Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009

A NOVEL LEARNING MODEL-KERNEL GRANULAR SUPPORT VECTOR MACHINE

HU-SHENG GUO, WEN-JIAN WANG, CHANG-QIAN MEN

School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China
Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Taiyuan 030006, China
E-MAIL: chaofei142@163.com , wjwang@sxu.edu.cn , menchangqian@sxu.edu.cn

Abstract:

This paper presents a novel machine learning model-Kernel Granular Support Vector Machine (KGSVM), which combines traditional support vector machine (SVM) with granular computing theory. By dividing granules and replacing with them in kernel space, the datasets can be reduced effectively without changing data distribution. And then the generalization performance and training efficiency of SVM can be improved. Simulation results on UCI datasets demonstrate that KGSVM is highly scalable for large datasets and very effective in terms of classification.

Keywords:

Kernel granular support vector machine; Support vector machine; Kernel space; Granules

1. Introduction

Support vector machine introduced by Vapnik [1] is a kind of universal and effective tool for solving pattern recognition and regression problems, like handwritten recognition [2], face image recognition [3], time series prediction [4], et al. At present, SVM has become a research hotspot of machine learning. The crux of SVM design is solving a convex quadratic programming (QP) problem with linear constrains, which depends on the training vectors and selection of kernel function and relative parameters. The solution of an QP problem provides the necessary information for choosing the most important vectors known as support vectors (SVs) among all the datasets, and these support vectors will play important roles in defining the discriminant hyperplane or predicting function. The solution of the QP problem can be obtained by gradient projection algorithm, reduced gradient algorithm, active set algorithm, interior point algorithm, et al. And it also can be obtained straightly by using standard QP packages such as MINOS, CPLEX, LOQO and QP ROUTINE provided in Matlab toolbox.

For a small training datasets, these methods are effective. However, with the great increase of training data in size, the memory space for storing the kernel matrix will increase with the level O(N^2), where N is the number of the training data. This indicates that the techniques mentioned above may be unsuitable for large size problems. Some approaches based on iteration or decomposition strategies for solving QP problems have been widely used to improve SVM training [5,10]. Although these approaches, to a certain extent, can decrease the size or the degree of difficulty for an QP problem, the high training cost for saving memory space must be endured [11,12].

As we know, the traditional SVM algorithm for large scale datasets is not efficient and up to now no rules of kernel function selection can guarantee the "linear separability" to any problem. Besides, it is prone to be affected by noisy samples or features. To improve the performance of traditional SVM, granular support vector machine (GSVM) is proposed through combining statistical learning theory and granular computing theory [13]. It works by building a sequence of information granules and then building SVM in some of information granules on demand. In general, GSVM works in three steps. Step 1 is building a sequence of information granules in the original dataset space. Step 2 is modeling SVM in some of these information granules when necessary. Finally, it aggregates information in these granules at suitable abstract level. This method can not only achieve good generalization for a linear separable classification problem, but also increase "linear separability" for a linear non-separable problem (or even transfer a linear non-separable problem to totally linear separable). Comparing with traditional SVM, this method may be more possible to grasp inherent data distribution by trade-off between local significance of a subset of data and global correlation among different subsets of data. Besides, GSVM can speed up the modeling

978-1-4244-3703-0/09/$25.00 ©2009 IEEE

process by eliminating redundant data locally. For many real world data mining applications, what people expect is not only to get a model with small prediction error, but also to explain the reason why it works so well. As we know, SVM is almost unable to provide this kind of information. But a few critical rules or cases can be extracted from information granules so that GSVM decision process is similar to human understandable rule-based reasoning or case-based reasoning. By this new SVM model, the speed of SVM training can be greatly improved and the satisfactory generalization performance can be obtained as well.

Generally, GSVM training is in high-dimensional space and it divides granules and replaces data with granules in low-dimensional space. So the inherent distribution feature after data replacement with granules may not be reflected well in high dimension feature space., and then the prediction function may not be appropriate. This paper presents a kernel granular support vector machine algorithm, which firstly maps the original data into a high-dimensional space, and then divides granules by some strategies like clustering, neural network, decision tree or rough set so as to obtain high-dimensional granules. At last, SVM learning is accomplished by them. The proposed KGSVM algorithm can solve the problems with different distributions in low-dimensional space and high-dimensional space. Therefore, the proposed KGSVM can largely improve the generalization performance of SVM.

2. Kernel Granular Support Vector Machine

Suppose the given dataset is $X=\{x_1,x_2,\cdots,x_n\}$ and $x_i \in R^m$, the goal of granulation is find a mapping $X \to \{X_1,X_2,\cdots,X_k\}$. k is the number of granules, $X_i=\{x_{1,}\cdots,x_{N_i}\} \in R^{N_i} \times R^m$, N_i is the number of samples in i th granule.

To specify kernel granular support vector machine algorithm, we give the following definition firstly.

Definition 1. In m -dimensional space, each divided granule based on Euclidean distance can be a m -dimensional superball.
The centre of this superball is

$$\mu=\frac{1}{N_i}\sum_{j=1}^{N_i}x_j \tag{1}$$

and the radius of this superball is

$$r=\max\left|x_i-\mu\right|,1\le i\le N_i \tag{2}$$

Suppose Φ is nonlinear mapping from low-dimensional space to high-dimensional space H. k granules divided in M -dimensional space H could obtain k M -dimensional superballs. The centre and the radius could be computed by formulas (3) and (4):

$$\mu=\frac{1}{N_i}\sqrt{\sum_{\substack{j=1\\k=1}}^{N_i}K(x_i,x_j)} \tag{3}$$

$$r=\max\sqrt{K(x_i,x_j)-\frac{2}{N_i}\sum_{j=1}^{N_i}K(x_i,x_j)+\frac{1}{N_i^2}\sum_{\substack{j=1\\k=1}}^{N_i}K(x_i,x_j)} \tag{4}$$

Comparing with the traditional SVM algorithm, the training speed of GSVM algorithm may be faster, but the accuracy may be declined. The main reason is the replacement of granules. Usually, when the centre of superball in low-dimensional space can represent perfectly the samples belonging to this superball, but it may not represent those samples belonging to the superball in high dimension space (see Fig.1).

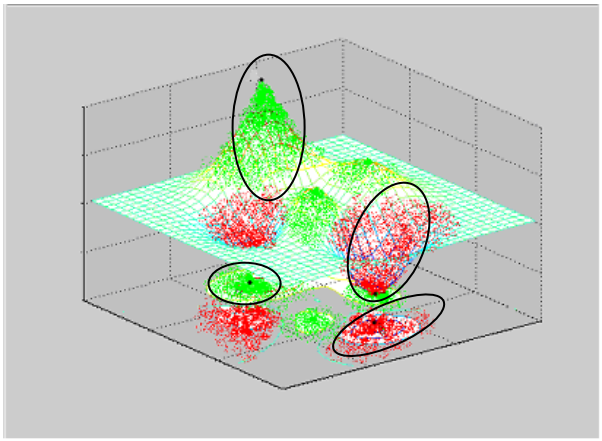


Figure 1. The difference of samples distribution in high-dimensional space

At first, we analyze the impacts of dividing granules and replacing with them in high-dimensional space shown in Fig.2.

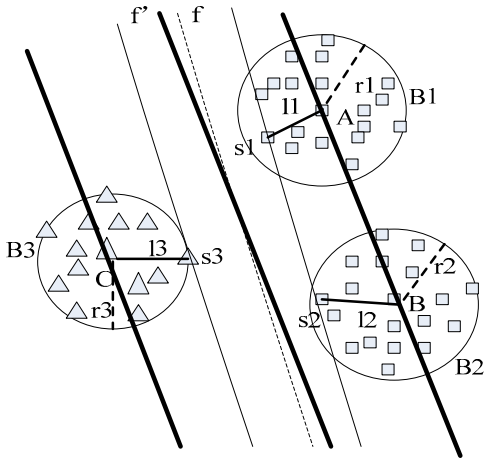


Figure 2. Granules dividing in high-dimensional space by KGSVM algorithm

As shown in Fig.2, the triangles refer to mapped negative samples, the squares are positive ones. Suppose hyperplane f is obtained by traditional SVM training, f' is obtained by KGSVM training, and s1, s2, s3 are support vectors. In high dimension space H, three M-dimensional superballs B1, B2 and B3 can be obtained by dividing granules. Then replacement with granules is implemented by formula (3). The positive sample A replaces all the samples in the superball B1, the positive sample B replace all the samples in the superball B2, and negative sample C replaces all the samples in the superball B3. After that, taking the centres of these superballs as the training set, original support vectors s1, s2 and s3 may not appear in the new training set. Then we could compute the radius of superballs r1, r2 and r3 by formula (4). Suppose that the distance from the centre A of superball to s1 is $l1$, the distance from the centre B of superball to s2 is $l2$, and the distance from the centre C of superball to s3 is $l3$. When the size of sample is very large, we could think that the data distribution in the spherical shell is relatively uniform, and the distance from any dot of the spherical shell to the core of superball is same. That means that $r1 \approx l1, r2 \approx l2$ and $r3 \approx l3$.

Now, let's observe the differences between the hyperplanes by training A, B, C and by training s1, s2, s3.

(1) Deflection of hypherplane

When $l1 >> l2$ or $l2 >> l1$ (see Fig.3). the hyperplane after KGSVM algorithm training will appear a large angle α comparing with that after traditional SVM algorithm training.

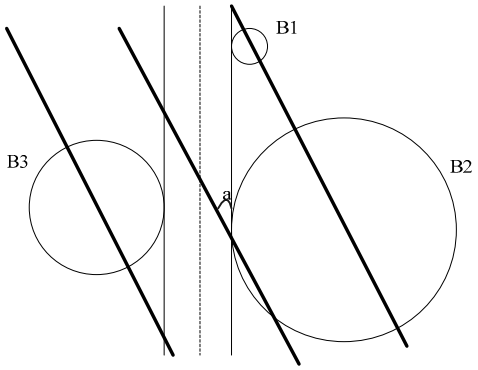


Figure 3. The deflection of hypherplane (the categories of B1, B2 and B3 are same to Fig.2)

As mentioned above, we could think $r1 \approx l1$, $r2 \approx l2$. Then the hyperplane will lead to the deviation of the corner when the radius of superballs belonging to the same categories in high-dimensional space H are different.

(2) Shift of hyperplane

When $l1 \approx l2 >> l3$ or $l3 >> l1 \approx l2$, that is $r1 \approx r2 >> r3$ or $r3 >> r1 \approx r2$ (see Fig.4). The hyperplane after KGSVM algorithm training will be shifted with b comparing with the original SVM training. Therefore, when the radius of different types of samples are different, the hyperplane after training will result in a certain shift.

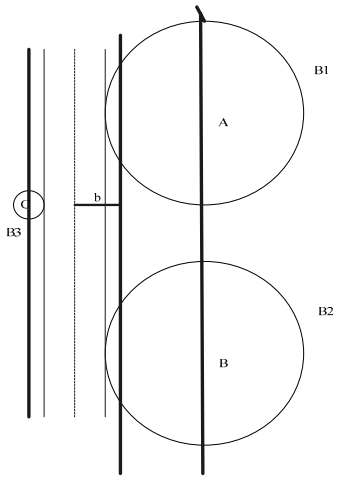


Figure 4. The shift of hyperplane (the categories of B1, B2 and B3 are same to Fig.2)

Based on the above analysis, an important rule for

dividing granules in KGSVM algorithm is given.

Rule 1 KGSVM_R Rule (The Key Rule of KGSVM Algorithm Dividing Granules)

In KGSVM algorithm, the radius of different superballs after dividing granules should be in a small interval [$r_{\min}$, $r_{\max}$].

For dividing granules and replacing with them in high-dimensional space, the size of the superball itself may not affect the hyperplane. Even if $r1 \approx r2 \approx r3$ and they are relatively large, the errors generated from the replacement of granules (that is the radius of superballs) will counteract each other and the final hyperplane is hardly changed. However, when the radius of the superball are different in despite of whether these superballs belonging to the same category, the training may be inefficient. Therefore, the radius of superballs granules should be consistent with the KGSVM_R Rule so as to make learning be more effective.

For the sake of simplicity, this paper adopts kernel clustering to divide granules. KGSVM algorithm is summarized as follows:

Step1: Initializing the number of granules k and the thresholds $r_{\min}$ and $r_{\max}$ of KGSVM_R Rule.

Step2: Mapping original samples $X = \{x_1, x_2, \cdots, x_n\}$ into the high-dimensional space H by a kernel function.

Step3: Dividing granules in high-dimensional space H by using the kernel clustering method and make each granule satisfy KGSVM_R Rule.

Step4: Taking the centers of all superballs including the training set $X' = (x_1, x_2, \cdots, x_{n'})$ ($n' < n$), which will contain the information as more as possible and then replace the original training samples with X'. In so doing, it can not only compress the original datasets, but also remove the noise effectively.

Step5: Training SVM in X'.

3. Simulation Results and Discussions

In order to verify the effectiveness of the KGSVM algorithm, six binary classification data from UCI data mining repository are used in this paper, and they are listed in Table 1. The popular Gaussian kernel with σ=1.0 is used, and C=100.

Table 1. Datasets used in experiments

Dataset	# training data	# testing data	dimension
Banana	8800	1000	2
Breast_cancer	2000	770	9
Diabetis	4680	3000	8
Heart	4250	2500	13
Thyroid	2800	1500	5
Titanic	750	10255	3

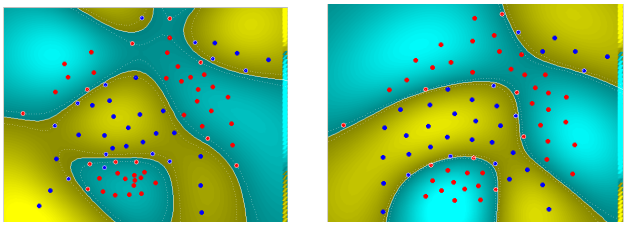
For KGSVM algorithm, the performance mainly depends on the kernel granule parameter (here, refers to kernel clustering parameter) k. Firstly, the impact of parameter k on the algorithm is analyzed. Table 2 compares the testing results between the KGSVM algorithm and GSVM algorithm. Here, we do not consider the traditional SVM algorithm due to its inefficient on certain size datasets by general QP solving approaches.

Table 2. The comparisons between two algorithms on Banana dataset

k	Testing Accuracy(%)		Training Time(s)	
	KGSVM	GSVM	KGSVM	GSVM
20	85.7	78.1	0.156	0.016
40	83.9	79.2	0.109	0.094
60	85.3	84.9	0.171	0.234
80	89.6	85.4	0.313	0.328
100	88.9	84.7	0.531	0.5

From Table 2, it can be seen that for each granular level of parameter k, the KGSVM algorithm is obviously superior to the GSVM learning algorithm. When k=80, the accuracy rate achieves the best for both algorithms, and the result of KGSVM algorithm is higher than traditional algorithm about 4%. Although the spend time of dividing granules is slightly longer than that by the traditional dividing granules in low-dimensional space, it is faster comparing with original SVM directly training.

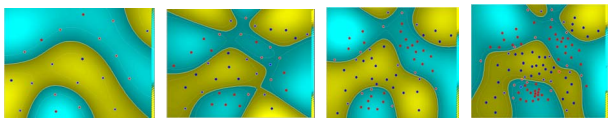
The hyperplanes with the best granular parameter k=80 by KGSVM and GSVM learning algorithms are shown in Fig.5.



(a) KGSVM (b) GSVM

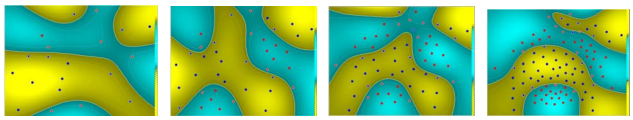
Figure 5. Hyperplanes by KGSVM and GSVM (k=80)

The hyperplanes with other parameters by KGSVM and GSVM learning algorithms are shown in Fig.6 and Fig.7, respectively.



(a) k=20 (b) k=40 (c) k=60 (d) k=100

Figure 6. Hyperplanes by KGSVM with different granular levels



(a) k=20 (b) k=40 (c) k=60 (d) k=100

Figure 7. Hyperplanes by GSVM with different granular levels

Table 3 lists the testing results of two algorithms on other datasets. The underlined values denote the best prediction results, and each corresponding granular parameter is optimal. From Table 3, it can be seen that the testing results are not always improved with increment of the number of granules both for two algorithms. In the process of replacement with granules, a number of data that have little effect on final results are removed factitiously, but in fact these data may take some roles in SVM training. However, it can be observed that KGSVM algorithm is superior to the traditional GSVM. These experiments supported that dividing granules and replacing with them in low-dimensional space may not achieve the same good results as they do in high-dimensional space.

Table 3. Comparisons of testing results between KGSVM and GSVM

数据集	k 值	20	40	60	80	100
Breast_cancer	KGSVM	74.9	74.6	72.1	75.7	77.6
	GSVM	65.7	71.6	72.7	72.2	73.9
Diabetis	KGSVM	70.6	71.4	70.3	69.9	73.2
	GSVM	64.5	67.5	69.3	71.3	70.9
Heart	KGSVM	75.8	75.3	86.6	83.3	81.4
	GSVM	75.2	77.4	86.2	79.5	83.2
Thyroid	KGSVM	94.1	96.6	97.6	98.2	97.8
	GSVM	92.3	97.2	96.7	97.9	97.6
Titanic	KGSVM	77.3	77.8	77.8	77.8	77.8
	GSVM	77.5	77.5	77.5	77.5	77.5

4. Conclusions

This paper proposes a novel SVM learning model KGSVM, which can compress data in high dimension feature space through dividing granules and replacing with these granules. In so doing, the problem of inconsistent data distribution due to dividing granules and replacing with them can be avoided, and then the satisfactory generalization performance can be obtained.

To improve the performance of KGSVM, combining the KGSVM algorithm with the model selection will be our future work. Besides, more other approaches for dividing granules like kernel neural network, kernel decision tree, kernel rough set, et al, will be taken into account.

Acknowledgements

The work described in this paper was partially supported by the National Science Foundation of China (No.60673095), Hi-tech R&D 863 Program (No.2007AA01Z165), Key Project of Science Technology Researches of Ministry of Education, China (No.208021), Program for New Century Excellent Talents in University (NCET-07-0525), Program for the Top Young Academic Leaders of Higher Learning Institutions (TYAL), Project for Returned Overseas (No.2008-14) of Shanxi Province and Key Project of Natural Science Foundation of Shanxi Province (2009011017-2).

References

[1] Vapnik V. Statictical Learning Theory. New York:Wiley, 1998:11-23.

[2] Suykens J, Vandewa L J. Least squares support vector machine classifiers. Neural Processing Letters, 1999, 9(3): 293-300.

[3] Osuna E, Freund R, Girosi F. Training support vector machines:An application to face detection//IEEE computer society conference on computer vision and pattern recognition, 1997, 130-136.

[4] Mukerjee S, Osuna E, Girosi F. Nonlinear prediction of chaotic time series using a support vector machine//Principle J, Giles L, Morgan N. Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing, IEEE press, 1997, 1125-1132.

[5] Keerthi S, Shevade S, Bhattcharyya C, et al. A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Trans on Neural Networks, 2000, 11(1): 124~136.

[6] Frieb T, Chistianini N and Campbell C. The kernel adatron algorithm: A fast and simple learning procedure for support vector machines. Shavlik J. ed. Proceedings of the The 15th International Conference of Machine Learning. CA: Morgan Kaufmann Publishers, 1998, 188-196.

[7] Joachims T. Make large-scale support vector machine learning practical. Advances in kernel methods: support vector machines. Cambridge, MA: MIT Press, 1998. 11: 169-184.

[8] Luenberger D G. Linear and nonlinear programming. NY: Addison-Wesley, 1986.

[9] Mangasarian O, Musicant R. Successive overrelaxation for support vector machines. IEEE Trans on Neural networks, 1999: 1032-1037.

[10] Vapnik V. Estimation of Dependence Based on Empirical Data. NY: Springer Verlag, 1982.

[11] Wang W J, Xu Z B, A heuristic training in support vector regression. Neurocompuing. 2004, 61: 259-275.

[12] Zhang X G. Using class-center vectors to build support vector machines. In Proceedings of NNSP'99,1999: 3-11

[13] Tang Y C. Granular support vector machines based on granular computing, soft computing and statistical learning. Georgia Stage University: College of Arts and Sciences , 2006