Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Data fusion for fault diagnosis using multi-class Support Vector Machines*

HU Zhong-hui (胡中辉)†, CAI Yun-ze (蔡云泽), LI Yuan-gui (李远贵), XU Xiao-ming (许晓鸣)

(*Department of Automation, Shanghai Jiao Tong University, Shanghai 200030, China*)

†E-mail: huhzh@sjtu.edu.cn

Received July 26, 2004; revision accepted Mar. 15, 2005

Abstract: Multi-source multi-class classification methods based on multi-class Support Vector Machines and data fusion strategies are proposed in this paper. The centralized and distributed fusion schemes are applied to combine information from several data sources. In the centralized scheme, all information from several data sources is centralized to construct an input space. Then a multi-class Support Vector Machine classifier is trained. In the distributed schemes, the individual data sources are processed separately and modelled by using the multi-class Support Vector Machine. Then new data fusion strategies are proposed to combine the information from the individual multi-class Support Vector Machine models. Our proposed fusion strategies take into account that an Support Vector Machine (SVM) classifier achieves classification by finding the optimal classification hyperplane with maximal margin. The proposed methods are applied for fault diagnosis of a diesel engine. The experimental results showed that almost all the proposed approaches can largely improve the diagnostic accuracy. The robustness of diagnosis is also improved because of the implementation of data fusion strategies. The proposed methods can also be applied in other fields.

Key words: Data fusion, Fault diagnosis, Multi-class classification, Multi-class Support Vector Machines, Diesel engine
doi:10.1631/jzus.2005.A1030 **Document code:** A **CLC number:** TP181

INTRODUCTION

The failure of machinery reduces the production rate and increases the costs of production and maintenance. Therefore, it is important to reduce maintenance costs and prevent unscheduled downtimes for machinery. So knowledge of what, where and how faults occur is very important. Condition-based maintenance (CBM) has the potential to decrease life-cycle maintenance costs, increase operational readiness and improve safety. Fault detection and failure mode diagnosis are also necessary for implementing CBM (Byington and Garga, 2001). Fault diagnosis of machinery is an intensively researched field. Many effective methods have been developed. However, because machinery is becoming more and more complex and the failure mode is becoming more and more complicated, and the enormous improvements in the performance and cost of digital signal processing and communication devices in recent years have made it practical and affordable to implement complex monitoring and fault diagnostic techniques for electrical drive systems in online or offline fashion, fault diagnosis is still an ongoing research problem in some sense (Tay and Shen, 2003; Hajiaghajani *et al.*, 2004; Shen *et al.*, 2000).

Shen *et al.*(2000) pointed out that it is difficult to diagnose more than one category of faults and that some results obtained from specific fault method are not easy to interpret. Therefore, they proposed a rough set theory based method that can diagnose more than one category of faults in a generic manner. However, one disadvantage of this method is that the rough set theory cannot be used to deal with continuous attributes. To apply this method, the discre-

*Project supported by the National Basic Research Program (973) of China (No. 2002cb312200), the Hi-Tech Research and Development Program (863) of China (No. 2002AA412010), and the National Natural Science Foundation of China (No. 60174038)

tization method has to be used. Because a prior knowledge about the attribute is difficult to obtain, it is hard to choose an appropriate discretization method. This disadvantage may adversely affect the robustness and accuracy of fault diagnosis.

Support Vector Machines (SVMs), derived from statistical learning theory and VC-dimension theory, have been widely used in many fields and show good performance (Vapnik, 1998). Good generalization ability is an important characteristic of SVMs. In addition, it is very fit for solving problems with small sample set and high dimension. Therefore, it is a promising theory for application to fault diagnosis. Little research is reported about the application of SVM to fault diagnosis. SVMs were originally developed to solve binary classification problems, and so cannot be easily applied to diagnose more than one category of faults. In real world problems, discrimination between more than two categories is often required. How to extend the SVM for binary classification to solve multi-class problem is a desired research goal. Currently there are two types of approaches for constructing multi-class SVM (MSVM) (Hsu and Lin, 2002): One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation. Hsu and Lin (2002) indicated that the one-against-one and directed acyclic graph (DAG) methods are more suitable for practical use than the other methods. In our methods of fault diagnosis, three typical methods of MSVM, one-against-all, one-against-one and directed acyclic graph SVM (DAGSVM), were created by combining several binary SVM classifiers, and applied and evaluated respectively.

In the data fusion area multi-source classification is an important research issue. Different or the same types of information from several data sources are used for classification in order to achieve higher classification accuracy and robustness (Hall and Llinas, 1997; Benediktsson *et al.*, 1997). Research on neural networks, expert systems and artificial intelligence will certainly result in new developments in the data fusion field which hitherto has benefited little from these advances (Grox, 1997). Benediktsson *et al.* (1997) pointed out that conventional statistical pattern recognition methods are not appropriate for classification of multi-source data since such data cannot, in most cases, be modelled conveniently by a multivariate statistical model. Neural-network models are superior to statistical methods in terms of overall classification accuracy of training data. In many classification problems the SVM classifiers outperform the neural-network classifiers (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000; Burges, 1998). The application of MSVMs to multi-source multi-class classification is still an ongoing research area. In this paper, novel schemes combining data fusion techniques with MSVMs are proposed to solve the problem of multi-source multi-class classification.

In an information processing system, fusion can take place at three levels: signal level, feature level, and decision level. Signal-level fusion is often used to reduce measurement uncertainty of a single sensor. Feature-level fusion can effectively use complementary information from different sources. One of the practical limitations is the tremendous size of the feature space and the resulting heavy computational burden. To alleviate this problem, many researchers do not consider dependence among features from different data sources, so that decisions are made individually based on signals from each data source, and then combined together. This approach is known as decision level fusion (Pan *et al.*, 1998). They proposed an entropy based estimation method to further improve the estimation accuracy, which uses neural networks as non-parametric estimators of a posteriori probabilities. Papastavrou and Athans (1992) analyzed the architectures of some very simple organizations in a binary hypothesis testing environment. Both serial and parallel architectures were proposed. Thomopoulos *et al.*(1989) considered the problem of decision fusion in distributed sensor systems. Distributed sensors pass their decisions on the same hypotheses to a fusion center that combines them into a final decision. We found that the feature level and decision level are more practical in fusion, and that the decision−level fusion is most flexible. Therefore, feature−level and decision−level fusion are mainly considered in this paper.

This paper is organized as follows. In Section 2, the standard SVMs for binary classification are reviewed, since all the MSVMs evaluated in this paper are obtained by solving several SVMs for binary classification and then combining them. We introduce one-against-all, one-against-one and DAGSVM methods based on solving and combining several binary SVM classifiers in Section 3. In Section 4, the fusion strategies for fault diagnosis are discussed. Numerical experiments are given in Section 5. Finally, conclusion is given in Section 6.

SUPPORT VECTOR MACHINES FOR BINARY CLASSIFICATION

A concise introduction of SVMs for binary classification is given in this section. An excellent description of SVMs is provided by Cristianini and Shawe-Taylor (2000). For the training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l} \in R^n \times \{+1,-1\}$, where $\boldsymbol{x}_i$ represents condition attribute and y_i represents class attribute, they must be normalized before being used to train SVM classifier. SVMs optimize the classification boundary by separating the data with maximal margin hyperplane, i.e., the optimal classification hyperplane. In real world situations, the data are usually inseparable, so linearly inseparable and nonlinearly inseparable cases are discussed in the following (Burges, 1998; Cristianini and Shawe-Taylor, 2000).

For the linearly inseparable case, the optimal classification hyperplane can be obtained by solving the optimization problem

$$\begin{aligned} &\min J(\boldsymbol{W},\xi) = \frac{1}{2}\|\boldsymbol{W}\|^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{s.t. } y_i[\boldsymbol{W}\boldsymbol{x}_i + b] \ge 1-\xi_i, \\ &\quad\ \xi_i \ge 0, \quad i=1,2,\cdots,l \end{aligned} \tag{1}$$

where C is the constant of capacity control and ξ_i is the slack factor that permits margin failure of corresponding $\boldsymbol{x}_i$.

According to the Lagrange optimization method and duality principle, the optimization problem Eq.(1) can be rewritten as follows:

$$\begin{aligned} &\max M(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle + \sum_{i=1}^{l}\alpha_i \\ &\text{s.t.} \quad \sum_{i=1}^{l}\alpha_i y_i = 0, \quad \alpha_i \in [0,C],\ i=1,2,\cdots,l \end{aligned} \tag{2}$$

By solving the above problem Eq.(2), we can get the optimal hyperplane with maximal margin

$$f(\boldsymbol{x}) = \sum_{sv}\alpha_i y_i \langle \boldsymbol{x}\cdot\boldsymbol{x}_i\rangle + b = 0 \tag{3}$$

Therefore, the decision function based on SVM for linear classification in the input space is

$$d(\boldsymbol{x}) = \text{sgn}[f(\boldsymbol{x})] = \text{sgn}\left[\sum_{sv} y_i\alpha_i \langle \boldsymbol{x}_i\cdot\boldsymbol{x}\rangle + b\right] \tag{4}$$

For the nonlinearly inseparable case, the original data are projected into a certain high dimensional Euclidean space H by a nonlinear map Φ:$R^n \rightarrow H$, so that the problem of nonlinear classification is transformed into that of linear classification in the space H. By introducing the kernel function $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j)\rangle$, it is not necessary to explicitly know $\Phi(\cdot)$ (Burges, 1998). So that the optimization problem Eq.(1) can be translated directly to the more general kernel version

$$\begin{aligned} &\min J(\boldsymbol{W},\xi) = \frac{1}{2}\|\boldsymbol{W}\|^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{s.t. } y_i[\boldsymbol{W}\cdot\Phi(\boldsymbol{x}_i) + b] \ge 1-\xi_i, \\ &\quad\ \xi_i \ge 0, \quad i=1,2,\cdots,l \end{aligned} \tag{5}$$

The problem Eq.(5) can be rewritten as follows

$$\begin{aligned} &\max M(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sum_{i=1}^{l}\alpha_i \\ &\text{s.t.} \quad \sum_{i=1}^{l}\alpha_i y_i = 0, \quad \alpha_i \in [0,C],\ i=1,2,\cdots,l \end{aligned} \tag{6}$$

By solving the problem Eq.(6), we can get the optimal hyperplane with maximal margin

$$f(\boldsymbol{x}) = \sum_{sv}\alpha_i y_i K(\boldsymbol{x},\boldsymbol{x}_i) + b = 0 \tag{7}$$

and the decision function that separates training vectors into two classes in the input space is

$$d(\boldsymbol{x}) = \text{sgn}[f(\boldsymbol{x})] = \text{sgn}\left[\sum_{sv} y_i\alpha_i K(\boldsymbol{x}_i,\boldsymbol{x}) + b\right] \tag{8}$$

MULTI-CLASS SUPPORT VECTOR MACHINES

Real world problems often require classification between more than two classes. However, the problem of multi-class classification does not usually have easy solution, especially for classifiers like SVMs (Platt *et al.*, 2000). How to extend the SVM for binary classification to solve multi-class problem is a desired research goal. Currently there are two types of approaches for constructing MSVMs (Hsu and Lin, 2002). One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation. It is generally simpler to construct classifier theories and algorithms for two classes than for more than two classes. Therefore, an effective strategy is to combine many two-class classifiers into a multi-class classifier. Hsu and Lin (2002) indicated that the one-against-one and DAG methods are more suitable for practical use than the other methods. Therefore, the approaches of creating MSVM by constructing and combining several binary SVM classifiers, such as one-against-all, one-against-one, and DAGSVM, are mainly discussed in this section.

Given a training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ represents condition attribute and $y_i \in \{1, \ldots, K\}$ is the class attribute of $\boldsymbol{x}_i$, the objective of multi-class classification is to correctly discriminate these classes from each other. Three approaches of constructing MSVMs are presented below.

The earliest implementation for SVM multi-class classification is probably the one-against-all method (Hsu and Lin, 2002). Based on the SVM for binary classification, the MSVM using one-against-all strategy can be constructed by applying the following procedure (Vapnik, 1998):

(1) Construct K binary SVM classifiers where $f_i(\boldsymbol{x})$ (i=1, …, K) separates training vectors of the class i from the other training vectors (sgn[$f_i(\boldsymbol{x})$]=1), if vector $\boldsymbol{x}$ belongs to the class i; sgn[$f_i(\boldsymbol{x})$]=−1 otherwise).

(2) Construct the K-class classifier by choosing the class corresponding to the function with maximal value among $f_i(\boldsymbol{x})$ (i=1, …, K). Therefore, the final decision function is

$$d(\boldsymbol{x}) = \arg\max\{f_1(\boldsymbol{x}), \ldots, f_K(\boldsymbol{x})\} \tag{9}$$

The one-against-one method constructs the classifiers where each one is trained on data from two classes. There are different methods for doing the future testing after all binary classifiers are constructed. Hsu and Lin (2002) used the voting strategy suggested in (Friedman, 1996), which is also called the "Max-Wins" strategy. However, if two classes have identical votes, it may not be a good strategy. Therefore, a modified testing strategy is proposed in this paper.

In our modified testing strategy, the "Max-Wins" strategy is first applied to test the one-against-one MSVM. If more than one class has identical votes, the following strategy is used based on the results obtained by using the "Max-Wins" strategy. Suppose the max votes of p classes are equal to m, each of the p classes has m functions $f_{ij}(\boldsymbol{x})$ (i=1, …, m), $j=k_h \in \{1, \ldots, K\}$ (h=1, …, p). The decision function for multi-class classification is

$$d(\boldsymbol{x}) = \arg\max_j \left\{ \sum_{i=1}^{m} \left| f_{ij}(\boldsymbol{x}) \right| \right\} \tag{10}$$

that is, the class attribute of $\boldsymbol{x}$ is determined by the sum of maximal distances to the optimal classification hyperplane.

The training phase of DAGSVM proposed by Platt *et al.*(2000) is the same as that of the one-against-one method. However, in the testing phase, it uses a rooted binary directed acyclic graph which has internal nodes and leaves. Each node is a binary SVM of ith and jth classes. Given a test sample, starting at the root node, the binary decision function is evaluated. Then it moves to either left or right depending on the output value. Therefore, we go through a path before reaching a leaf node which indicates the predicted class. An advantage of using a DAG is that some analysis of generalization can be established (Hsu and Lin, 2002; Platt *et al.*, 2000). There are still no similar theoretical results for one-against-all and one-against-one methods yet. In addition, its testing time is less than the one-against-one method.

DATA FUSION STRATEGIES

During the past decades, the data fusion problem has been well researched. However, it is still an ongoing research area because of the promotion from the advances in other fields. The synergistic use of overlapping and complementary data sources provides information that is otherwise not available from individual sources. Furthermore, multiple data sources can provide more robust performance due to the inherent redundancy (Liggins *et al.*, 1997). Therefore, data fusion techniques of combining data from several data sources can yield higher accuracy and robustness than that achieved by single data source. The traditional architecture for fusion is centralized. Data from multiple data sources are sent to a single location where the data are fused. Due to the advances in computer science and communication, the distributed architecture becomes feasible. In this architecture the data from individual data sources in lower level nodes are processed and then the results are sent to higher level nodes to be combined. Although it is conceptually more complicated, the distributed fusion architecture has the following advantages: lighter processing load at each fusion node; no need to maintain a large centralized database; lower communication load; higher robustness. The distributed fusion architecture is also a necessity since many fusion systems have to be built with existing fusion systems as components. Therefore, the proposed fusion schemes in this paper are focused on the distributed architecture.

In this paper, new centralized and distributed architectures based on MSVMs are proposed. The proposed schemes make full use of MSVM characteristic. One is that the MSVMs discussed in this paper are created by constructing and combining several binary SVM classifiers. The other is that the training of a binary SVM classifier is to obtain an optimal classification hyperplane with maximal margin. Therefore, the distances from a vector with unknown class label to the optimal classification hyperplane and the binary outputs of the decision functions, such as Eqs.(3), (4), (7), (8), are the key bases of the proposed schemes. The centralized fusion scheme is labelled as the fusion Scheme I. The distributed fusion schemes include the fusion Schemes II, III, IV and V. In each of the proposed schemes, three types of aforementioned MSVMs are applied separately.

The fusion Scheme I is illustrated in Fig.1. In this scheme, the features of all the data sources are extracted and combined to form a single input space. Then the MSVM for multi-class classification is trained and tested to create a decision maker.

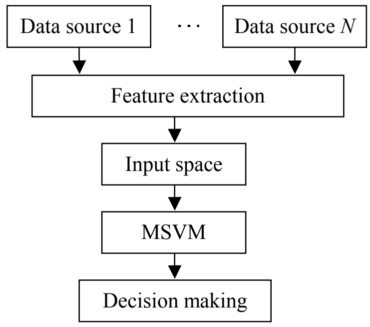

Fig.1 Data fusion Scheme I

In the fusion Scheme II (Fig.2), the features of every data source are extracted and used to form an input space, respectively. Then the sub-MSVM decision makers are created. Suppose that the class attribute set is $CS=\{1, 2, \ldots, K\}$, using the majority vote strategy, the final decision is

$$\begin{aligned} &d(\boldsymbol{x}) = \arg\max\{V_1, V_2, \ldots, V_K\}, \\ &V_j = \sum_{i=1}^{N} \delta_{ij},\ \delta_{ij} = \begin{cases} 1, & d_i(\boldsymbol{x}) = j \\ 0, & d_i(\boldsymbol{x}) \neq j \end{cases} \\ &(i = 1, \ldots, N;\ j = 1, \ldots, K) \end{aligned} \tag{11}$$

where $d(\boldsymbol{x})$ is the final decision function, V_j is comprised of the obtained votes of class j and $d_i(\boldsymbol{x})$ is the output of the ith MSVM trained using the data from the ith data source. It is possible that more than one class have equal maximal votes. That is the limitation of this scheme. In this situation, the final decision can be decided by the following strategies.

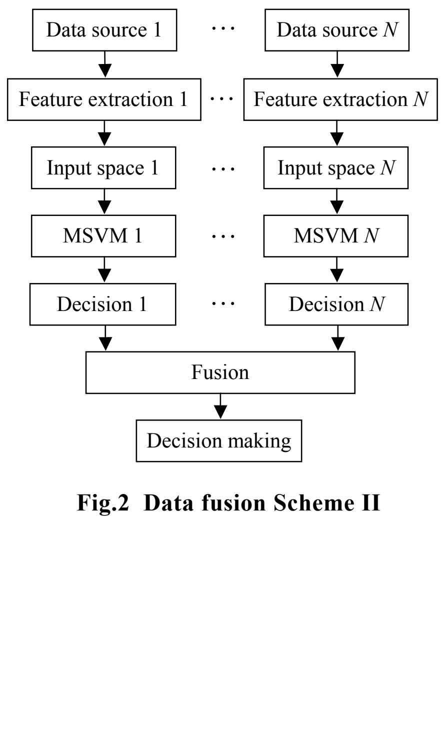

Fig.2 Data fusion Scheme II

Suppose the max votes of p classes are equal to m. Thus, the jth class has m functions $f_{ij}(\boldsymbol{x})$ (i=1, …, m), $j=k_h\in\{1, …, K\}$ (h=1, …, p). For the one-against-all MSVM, the class attribute of the vector $\boldsymbol{x}$ is determined by the sum of maximal distances to the optimal classification hyperplane.

$$d(\boldsymbol{x})=\arg\max_{j}\left\{\sum_{i=1}^{m} f_{ij}(\boldsymbol{x})\right\} \tag{12}$$

For the one-against-one and DAG MSVMs, the auxiliary decision function is

$$d(\boldsymbol{x})=\arg\max_{j}\left\{\sum_{i=1}^{m} \left|f_{ij}(\boldsymbol{x})\right|\right\} \tag{13}$$

In the fusion Scheme III (Fig.3), the difference from the fusion Scheme II is that the output of the ith MSVM is not a decision, but the maximal distance of the vector $\boldsymbol{x}$ to the optimal classification hyperplane. For the one-against-all method, the distance function $f_{ij}(\boldsymbol{x})$ (i=1, …, N; j=1, …, K) can be directly obtained by using the real output of each binary sub-SVM classifier.

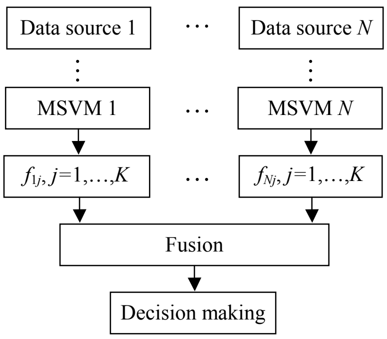

Fig.3 Data fusion Scheme III

For the one-against-one MSVM, the function $f_{ij}(\boldsymbol{x})$ is defined as

$$f_{ij}(\boldsymbol{x})=\sum_{h=1}^{V_{ij}}\left|f_{ijh}(\boldsymbol{x})\right|,\ i=1, …, N; j=1, …, K \tag{14}$$

where V_{ij} is the vote number of the jth class of the ith one-against-one MSVM trained by using the data from the ith data source, and $f_{ijh}(\boldsymbol{x})$ is the real output of the binary sub-SVM classifier, which belongs to the ith one-against-one MSVM and gives its vote to the jth class.

In the DAGSVM method, for a problem with K classes, (K−1) decision nodes will be serially evaluated in order to derive a final decision (Platt *et al.*, 2000). The last decision node is a binary sub-SVM classifier. Suppose that the final decision of the ith DAGSVM is that the vector $\boldsymbol{x}$ belongs to the class r, and that the final maximal distance to the optimal classification hyperplane is md. We set the corresponding maximal distance function is $f_{ir}(\boldsymbol{x})=|md|$. The maximal distance functions of the ith DAGSVM for the other (K−1) classes are defined as $f_{iq}(\boldsymbol{x})=0$ (i=1, …, N), $q\in\{1, …, K\}$ and $q\neq r$.

Among the N MSVMs corresponding to the N data sources, the maximal distance function for each class is defined as

$$f_j(\boldsymbol{x})=\max\left\{f_{1j}(\boldsymbol{x}), f_{2j}(\boldsymbol{x}), \dots, f_{Nj}(\boldsymbol{x})\right\},\quad j=1,\dots,K \tag{15}$$

Thus, the final decision function is

$$d(\boldsymbol{x})=\arg\max\left\{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_K(\boldsymbol{x})\right\} \tag{16}$$

In the fusion Scheme IV (Fig.4), the difference from the fusion Scheme III is that the real output of each binary sub-SVM in the ith MSVM, the respective maximal distance of the vector $\boldsymbol{x}$ to the optimal classification hyperplane, is used as a feature for fusion. Furthermore, these features are weighted in the fusion process (Yan *et al.*, 2003). The final decision function is defined as

$$\begin{aligned} &d(\boldsymbol{x})=\arg\max_{j\in\{1,\cdots,K\}}\left\{\sum_{i=1}^{N} c_{ij}\left|f_{ij}(\boldsymbol{x})\right|\right\},\\ &c_{ij}=\left|f_{ij}(\boldsymbol{x})\right| \Big/ \sum_{\substack{i=1,\cdots,N\\ j=1,\cdots,K}}\left|f_{ij}(\boldsymbol{x})\right| \end{aligned} \tag{17}$$

where c_{ij} (i=1, …, N; j=1, …, K) is the weighted factor.

For the one-against-one MSVM, there exists another fusion strategy illustrated by Fig.5. The decision function is given as Eq.(18).

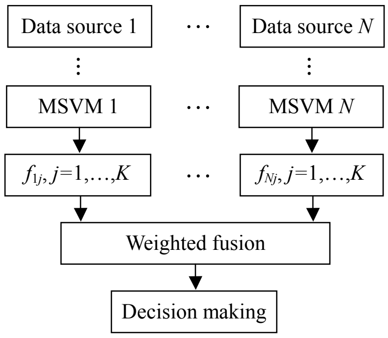

Fig.4 Data fusion Scheme IV

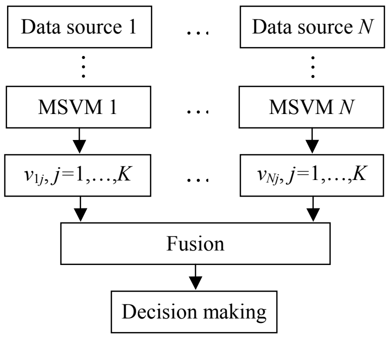

Fig.5 Data fusion Scheme V

$$d(\boldsymbol{x}) = \arg\max\{V_1, V_2, \cdots, V_K\}, \ V_j = \sum_{i=1}^{N} v_{ij} \quad (18)$$

where v_{ij} is the vote number of the jth class of the ith MSVM trained by using the data from the ith data source.

Table 1 summarizes the schemes presented in this section. No additional formula is needed in Scheme I since the centralized strategy is used and only one input space exists. In Schemes II–V the distributed strategy is applied.

EXPERIMENTAL RESULTS

Tay and Shen (2003) proposed a method that uses rough set theory to diagnose the valve fault for a multi-cylinder diesel engine. The original vibration signals are sampled from a 4135 engine surface. The rated engine power is 80 hp and the rated engine speed is 1500 rpm. Due to the complex structure and multi-excitation sources that exist in the diesel engine, a great deal of periodical self-exciting and forced vibration is present. Four states are researched (Tay and Shen, 2003): Normal state; intake valve clearance is too small; intake valve clearance is too large; exhaust valve clearance is too large. Among these four states, three fault types are obtained by deliberately introducing the corresponding fault conditions into the intake valve and exhaust valve on the second cylinder head. Three sampling points are selected for collecting vibration signals. They are located at the first cylinder head, the second cylinder head and the centre of the piston stroke, on the surface of the cylinder block.

Shen *et al.*(2000) pointed out that due to the complexity of vibration signals, the original time series waveform cannot indicate conspicuous difference among the different fault types, and the corresp-

Table 1 Comparison of the proposed schemes

S	Characteristic	Fusion formula	Auxiliary formula		
			OAA	OAO	DS
I	Centralized feature fusion. Features of all data sources are extracted and combined into an input space. Train a MSVM to create a decision maker.	None	None		
II	Majority vote rule. Extract features of each data source and create input spaces, respectively. Train corresponding sub-MSVMs. Decision output of each sub-MSVM is used.	Eq.(11)	Eq.(12)	Eq.(13)	Eq.(13)
III	Maximal distance rule. Real output of each sub-MSVM, respective maximal distance to the optimal hyperplane, is used.	Eq.(16)	Eq.(15)	Eq.(14) Eq.(15)	Eq.(15)
IV	Maximal weighted distance rule. Real output of each binary SVM in all sub-MSVMs, the respective maximal distance to the optimal hyperplane, is used with the weighted factor.	Eq.(17)	No auxiliary formula is needed		
V	Majority vote rule. Decision output of each binary SVM in all sub-MSVMs is used as votes.	Eq.(18)	Only applicable to one-against-one		

S: Scheme; OAA: one-against-all; OAO: one-against-one; DS: DAGSVM onding conventional FFT spectrum also cannot discriminate between these fault types. For thoroughly utilizing the useful information, six features are extracted from the vibration signals at each sampling point. These features represent the information from both the frequency domain and time domain. The transformation formulas for generating the features are introduced as follows (Shen *et al.*, 2000).

(1) *FE*–waveform complexity in frequency domain

$$FE = -\sum_{i=1}^{N/2} X(i)\ln X(i) \tag{19}$$

where $X(i)$ is the FFT spectrum. *FE* can be seen as frequency domain entropy.

(2) *CF*–the center frequency of spectrum

$$CF = \sum_{i=1}^{N/2} \frac{i}{N/2}\mu(X(i)) \tag{20}$$

where $\mu(X(i))$ is defined as

$$\mu(X(i)) = X(i) \Big/ \sum_{j=1}^{N/2} X(j),\ i=1, \ldots, N \tag{21}$$

(3) *TE*–waveform complexity in time domain

$$TE = -\sum_{i=1}^{m} \lambda_i \ln \lambda_i \tag{22}$$

(4) *NC*–nonperiodic complexity

$$NC = \frac{m}{m-1}\sum_{i=2}^{m}\lambda_i^2 \Big/ \sum_{i=1}^{m}\lambda_i^2 \tag{23}$$

In Eqs.(22) and (23), λ_i is the singular value of a time series in accordance with its period and m is the number of periods in a time series. *TE* serves as the time domain entropy.

(5) V_x–the variance of a time series

$$V_x = \frac{1}{n}\sum_{i=1}^{n}[x(t_i) - \bar{x}]^2 \tag{24}$$

(6) σ_4-kurtosis of a time series

$$\sigma_4 = \frac{1}{n}\sum_{i=1}^{n}[x(t_i)]^4 \tag{25}$$

In Eqs.(24) and (25), n is the length of a time series, $\bar{x}$ is the mean value of the whole series and $x(t_i)$ is the time series.

Thus, each instance in the dataset is composed of 18 condition attributes (six features from each sampling point) and a class attribute (four states). The whole dataset was listed by Shen *et al.*(2000). It consists of 37 instances. In the distributed schemes, the six features from one sampling point, added the class attribute, form an individual dataset. Therefore, three datasets from corresponding three data sources are constructed.

The cross-validation test is divided into two equal parts for showing the effect of the rough set theory in fault diagnosis by Tay and Shen (2003). Each part contains the half instances in every class, and these two parts are trained and tested alternatively. The excerpted classification accuracy is listed in Table 2 showing that the average classification accuracy is 76.32%.

Table 2 Classification accuracy of each part

Dataset	Classification accuracy (%)
1st part−TRD; 2nd part−TD	78.95
2nd part−TRD; 1st part−TD	73.68

TRD: training data; TD: testing data

In our experiment, 25 instances of the whole dataset are used as training set and the rest are used as test set. The choice of kernel and of the regularizing parameter was determined via performance on a validation set. Eighty percent of the training set is used for training binary SVM classifiers and the other 20% of the training set is used as validation set. This avoids the problem of "training on the test data", which arises when a classifier undergoes a long series of refinements guided by the results of repeated testing on the same test data (Duda *et al.*, 2001). The test set is used to test the classification accuracy in different strategies. The whole experiment is repeated 50 times, where the training set and test set, with the aforementioned fixed set sizes, are randomly selected without replacement every time. The average experimental results are given in Table 3. Given a complete comparison, the data from every data source (sampling point) are used to train and test MSVM respectively. The classification accuracies of three types of MSVM are given in D1, D2 and D3 columns. Scheme I is the centralized fusion strategy. All the Schemes II−V are distributed strategies of data fusion. Scheme II is mainly based on majority-vote strategy. Scheme III is mainly based on max-distance strategy. Scheme IV is mainly based on max-weighted-distance strategy. Scheme V is also based on majority-vote strategy in some sense, and is different from Scheme II in that Scheme V takes into account the vote of each binary sub-SVM classifier in all MSVMs. Table 3 shows that all the average classification accuracies of combined classifiers in fusion strategies outperform those of the classifiers not using fusion strategies, except for Scheme III for DAGSVM. The Scheme IV for one-against-all MSVM has the best classification accuracy. The classification accuracy obtained by our methods remarkably outperforms that obtained by using rough set theory (Table 2).

CONCLUSION

This paper proposed several new centralized and distributed data fusion strategies based on MSVMs. They are evaluated by applying them to fault diagnosis for a diesel engine. Three methods of constructing MSVMs by combining several binary classifiers, one-against-one, one-against-all and DAG, are mainly discussed. The proposed schemes make full use of the characteristics of MSVMs. One is that the MSVMs discussed in this paper are created by constructing and combining several binary SVM classifiers. Another is that the training of a binary SVM classifier is done by finding the optimal classification hyperplane with maximal margin. The distance from a vector with unknown class label to the optimal classification hyperplane and the binary outputs of the decision functions are the key bases of the proposed fusion schemes. When the data fusion strategy is not used, the one-against-one and DAG methods have higher accuracy (Columns D1, D2, and D3 in Table 3). Hsu and Liu (2002) pointed out that the one-against-one and DAG methods may be more suitable for practical use. If the centralized data fusion strategy is used, the accuracy of the three types of MSVM methods differs little (Column S.I in Table 3). However, in the distributed data fusion situation, different fusion strategies result in different performance for the three types of MSVM methods. For the fusion Schemes II and III, the one-against-one method has the highest accuracy. Therefore, the fusion Schemes II and III are most suitable for the one-against-one method. The fusion Scheme IV is most suitable for the one-against-all method and the improvement of accuracy is outstanding. The one-against-one method also yields high accuracy in fusion Scheme IV. The fusion Scheme V for the one-against-one method yields highest accuracy. In all three distributed fusion strategies, the DAGSVM always has the lowest accuracy. The fusion Scheme III for all three MSVM methods always has the lowest accuracy among all the fusion strategies. To sum up, the one-against-all and one-against-one methods with the distributed fusion strategies are more suitable for practical use, and for all MSVM methods, the fusion Scheme III is not satisfactory. In conclusion, our proposed methods are promising approaches that can improve the robustness and accuracy of fault diagnosis, and also have other applications. In practice, with different emphasis on the performance, different fusion strategy is applied. For example, while emphasizing the robustness, the distributed fusion strategy is more appropriate. Future work is to evaluate the effectiveness of proposed methods if the distributed data sources are heterogeneous.

Table 3 Comparison of experimental results

MSVM	Single data source			S.I	S.II	S.III	S.IV	S.V
	D1	D2	D3					
One-against-all (%)	91.25	75.42	91.67	94.59	95.00	92.50	96.25	–
One-against-one (%)	92.92	80.00	92.50	94.59	96.25	92.92	95.00	96.25
DAGSVM (%)	92.92	80.00	92.50	94.59	95.00	87.92	94.17	–

D1, D2, D3: Data source 1, Data source 2, Data source 3; S.I, S.II, S.III, S.IV, S.V: Scheme I, Scheme II, Scheme III, Scheme IV, Scheme V

References

Benediktsson, J.A., Sveinsson, J.R., Ersoy, O.K., Swain, P.H., 1997. Parallel consensual neural networks. *IEEE Transactions on Neural Networks*, **8**(1):54-64.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**(2):121-167.

Byington, C.S., Garga, A.K., 2001. Data Fusion for Developing Predictive Diagnostics for Electromechanical Systems. *In*: Hall, D.L., Llinas, J.(Eds.), Handbook of Multisensor Data Fusion. CRC Press, Boca Raton, FL, p.1-32.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge University Press, Cambridge.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification (2nd Edition). John Wiley & Sons, Inc., New York, p.483.

Friedman, J., 1996. Another Approach to Polychotomous Classification. Technical Report, Department of Statistics, Stanford University, Stanford, CA.

Grox, X.E., 1997. NDT Data Fusion. John Wiley & Sons, Inc., New York, p.1-42.

Hajiaghajani, M., Toliyat, H.A., Panahi, I.M.S., 2004. Advanced fault diagnosis of a DC motor. *IEEE Transactions on Energy Conversion*, **19**(1):60-65.

Hall, D.L., Llinas, J., 1997. An introduction to multisensor data fusion. *Proceedings of the IEEE*, **85**(1):6-23.

Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, **13**(2):415-425.

Liggins, M.E. II, Chong, C.Y., Kadar, I., Alford, M.G., Vannicola, V., Thomopoulos, S., 1997. Distributed Fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, **85**(1):95-107.

Pan, H., Liang, Z.P., Anastasio, T.J., Huang, T.S., 1998. A Hybrid NN-bayesian Architecture for Information Fusion. Image Processing, ICIP 98, **1**:368-371.

Papastavrou, J.D., Athans, M., 1992. On optimal distributed decision architectures in a hypothesis testing environment. *IEEE Transactions on Automatic Control*, **37**(8): 1154-1169.

Platt, J.C., Cristianini, N., Shawe-Taylor, J., 2000. Large Margin DAG's for Multiclass Classification. *In*: Solla, S.A., Leen, T.K., Müller, K.R.(Eds.), Advances in Neural Information Processing Systems. MIT Press, Cambridge, MA, **12**:547-553.

Shen, L., Tay, F.E.H., Qu, L., Shen, Y., 2000. Fault diagnosis using rough sets theory. *Computers in Industry*, **43**:61-72.

Tay, F.E.H., Shen, L., 2003. Fault diagnosis based on rough set theory. *Engineering Application of Artificial Intelligence*, **16**:39-43.

Thomopoulos, S.C.A., Viswanathan, R., Bougoulias, D.K., 1989. Optimal distributed decision fusion. *IEEE Transactions on Aerospace and Electronic Systems*, **25**(5):761-765.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley, New York.

Yan, W., Shao, H., Wang, X., 2003. Parallel Decision Models Based on Support Vector Machines and Their Application to Distributed Fault Diagnosis. Proceedings of the American Control Conference, **2**:1770-1775.