h(x_i)\beta - t_i)^2$$

$s.t. t_i \in \{\pm 1\}$ **for two-cla**ss clustering or $t_i \in \{1,...,m\}$ for *m*-classes clustering. $i = 1,...,n$.

Proposition 1: For two-class clustering, the optimal strategy to determine t_i s in (3) is to assign all t_i s as -1 for those with $h(x_i)\beta \le 0$, and assign t_i 's as 1 for those with $h(x_i)\beta > 0$; For muticlass clustering, the optimal strategy to determine t_i s in (16) is to assign all t_i 's as *i* for those with $i-1 < h(x_i)\beta \le i$, $i \in \{1,...,m\}$, where *m* is the number of classes.

The proof of Proposition 1 is similar to that of the Iterative SVR proposed in [12], we don't discuss it further.

If ELM has multioutput nodes, an *m*-class classifier is corresponding to *m* output nodes. If the original class label is *l*, the expected output vector of the *m* output nodes is $\mathbf{t}_i = [0,...,0,\overset{l}{1},0,...,0]^T$. That is, the *l*th element of $t_i = \{t_{i1},...,t_{im}\}^T$ is one, while the rest of the elements are set to zero. The classification problem for ELM with multioutput nodes can be formulated as [9]

(4) $$\min_{\beta,\xi_i} \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{n}\xi_i^2$$
$$s.t. h(x_i)\beta = t_i^T - \xi_i^T, i = 1,...,n,$$

where $t_i = \{t_{i1},...,t_{im}\}^T$ is the training error vector of the *m* output nodes with respect to the training sample x_i. The corresponding MMC problem is

(5) $$\min_{t,\beta,\xi_i} \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{n}\xi_i^2$$
$$s.t. h(x_i)\beta = t_i^T - \xi_i^T, i = 1,...,n,$$

$t_i = \{t_{i1},...,t_{im}\}^T$ the *w*th element is one and the rest of the elements are set to zero, $w \in \{1,...,m\}$ where *m* is the number of classes, $-l \le N_p - N_q \le l$, $p,q \in \{1,...,m\}$, N_p and N_q are the number of samples in the *p*th and *q*th class, respectively.

Solving Eq. (5) by the alternative optimization method and enforcing the class balance constraint are similar to those of EMMC based on ELM with the single output. The difference is that the output function of ELM with multioutputs is the function vector, Thus we first compute $h(x_i)\beta$, and then assign the labels according to the distance between $h(x_i)\beta$ and t_i. Finally, we sort the $\max_i f(x_i)(1 \le i \le m)$ and reassign the labels to enforce the class balance constraint.

For the sake of clarity, the complete algorithm is summarized：

Step 1: Initialize the labels t

Step2: For two-class clustering, fix t, where $t_i \in \{\pm 1\}$ and perform training of ELM with single output. For muticlass clustering, fix t, where $t_i \in \{1,...,m\}$ and perform training of ELM with single output, or fix t , where $t_i = \{t_{i1},...,t_{im}\}^T$ and perform training of ELM with multi-outputs.

Step3：Assign the labels as described above.

Step4: Check the class balance constraint, if it is violated, sort the $h(x_i)\beta$'s and reassign the labels as described above.

Step5: Repeat steps 2–4 until convergence.

Hence, EMMC with single output has comparable performance to that based on multioutputs, which will be validated in Section 4.

Results

In this section, we will validate the performance of the proposed EMMC algorithm on a number of real-world data sets. We use seven data sets from the UCI machine learning repository. The same experimental setup was set as in [5].

Firstly, we study the effect of initialization on EMMC with single output. The two initialization schemes are included in the experiment:1)random; 2)standard *k*-means clustering (KM). As can be seen from Table 1, the clustering error of the random scheme is close to 50% error, EMMC with random initialization has poor performance with the poor initialization.

As can be seen from Table 2 and 3, the clustering accuracy of EMMC with single output is slightly lower than that of EMMC with multi-outputs, Thus, for simplicity, we use the EMMC algorithm with single output in both two-class and multiclass clustering, and then compare it with the other MMC algorithms.

Table 1. Average Performance on the 45 Clustering Tasks Under Different Initialization Schemes

Clustering Scheme	Clustering error(%)	CPU time(Second)
Random only	48.21	0.001
Random + EMMC	26.37	11.06
KM only	3.49	0.025
KM + EMMC	1.84	1.69

Table2. Clustering Results Comparisons between EMMC with single output and multiouputs on LetterABCD data set.

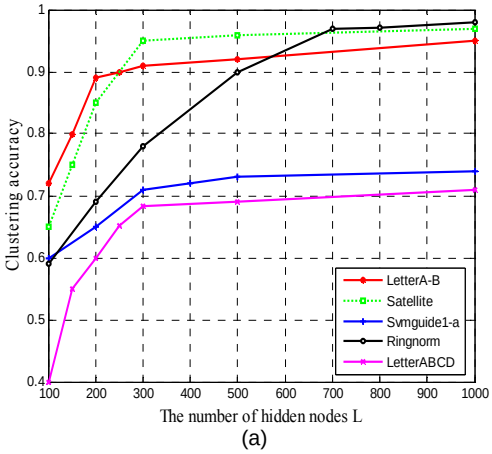
The number of hidden nodes L	EMMC with single output		EMMC with multioutpus	
	Acc(%)	Time(s)	Acc(%)	Time(s)
100	40.17	3.36	40.86	3.79
150	56.35	4.70	56.25	5.58
200	60.74	6.27	61.44	7.36
300	68.25	13.32	68.73	14.86
500	69.67	19.68	70.85	22.52
1000	71.53	103.76	71.72	109.69

Table 3. Clustering Results Comparisons between EMMC with single output and multiouputs on USPS data set.

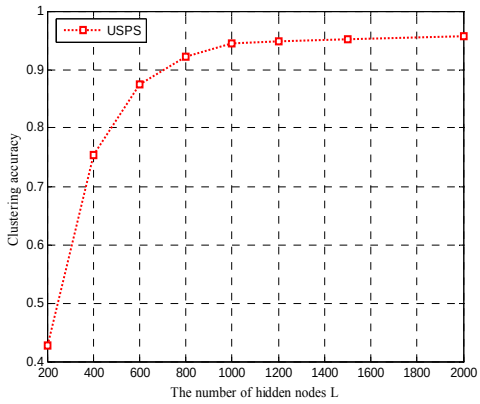
The number of hidden nodes *L*	EMMC with single output		EMMC with multioutpus	
	Acc(%)	Time(s)	Acc(%)	Time(s)
200	42.65	30.51	41.31	32.62
400	75.29	87.90	75.54	92.11
600	87.38	190.56	87.74	196.885
800	92.26	342.21	92.45	350.875
1000	94.47	536.61	94.89	547.045
1200	94.75	742.05	95.03	754.705
1500	95.11	1173.06	95.57	1188.76

We further perform EMMC with different numbers of hidden nodes on several data sets, whose size is bigger than 1000. Fig. 1 shows the clustering accuracy of EMMC with various values for *L*. It can be seen from Fig. 1

In Fig.2 the CPU time of EMMC grows nonlinearly with the increase of the variable *L*. From Fig.1 and Fig.2, it is not difficult to find out that letting *L* equal 300 is suitable for letterA-B, satelliteC1-C2, svmguide1-a and letterABCD. For ringnorm and USPS, we let *L* equal 300 and 1000, respectively.

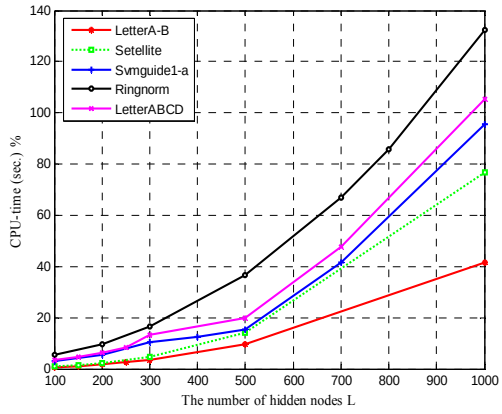


(a)

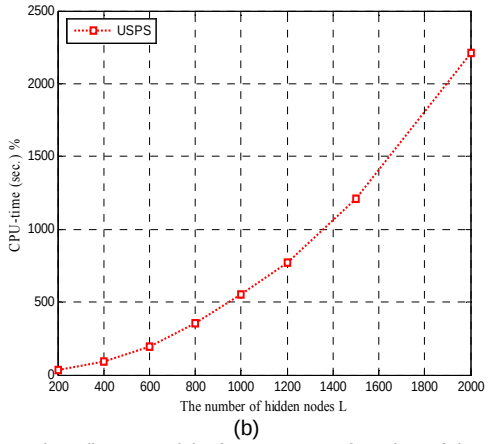


(b)

Fig.1. Clustering accuracy of EMMC with various values for L. (a) several data sets. (b) USPS.



(a)



(b)

Fig.2. CPU time (in seconds) of EMMC as a function of the number of hidden nodes. (a) several data sets. (b) USPS.

Overall, EMMC can handle not only two-class but multiclass problems, and has good clustering performance at the fastest speed.

Conclusion

In this paper, we propose an efficient approach for solving MMC via ELM. While traditional MMC algorithms are formulated as SDPs or based on the SVM model, our approach is formulated as a sequence of efficient ELM training. Meanwhile, the symmetric square loss function in ELM discourages premature convergence by penalizing overconfident predictions. It is also noted that our method can handle imbalanced data effectively by enforcing the class balance constraint. Empirically, the clustering performance of EMMC is comparable to that of the other MMC algorithms. Moreover, it is much faster and can handle much larger data sets. In the future, we will study how to extend our clustering method to the semi-supervised learning setting. In addition, in order to enhance the performance of EMMC further, we will combine kernel learning methods with our methods.

REFERENCES

[1] L. Xu, J. Neufeld, B. Larson, D. Schuurmans, Maximum margin clustering, Advances in neural information processing systems, 17 (2004) 1537-1544.

[2] J.F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization methods and software, 11 (1999) 625-653.

[3] K.C. Toh, M.J. Todd, R.H. Tütüncü, SDPT3—a MATLAB software package for semidefinite programming, version 1.3, Optimization methods and software, 11 (1999) 545-581.

[4] H. Valizadegan, R. Jin, Generalized maximum margin clustering and unsupervised kernel learning, Advances in Neural Information Processing Systems, 19 (2007) 1417.

[5] K. Zhang, I.W. Tsang, J.T. Kwok, Maximum margin clustering made practical, Neural Networks, IEEE Transactions on, 20 (2009) 583-596.

[6] G.B. Huang, K. Mao, C.K. Siew, D.S. Huang, Fast modular network implementation for support vector machines, Neural Networks, IEEE Transactions on, 16 (2005) 1651-1663.

[7] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, Neural Networks, IEEE Transactions on, 13 (2002) 415-425.

[8] Q. Liu, Q. He, Z. Shi, Extreme support vector machine classifier, Lecture Notes in Computer Science, 5012 (2008) 222-233.

[9] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, (2010) 1-17.

[10] Y.H. Pao, G.H. Park, D.J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing, 6 (1994) 163-180.

[11] L.P. Wang, C.R. Wan, Comments on “The Extreme Learning Machine”, Neural Networks, IEEE Transactions on, 19 (2008) 1494-1495.

[12] L. Xu, D. Wilkinson, F. Southey, D. Schuurmans, Discriminative unsupervised learning of structured predictors, in:Poceedings of the 23rd international conference on Machine learning ,2006, pp. 1057-1064.

***Authors**: Chen Zhang, School of Computer Science and Technology, China University of Mining and Technology, No.1 Daxue Road, 221116,Xuzhou,China,E-mail:zc@cumt.edu.cn ; Shi-xiong Xia, School of Computer Science and Technology, China University of Mining and Technology, No.1 Daxue Road, 221116,Xuzhou,China,Email:xiasx@cumt.edu.cn ; Bing Liu, School of Computer Science and Technology, China University of Mining and Technology, No.1 Daxue Road, 221116,Xuzhou,China,E-mail:liubing@cumt.edu.cn*