

文章编号:1000-8055(2012)07-1604-09

Aero-engine fault diagnosis applying new fast support vector algorithm

XU Qi-hua¹, GENG Shuai¹, SHI Jun²

(1. College of Electronic Engineering,

Huaihai Institute of Technology, Lianyungang Jiangsu 222005, China;

2. No. 365 Institute, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: A new fast learning algorithm was presented to solve the large-scale support vector machine (SVM) training problem of aero-engine fault diagnosis. The relative boundary vectors (RBVs) instead of all the original training samples were used for the training of the binary SVM fault classifiers. This pruning strategy decreased the number of final training sample significantly and can keep classification accuracy almost invariable. Accordingly, the training time was shortened to 1/20 compared with basic SVM classifier. Meanwhile, owing to the reduction of support vector number, the classification time was also reduced. When sample aliasing existed, the aliasing sample points which were not of the same class were eliminated before the relative boundary vectors were computed. Besides, the samples near the relative boundary vectors were selected for SVM training in order to prevent the loss of some key sample points resulted from aliasing. This can improve classification accuracy effectively. A simulation example to classify 5 classes of combination fault of aero-engine gas path components was finished and the total fault classification accuracy reached 96.1%. Simulation results show that this fast learning algorithm is effective, reliable and easy to be implemented for engineering application.

Key words: aero-engine; support vector machines; fault diagnosis; large-scale training set; relative boundary vector; sample pruning

CLC number: V263.6; TP181

Document code: A

Support vector machine (SVM) is a new kind of machine learning method first introduced by Vapnik. Based on statistical learning theory, SVM has excellent learning performance. SVM can obtain the only global optimal solution and has good generalization performance. Furthermore, its algorithmic complexity has nothing to do with sample dimension by using kernel function. Owing to the above-mentioned advantages, SVM is actively used for aero-engine fault diagnosis and is proved to be an effective method^[1-8].

When the training pattern set is very large, SVM may have great difficulty in its realization,

even can not work properly because of the sharp increase of compute time and memory to solve the quadratic programming (QP) problem^[9-12].

To overcome these difficulties, researchers usually consider from two aspects for large training set. First, design a new realization algorithm which is different from the traditional quadratic optimization method to reduce the number of the samples in optimization computing. On the other hand, some useful reducing algorithms were proposed and tried by many researchers to reduce the number of training samples^[11-16]. The basic idea of the reducing algorithms is to re-

Received: 2011-11-28

URL: http://www.cnki.net/kcms/detail/11.2297.V.20120801.1438.201207.1604_Q23.html

Foundation item: "Six professional talent summit projects" of Jiangsu Province(07-E-029);

Natural Science Foundation of Colleges and Universities in Jiangsu Province(JHSD08-40);

"Qing-Lan Project" Foundation of Jiangsu Province(2007).

move the samples with little help for learning and to reserve those ones which have greater impact on the classification hyperplane. Based on this idea, the training set scale can be decreased and the training time can accordingly be reduced, while the classification accuracy of the SVM classifier can be ensured.

It is well known that the optimal hyperplane which determines the classification results are decided by the so-called support vectors. This means that the other samples except support vectors are redundant, so they do not affect the location of the optimal hyperplane and can be pruned away. In the process of SVM training, not all of the training samples can become the support vectors. It is obvious that only those samples near the border are likely to become support vectors. If we define the above near border samples to be relative boundary vectors (RBV) ^[12-13], and use RBVs instead of all the original samples for SVM training, we may get better training effect. The training speed will certainly have a substantial increase due to the smaller proportion of the RBVs in the training set. What is more, the classification accuracy can be guaranteed because this strategy removes only the samples which are little helpful for SVM learning from the training set and retain the RBVs which have greater effect on the optimal hyperplane.

When sample aliasing exists between the two types of training data, we will obtain a wrong optimal hyperplane if we only use the RBVs calculated directly according to Reference [12] for SVM training. Under this condition, the so-called "RBVs" are not really near the border of the two categories because of aliasing. If only the wrong "RBVs" are chosen as the final training samples and the other samples are pruned away, including some critical data points which are really in the vicinity of the boundaries of the two categories and may become support vectors, it is easy to know that the classification error is likely to emerge. The experiment results in this paper proved the above ideas.

This paper presents a new fast SVM learning algorithm for aero-engine fault diagnosis at the condition of large-scale training set. The main idea of this algorithm is that those aliasing training samples which are not the same class are eliminated first, and then the RBVs are calculated. Moreover, a certain margin is given for the relative boundary vector computing in order to prevent the loss of critical sample points resulted from aliasing. In other words, those samples whose distances to the relative boundary vectors are smaller than a certain amount will be selected for SVM training. Simulation results given later in this paper proved the above algorithm effective and necessary.

1 Aero-engine fault diagnosis based on SVM

A fault diagnosis problem for the gas path components of a turbojet engine is discussed in this paper. we use SVMs to infer the types of combination fault for the gas path components according to the engine fault symptomatic data.

Engine fault diagnosis is essentially a pattern classification problem^[2]. The SVM algorithm for the classification of two types of engine fault data is described as below.

For training sample $(x_i, y_i) \in R^n \times \{+1, -1\}$, $i = 1, 2, \dots, m$, the training process of SVM is to solve the following optimization problem^[10]

$$\left. \begin{aligned} \max W(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{s. t.} \quad &\sum_{i=1}^m \alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C, i = 1, 2, \dots, m \end{aligned} \right\} \quad (1)$$

We can obtain the classifier which consists of optimal hyperplane, support vector and corresponding parameters after the training is completed. The test sample x is predicted by the following formula

$$f(x) = \text{sgn} \left[\sum_{i=1}^m y_i \alpha_i K(x, x_i) + b \right] \quad (2)$$

The meaning of the parameters used in e-

quation (1), (2) are as follows. α_i is a non-negative Lagrange multiplier, C is the regularization parameter which impacts on the accuracy of classification. K is the Kernel function satisfying the Mercer condition and here we use the Gauss kernel function

$$K(x, x_i) = \exp\left(-\frac{|x - x_i|^2}{2\sigma^2}\right) \quad i = 1, \dots, m$$

b is the bias, the average of b can be got according to all the support vectors^[3]:

$$b = \frac{1}{I_s} \sum_{i \in S_v} \left[y_i - \sum_{j=1}^m y_j \alpha_j K(x_i, x_j) \right]$$

Where S_v is the set of support vectors, which number is I_s .

SVM is realized with sequential minimal optimization (SMO) algorithms. Reference [10] gives a detailed principle description and program pseudo-code, so we do not repeat them here.

Here, each SVM involves 8 input parameters: high/low pressure rotator speed n_h/n_l , exportation pressure in high/low pressure compressor p_2/p_x , exportation pressure in high/low pressure turbine p_y/p_4 , exportation temperature in high pressure compressor t_2 and exportation temperature in low pressure turbine t_4 . All the input parameters are indicated with the relative values deviating from standard state and have no unit.

A binary SVM can finish the classification of two fault types directly. The combination fault of the engine gas path components consists of 5 classes including no fault (NON), low pressure compressor plus high pressure compressor fault (LP+HP), high pressure compressor plus high pressure turbine fault (HP+HT), high pressure turbine plus low pressure turbine fault (HT+LT) and low pressure turbine plus low pressure compressor fault (LT+LP)^[17]. To diagnose the above 5 categories of faults, multiple classifiers must be trained. In this paper, we select the widely used “one-against-one” method to construct multiple classifiers^[3].

2 Some related definitions

1) Distance between two training samples

In the original input space, the Euclid distance of the two training sample inputs $x_i = (x_i^1, x_i^2, \dots, x_i^n)$, $x_j = (x_j^1, x_j^2, \dots, x_j^n)$, can be expressed as

$$d(x_i, x_j) = \|x_i - x_j\| = \sqrt{\sum_k (x_i^k - x_j^k)^2} \quad (3)$$

Later in this paper, the distance formula (1) will be used to calculate the relative boundary vectors and the nearest neighbor points of training samples.

2) Relative boundary vector

Suppose the first class of samples are x_{i1} , $i = 1, \dots, m_1$, and the second class of samples are x_{j2} , $j = 1, \dots, m_2$, where m_1 is the number of class 1, and m_2 is the number of class 2.

Make

$$d_i = \min(d(x_{i1}, x_{j2})), \quad j = 1, \dots, m_2 \quad (4)$$

then

$$x_{l2} \quad (1 \leq l \leq m_2),$$

fulfilling

$$d(x_{i1}, x_{l2}) = d_i \quad (5)$$

is defined to be a relative boundary vector among class 2 samples corresponding to x_{i1} , where $d(x_{i1}, x_{j2})$ represents the distance from x_{i1} to x_{j2} .

After i taken over all of the m_1 values, all the relative boundary vectors of class 2, locating at the relative boundary position to class 1, can be obtained. The set which consists of all these relative boundary vectors is denoted as S_2 .

Similarly, the set called S_1 , consisting of all the relative boundary vectors of class 1, can be got.

3) Nearest neighbor of training samples

For sample x_i , if there is a sample x_p , meeting

$$d(x_i, x_p) = \min(d(x_i, x_j)), \quad j = 1, \dots, m_0 \quad (6)$$

then x_p is defined as the nearest neighbor of sample x_i , where the distance between each sample point and itself is defined as ∞ , and m_0 is the total number of the original training samples without any pruning.

3 Pruning strategy and fast algorithm for large-scale training sample set

Shown as Fig. 1, the symbol “ $\cdot$ ” represents class 1 and “ $+$ ” represents class 2, while “ $\bigcirc$ ” is for RBVs. According to the definition of RBV, the two heterogeneous samples with $\triangle$ symbol, which are mixed into the other class, will also be counted as RBV. Obviously, they are two wrong relative boundary vectors resulting from aliasing. It is not appropriate to put them into the final training set, on the contrary, they should be removed primarily.

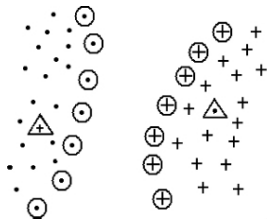


Fig. 1 Wrong relative boundary vectors because of aliasing samples

In addition, when there is sample aliasing, data points may be very dense in some area in original input space. If we only take the RBVs calculated according to the definition directly, some critical data points which have important impact on the optimal hyperplane may be missing ones and can not be the final training samples. Therefore, it is necessary to give RBV a proper margin δ .

The meaning of parameter δ is as follows: a data point will be selected as a training sample if it is located near some RBV, and its distance to the other class sample is smaller than $(1 + \delta)$ times of the RBV distance to the same sample. If $\delta = 0$, the final training set consists of only the relative boundary vectors.

According to the above discussion, a pruning strategy for large-scale training set and the corresponding fast learning algorithm is given as follows.

For a given training set

$(x_1, y_1), (x_2, y_2), \dots, (x_{m_0}, y_{m_0}), x_i \in R^n; y_i \in \{-1, 1\}, i = 1, 2, \dots, m_0$, we use the follow-

ing strategy to reduce it.

Step 1 Compute the distance between every two points

Compute the distance between every two sample points according to formula (3).

Step 2 Eliminate the aliasing samples which are not the same class

1) For each sample point, according to formula (6), find out its nearest neighbor.

2) If the point and its nearest neighbor belong to the same class, i. e. $y_i = y_p$, keep this point. Otherwise, if the point and its nearest neighbor belong to different categories, i. e. $y_i \neq y_p$, then remove the point.

3) Get the remaining training set: the number of class 1 is m_1 and the corresponding outputs are all 1, the number of class 2 is m_2 and the corresponding outputs are all -1.

Step 3 Compute the RBV sets

1) If the proportion of aliasing samples is small (here taking 3%), set $\delta = 0$, otherwise, set constant $\delta \in (0, 1)$.

2) According to formula (4) and (5), calculate the relative boundary vector corresponding to x_{i1} , and assign it to $S_2(1), L=1$.

3)

for $i=1:m_1$

{

Calculate the distance d_i between x_{i1} and the corresponding RBV according to formula (3), search out all the sample points whose distances to x_{i1} are smaller than $(1 + \delta) \times d_i$ in class 2, then note these sample points as N_b , of which the number is L_i .

for $j=1:L_i$

{

if $N_b(j) \neq S_2(1:L)$

$L=L+1;$

$S_2(L) = N_b(j);$

}

}

4) According to the steps similar to 1)–3), find out the set S_1 of the RBVs of class 1.

Step 4 Get the final training sample set

The union of the two sets S_1 and S_2 is used

as the reduced sample collection for SVM training. After some pruning, the number of the final training samples is m , which is denoted by

$$(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m), x_i \in S_1 \cup S_2, \\ y_i \in \{1, -1\}, i = 1, 2, \dots, m.$$

With the final training samples, the SVM is trained and we call the final obtained classifier L-SVM (Lagrangian support vector machine).

4 Simulation experiments

4.1 data generation and pre-process

We have 56 engine fault samples of 5 classes in Table 7-3 and Table 7-5 from Reference [17], including 32 samples of NON, every 6 samples of LP+HP, HP+HT, HT+LT and LT+LP individually. Each sample input consists of 8 fault symptomatic parameters and is 8-dimensional. The 56 samples are used to calculate the mean and variance of the 5 classes respectively. Five classes of normal distributed new fault samples with their own mean and variance are generated: 4 000 samples for each class. When training a binary SVM, 2 000 samples (1 000 each class) are selected for training and the other 1000 samples (500 each class) are arranged as test ones.

In order to reduce the computation workload, the training sample data are pre-processed with principal component analysis (PCA): the eigenvalues (%) corresponded to the characteristic components are [44.4770, 24.3257, 12.9112, 9.6065, 4.9836, 2.1524, 0.9213, 0.6223]. After PCA, we take each sample input a 4-dimensional vector because the contribution rate of the first 4 characteristic components is more than 91.32%, being able to meet the requirements.

4.2 Simulation results according to the algorithm in this paper

4.2.1 L-SVM classifier

Fault classification was finished in MATLAB environment. The computer used for simulation is Pentium(R) 4, CPU 3.06GHz, RAM 2 GB.

Take the two types of fault for example;

NON and HP + HT. Set kernel parameter $\sigma^2 = 0.8$, SVM regularization parameter $C = 100$ and RBV margin $\delta = 0.1$. The simulation results applying basic SVM without sample pruning as well as our L-SVM are both listed in Table 1.

Table 1 Simulation results according to basic SVM and our L-SVM (for fault NON and HP+HT)

Classifier	SVM	L-SVM
Classification accuracy/%	92.7	92.5
Number of training samples	2000	281
Training time/s	4060	202
Number of support vectors	275	63

The training time of L-SVM in Table 1 includes the sample pruning time. Compared with basic SVM classifier, the training time of the L-SVM classifier with our pruning algorithm is about 1/20 of the former, but the classification accuracy is almost invariable. The improvement of the training time is obvious because the number of the final training samples is reduced from 2000 to 281. Meanwhile, the number of support vectors is reduced to 63, only 22.9% of basic SVM, and this may shorten the classification time.

The “numChanged” is an important variable in Platt’s SMO algorithm, which refers to the pairs of optimization variable obtained in each round of iteration. The iteration can be ended only when numChanged is equal to zero in both inner loop and outer loop^[10]. Therefore, we can estimate the convergent speed from the change of numChanged in each round. Fig. 2 shows the change curves of numChanged computed according to basic SVM and L-SVM classifiers, respectively.

In Fig. 2, the red dotted line corresponds to L-SVM and the blue solid line stands for the basic SVM without sample pruning. In iteration round No. 200 – 3 200, the number of numChanged represented by the blue solid line changes repeatedly within a small range because of the large scale of training sample. So the convergent speed is very slow and the total iteration

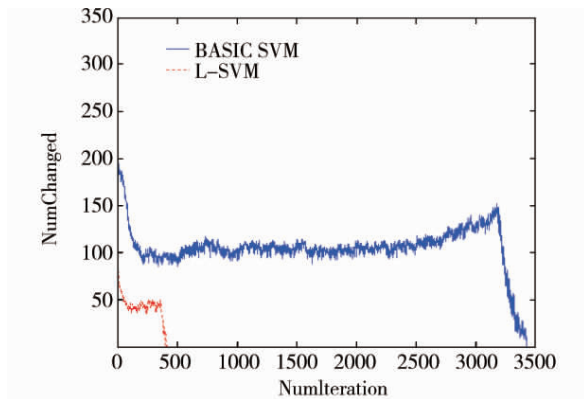


Fig. 2 Curves of numChanged according to different classifiers

round is much more, which extends the computing time greatly.

4.2.2 Comparison with related algorithms

The results of training-sample reducing through different strategies are compared and the remaining final training samples after pruning are shown in Fig. 3.

Fig. 3 shows the first two-dimensional components after PCA according to the 2 classes of fault sample data (NON and HP+HT). As the contribution rate of the first two-dimensional components reaches up to 68.8%, the two-dimensional data can represent the sample distribution basically. Obviously, aliasing exists among the samples. Fig. 3(a) is obtained in the case that we use the pruning strategy in this paper and compute the RBVs with a RBV margin after the elimination of sample aliasing. The final training samples obtained through the pruning strategy in this paper character the boundary of the two classes properly. When we use these samples for SVM training, we can get a good classification result which is very close to the result achieved by using all the training samples.

Fig. 3(b) is the situation of computing RBVs directly. Due to aliasing, the calculated relative boundary vectors have some distortion and this makes it impossible for the reserved samples to character the boundary of the two classes properly. If we use these reserved samples for SVM training, a greater classification error will occur certainly. In fact, the classification accu-

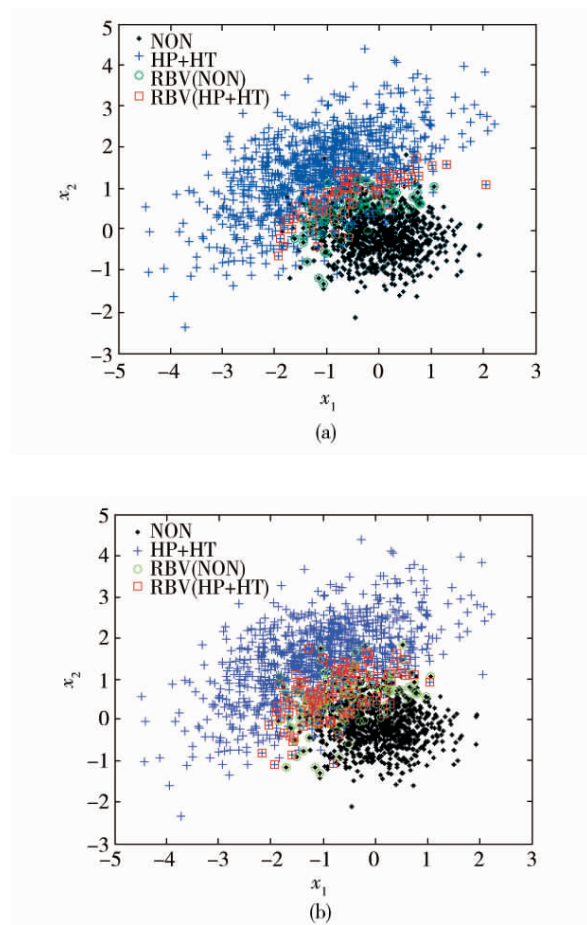


Fig. 3 Training samples and remained RBVs at the condition of aliasing

racy now is only 60.9%, having a greater difference with Fig. 3(a).

The related classification accuracy is listed in Table 2. The results are also given when aliasing is eliminated but δ is set to 0. That is to say, only the relative boundary vectors are considered as the final training samples in this case. If only the RBVs are selected strictly, it is very likely to lose those samples which have very important impact on the decision boundary since-sample aliasing near the border is usually more serious. This is called over-pruning. Similarly, the final training samples got in this case can not character the boundary of the two classes correctly, and the classification accuracy is not high, which is only 87.4%.

Fig. 4 shows the results at the case of calculating RBVs directly without aliasing, and the two classes of fault are LP+HP, HT+LT. The final training samples obtained through the

pruning strategy in this paper character the boundary of the two classes correctly. When using these samples for SVM training, we got a perfect classification result which was the same as the result achieved by using all the training samples. The classification accuracy now reached 100%.

Table 2 Classification results according to different pruning strategies

Sample aliasing elimination	RBV margin δ	Classification accuracy/%
✓	0.1	92.5
	0	60.9
✓	0	87.4

4.2.3 Results of engine multi-faults classification

We want to classify 5 classes of enginefault. According to the multi-classification method “one-against-one”, 10 binary SVM classifiers are needed to be trained and the final classification results are decided by vote^[3]. Similarly to the binary classifier for fault NON~LP+HP, the binary L-SVM classifiers obtained using the other classes of training fault samples also have satisfactory results shown in Table 3, where m_1

and m_2 are the sample numbers of the two related classes after aliasing elimination.

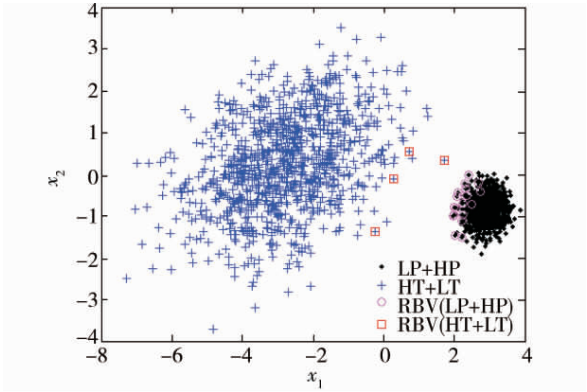


Fig. 4 Training samples and remained RBVs without aliasing

Fig. 5 is the fault sample distribution obtained with the first two-dimensional components after PCA (300 samples for each class). The aliasing among various classes of samples can be clearly observed and the results in Table 3 describe the sample distribution properly. For example, since there is no aliasing between the 2 classes of samples, the binary SVM classifiers LP+HP~HP+HT, LP+HP~HT+LT and LP+HP~LT+LP all get 100% classification accuracy. While the classification accuracy of classifier HP+HT~HT+LT is only 81.3%

Table 3 Simulation results of the binary SVMs

Binary SVMs	Classification accuracy/%	Number of training samples	Training time/s	Number of support vectors	m_1/m_2	Kernel parameter σ^2	RBV margin δ
NON~LP+HP	100.0	31	125	16	999/1000	0.80	0
NON~HP+HT	92.5	281	202	63	929/913	0.80	0.1
NON~HT+LT	94.0	274	195	58	960/944	0.75	0.1
NON~LT+LP	99.4	64	153	22	990/988	1.40	0
LP+HP~HP+HT	100.0	53	151	13	1000/1000	0.90	0
LP+HP~HT+LT	100.0	22	122	12	1000/1000	0.75	0
LP+HP~LT+LP	100.0	12	118	5	1000/1000	5.00	0
HP+HT~HT+LT	81.3	305	220	263	740/723	30.00	0.1
HP+HT~LT+LP	99.0	85	167	21	992/996	10.00	0
HT+LT~LT+LP	96.8	266	190	45	957/971	3.50	0.1

due to the serious aliasing.

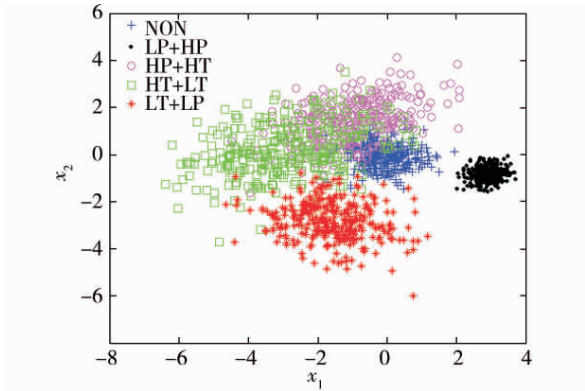


Fig. 5 Five classes of fault samples

Finally, the classification results for 2 500 test samples which belong to 5 classes, i. e. NON, LP+H, HP+HT, HT+LT and LT+LP, are as follows: 2 403 samples are classified correctly and only 97 samples have a wrong classification, the total classification accuracy reaches 96.1%. Because the training of multiple binary SVM classifiers is needed for multi-class fault classification, the total training time may be shortened more obviously if we apply the algorithm presented in this paper.

As the final number of support vector is reduced greatly, the classification time is also shortened accordingly. The classification time of classifier HP+HT~HT+LT is about 3 seconds, and the others all have a classification time less than 1 second.

4.2.4 Results for large-scale training set

To validate effectiveness of the fast learning algorithm in this paper for large-scale training set, we computed the training-sample numbers after pruning when the total original training-sample numbers varied from 2 000 to 6 000 and the results are now listed in Table 4. RBV margin δ is set to 0.1. According to our fast learning algorithm, the numbers of training samples after pruning are only 1/7–1/10 of the total numbers of the original samples. When the original training set is very large, the decrease of training-time will be very considerable by using the sample points after pruning for SVM training. Fur-

thermore, as the number of training samples is reduced, the final number of support vectors will also be reduced accordingly and that is certainly very beneficial to the reduction of the classification computing-time.

Table 4 Training sample numbers after pruning using our algorithm

Total original training-sample numbers	Training-sample numbers after pruning	Percent of training samples after pruning/%
2 000	81	14.05
3 000	305	10.17
4 000	410	10.25
5 000	571	11.42
6 000	678	11.30

5 Conclusions

1) To the binary SVMs for engine fault classification, using RBVs instead of all the original training samples can decrease the number of the final training samples significantly and keep the classification accuracy almost invariable. Accordingly, the training time is shortened obviously and the classification time is also reduced.

2) When sample aliasing exists, the aliasing sample points which are not the same class should be eliminated before the relative boundary vectors are computed. Besides, the samples near the relative boundary vectors should be selected for SVM training in order to prevent the loss of some key sample points resulted from aliasing. This can improve classification accuracy effectively.

3) Since the training of multiple binary SVM classifiers is needed for multi-class fault classification, the algorithm presented in this paper may be more effective to the multi-class engine fault diagnosis problem with a large-scale training set. Simulation examples show that the classification accuracy of 5 classes of engine combination fault reaches 96.1%. Therefore, our algorithm can be applied as a good method for

aero-engine fault diagnosis with a large-scale training set.

References:

- [1] XU Qihua, SHI Jun. Fault diagnosis for aero-engine applying a new multi-class support vector algorithm[J]. Chinese Journal of Aeronautics, 2006, 19(3): 175-18.
- [2] XU Qihua, SHI Jun. Some studies in aero-engine fault diagnosis using support vector machines[J]. Acta Aeronautica et Astronautica Sinica, 2005, 26(6): 687-690. (in Chinese)
- [3] XU Qihua, SHI Jun. Aero-engine fault diagnosis based on support vector machine[J]. Journal of Aerospace Power, 2005, 20(2): 298-302. (in Chinese)
- [4] XU Qihua, SHI Jun. New multi-class support vector algorithm and its application in fault diagnosis[J]. Acta Simulata Systematica Sinica, 2005, 17(11): 2766-2768, 2784. (in Chinese)
- [5] WEI Xunkai, LU Cheng, WANG Cheng. Applications of support vector machines to aeroengine fault diagnosis[J]. Journal of Aerospace Power, 2004, 19(6): 844-848. (in Chinese)
- [6] HAO Ying, SUN Jianguo, YANG Guoqing, et al. The application of support vector machines to gas turbine performance diagnosis[J]. Chinese Journal of Aeronautics, 2005, 18(1): 15-19.
- [7] WANG Xuhui, HUANG Shengguo, SHU Ping. Remote diagnosis of aeroengine's fault using LS-SVM[J]. Mechanical Science and Technology, 2007, 26(5): 595-599. (in Chinese)
- [8] YANG Jun, XIE Shousheng, YU Dongjun. Aero-engine fault diagnosis based on support vector machine[J]. Mechanical Science and Technology, 2005, 24(1): 123-126. (in Chinese)
- [9] Haykin S, YE Shiwei, SHI Zhongzhi. Neural networks: a comprehensive found, second edition[M]. Beijing: China Machine Press, 2004. (in Chinese)
- [10] Cristianini N, Shawe-Taylor J, LI Guozheng et al. An introduction to support vector machines and other kernel-based learning methods[M]. Beijing: Publishing House of Electronic Industry, 2004. (in Chinese)
- [11] Almeida M B, Braga A, Braga J P. SVM-KM: Speeding SVMs learning with a priori cluster selection and k -means [C] // Proceedings of the Sixth Brazilian Symposium on Neural Networks. Washington, DC, USA: IEEE Computer Society, 2000: 162-167.
- [12] XU Hongmin, WANG Ruopeng, ZHANG Huainian. Fast classification algorithm for support vector machine[J]. Journal of Beijing Institute of Petro-Chemical Technology, 2009, 17(4): 55-58. (in Chinese)
- [13] AN Jinlong, WANG Zhengou. Pre-extracting support vectors for support vector machine[J]. Computer Engineering, 2004, 30(10): 10-11, 48.
- [14] HE Qiang, XIE Zongxia, HU Qinghua. Neighborhood based sample and feature selection for SVM classification learning[J]. Neurocomputing, 2011, 74(10): 1585-1594.
- [15] LI Honglian, WANG Chunhua, YUAN Baozong. An improved SVM: NN-SVM[J]. Chinese Journal of Computers, 2003, 26(8): 1015-1020. (in Chinese)
- [16] LI Honglian, WANG Chunhua, YUAN Baozong. A learning strategy of SVM used to large training set[J]. Chinese Journal of Computers, 2004, 27(5): 715-719. (in Chinese)
- [17] YANG Weihua. Aero-engine modelling and fault diagnosis [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2000. (in Chinese)