916. Genetic algorithm for Lagrangian support vector machine optimization and its application in diagnostic practice

Liangmin Li[1], **Guangrui Wen**[2], **Jingyan Ren**[3], **Xiaoni Dong**[4]
[1]Key Laboratory of Automotive Transportation Safety Enhancement Technology of the Ministry of Communication, School of Automobile, Chang'an University, Xi'an 710064, China
[2]Research Institutes of Diagnostics & Cybernetics, Xi'an Jiaotong University, Xi'an 710049, China
[3]Grid Electric Power Corporation, Xi'an 710065, China
[4]School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[2]Corresponding author
E-mail: [1]*leelm@chd.edu.cn*, [2]*grwen@mail.xjtu.edu.cn*, [3]*rainy.wen@163.com*, [4]*birdy_dong@163.com*

(*Received 11 November 2012; accepted 28 February 2013*)

Abstract. In this article a genetic algorithm optimized Lagrangian support vector machine algorithm and its application in rolling bearing fault diagnosis is introduced. As an effective global optimization method, genetic algorithm is applied to find the optimum multiplier of Lagrangian support vector machine. Synthetic numerical experiments revealed the effectiveness of this genetic algorithm optimized Lagrangian support vector machine as a classifier. Then this classifier is applied to recognize faulty bearings from normal ones. Its performance is compared with that of backpropagation neural network and standard Lagrangian support vector machine. Experimental results show that the classification ability of our classifier is higher and the computing time required to find the separating plane is relative shorter.

Keywords: Lagrangian support vector machine, genetic algorithm, fault diagnosis, backpropagation neural network.

1. Introduction

Nowadays there is an explosion in the number of research papers on support vector machines (SVMs) [1-2], which have been successfully applied to a number of areas ranging from expression recognition [3], text categorization [4] to machine fault diagnosis [5-6] and image retrieval [7-8]. These references illustrate the effectiveness of SVM as a classification tool. In SVM, a typical quadratic program problem has to be solved to find the best separating plane. It is a computational very demanding process. O. L. Mangasarian and David R. Musicant (2001) give a fast simple algorithm named Lagrangian support vector machine (LSVM) [9], which is "based on an implicit Lagrangian formulation of the dual of a simple reformulation of the standard quadratic program of a linear support vector machine". Some other SVM algorithms [10-11] are proposed based on the principle of LSVM. In this article genetic algorithm is applied to find the optimum multiplier of Lagrangian support vector machine for its powerful global optimization ability. Synthetic data tests the effectiveness of the proposed algorithm. Finally, the genetic algorithm based LSVM is applied to diagnose rolling bearing fault and its performance is compared with that of backpropagation network and standard LSVM.

2. Principle of LSVM

Suppose we have m given observation points $\mathbf{x}_i \in \mathbf{R}^n, i = 1,2,\dots,m$. Denote these observations as a $m \times n$ matrix $\mathbf{A}$. A given $m \times m$ diagonal matrix $\mathbf{D}$ (some of the entries on the main diagonal are +1 and the rest are –1) reveals the membership of each point in the class A_+ or A_-. Support vector machine is applied to construct a hyperplane that separates the data into two given classes with maximal margin.

For standard support vector machine, the separating plane with a soft margin can be found by solving the following quadratic program problem:

$$\begin{aligned}&\min_{(\mathbf{w},\mathbf{r},\mathbf{y})\in\mathbf{R}^{n+1+m}} \nu\mathbf{e}^{\mathrm{T}}\mathbf{y}+\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w},\\ &\text{s.t.: } \mathbf{D}(\mathbf{A}\mathbf{w}-\mathbf{e}\mathbf{r})+\mathbf{y}\geq\mathbf{e}, \qquad \mathbf{y}\geq 0,\end{aligned} \tag{1}$$

where $\mathbf{w}$ is the normal vector of the separating plane, $\mathbf{r}$ determines the location of the plane relative to the origin, $\mathbf{y}$ is a non-negative slack variable, ν controls the tradeoff between errors on training data and margin maximization, $\nu > 0$, $\mathbf{e}$ is an unity vector.

The dual to the above quadratic programming problem is the following:

$$\begin{aligned}&\min_{\mathbf{u}\in\mathbf{R}^m}\left(\frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{D}\mathbf{A}\mathbf{A}^{\mathrm{T}}\mathbf{D}\mathbf{u}-\mathbf{e}^{\mathrm{T}}\mathbf{u}\right),\\ &\text{s.t.: } \mathbf{e}^{\mathrm{T}}\mathbf{D}\mathbf{u}=0, \qquad 0\leq\mathbf{u}\leq\nu\mathbf{e},\end{aligned} \tag{2}$$

where $\mathbf{u}$ is the Lagrangian multiplier.

For a support vector machine with nonlinear decision function in non-separable cases, the dual quadratic programming problem is defined as:

$$\begin{aligned}&\min_{\mathbf{u}\in\mathbf{R}^m}\left(\frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{D}\mathbf{K}(\mathbf{A},\mathbf{A}^{\mathrm{T}})\mathbf{D}\mathbf{u}-\mathbf{e}^{\mathrm{T}}\mathbf{u}\right),\\ &\text{s.t.: } \mathbf{e}^{\mathrm{T}}\mathbf{D}\mathbf{u}=0, \qquad 0\leq\mathbf{u}\leq\nu\boldsymbol{e},\end{aligned} \tag{3}$$

where $\mathbf{K}(x,y)$ is a kernel function. There are many popular kernels available, some common are polynomial, radial basis function (RBF), neural network kernels and so on.

To find the separating plane, a quadratic programming problem has to be solved, whereas such a problem is complex and computationally very demanding even though there are many existing methods to solve this problem [12] and SVM codes available online such as SVMlight [13] and libSVM [14]. So, O. L. Mangasarian and David R. Musicant (2001) presented a fast simple algorithm named Lagrangian support vector machine (LSVM) [9], in which "two simple changes are made to the standard SVM: (1) the margin between the parallel bounding planes is maximized with respect to both $\mathbf{w}$ and r; (2) the error $\mathbf{y}$ is minimized using the 2-norm squared instead of 1-norm". Based on these changes, a simpler positive definite dual problem with non-negativity constraints replaces the complex quadratic problem of standard SVM.

These two changes lead to the modification of Eq. (2):

$$\min_{0\leq\mathbf{u}\in\mathbf{R}^m}\frac{1}{2}\mathbf{u}^{\mathrm{T}}\left(\frac{\mathbf{I}}{\nu}+\mathbf{D}(\mathbf{A}\mathbf{A}^{\mathrm{T}}+\mathbf{e}\mathbf{e}^{\mathrm{T}})\mathbf{D}\right)\mathbf{u}-\mathbf{u}^{\mathrm{T}}\mathbf{e}, \tag{4}$$

where $\mathbf{I}$ is an identity matrix.

To simplify notation, we define two matrices as followings:

$$\mathbf{H}=\mathbf{D}[\mathbf{A}-\mathbf{e}], \qquad \mathbf{Q}=\frac{\mathbf{I}}{\nu}+\mathbf{H}\mathbf{H}^{\mathrm{T}}. \tag{5}$$

Then Eq. (4) can be rewritten as following:

$$\min_{0\leq\mathbf{u}\in\mathbf{R}^m}\frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{Q}\mathbf{u}-\mathbf{e}^{\mathrm{T}}\mathbf{u}. \tag{6}$$

The optimum $\mathbf{u}$ can be found by the following iterative expression:

$$\mathbf{u}(i+1)=\mathbf{Q}^{-1}(e+((\mathbf{Q}\mathbf{u}(i)-e)-\alpha\mathbf{u}(i))_{+}), i=1,2\ldots, \tag{7}$$

for which we will establish global linear convergence from any start point under the condition:

$$0 < \alpha < \frac{2}{v}, \tag{8}$$

where $u(i)$ denotes the i-th $\mathbf{u}$ during recursion, $(\mathbf{x})_+$ denotes a vector with all of negative components in $\mathbf{X}$ set to zero.

Like standard SVM, LSVM can also be extended to solve nonlinear classification problems. In nonlinear cases, we give $\mathbf{G}$ and $\mathbf{Q}$ different definitions:

$$\mathbf{G} = [\mathbf{A} - \mathbf{e}], \qquad \mathbf{Q} = \frac{\mathbf{I}}{v} + \mathbf{DK}(\mathbf{G}, \mathbf{G}^{\mathrm{T}})\mathbf{D}. \tag{9}$$

Just as the linear cases, we can obtain the same iterative scheme as Eq. (7).

3. Genetic Algorithm for the Optimization of LSVM

The above descriptions reveal that the essence of LSVM is to find an optimum Lagrangian multiplier $\mathbf{u}$ that minimizes Eq. (6) under constraint condition of $\mathbf{u} \geq 0$. In this section, we try to apply genetic algorithm to solve this problem.

In this article two kinds of genetic algorithm are designed: one is the standard genetic algorithm, named as SGA, and the other is a modified genetic algorithm, denoted by MGA. Both algorithms adopt float encoding and arithmetical crossover operation [15-16]. The difference of these algorithms mainly lies in the mutation operation, which is the main search operation when float encoding is adopted.

Three major steps in executing genetic algorithm (both SGA and MGA) to find the optimum $\mathbf{u}$ is as following:

A. Generate initial population.

As the dimension of optimization variable $\mathbf{u}$ equals to the amount of data points to be classified, it is obviously a high dimensional optimization problem. Float encoding is more suitable for this situation. The uniform distribution is used to generate initial individuals of SGA, while individuals of MGA are formed by superimposing perturbation values upon variable $\mathbf{C}$ which is generated based on the following expression:

$$\mathbf{C} = \mathbf{Q}^{-1}\mathbf{e}. \tag{10}$$

B. Iteratively perform the following substeps on the population until the termination criterion is satisfied. The maximum number of generation to be run is determined beforehand.

(1) Assign a fitness value to each individual in the population using the fitness measure.

For a constrained optimization problem defined as Eq. (6), the fitness function with penalty function is constructed to measure the feasibility of individuals:

$$f_i = -\frac{1}{2}\mathbf{I}_i^{\mathrm{T}}\mathbf{Q}\mathbf{I}_i + \mathbf{e}^{\mathrm{T}}\mathbf{I}_i + \beta\sum_{j=1}^{n}|\min\{0, \boldsymbol{I}_i j\}|, \tag{11}$$

where f_i is the fitness value of individual $\mathbf{I}_i$ β is the penalty factor, $\beta > 0$ and n is the size of $\mathbf{I}_i$.

(2) Create a new population by applying the following three genetic operations.

Create two new individuals using crossover operation. Suppose $\mathbf{P}^{(1)} = (p_1^{(1)}, p_2^{(1)} \ldots . p_n^{(1)})$, $\mathbf{P}^{(2)} = (p_1^{(2)}, p_2^{(2)} \ldots . p_n^{(2)})$ are the selected parents, the sons $\mathbf{s}^{(1)} = (s_1^{(1)}, s_2^{(1)} \ldots . s_n^{(1)})$, $\mathbf{s}^{(2)} = (s_1^{(2)}, s_2^{(2)} \ldots . s_n^{(2)})$ are formed based on the following expression:

$$\begin{cases} s_i^1 = \gamma p_i^1 + (1-\gamma)p_i^2, \\ s_i^2 = \gamma p_i^{(2)} + (1-\gamma)p_i^1, \end{cases} \tag{12}$$

where γ is a random number, $0 \le \gamma \le 1$.

Create a new individual by mutation operation. In SGA, non-uniform mutation operation is adopted. It is operated as following:

Suppose $\mathbf{P} = (p_1, p_2, \dots, p_n)$ is the selected parent, the component p_k whose definition interval is $[a_k, b_k]$, is selected to mutate. Then after mutation, the component would change as following:

$$p_k' = \begin{cases} p_k + \Delta(t, b_k - p_k) & \text{if ROUND}(R_{nd}) = 0, \\ p_k - \Delta(t, p_k - a_k) & \text{if ROUND}(R_{nd}) = 1, \end{cases} \tag{13}$$

where variable R_{nd} is a random value in the range $(0, 1)$, and function ROUND rounds R_{nd} to the nearest integer, t is the current number of generation, function $\Delta(t, y)$ is defined as following:

$$\Delta(t, y) = y\left(1 - a^{\left(1-\frac{t}{T}\right)^{\lambda}}\right), \tag{14}$$

where a is a random number in the range $[0, 1]$, T is the biggest generation number, λ is a constant.

In MGA, some knowledge of the function being optimized is put into the operation design. The global search ability of genetic algorithm and local search ability of LSVM are combined to form more powerful search operators. After mutation, the offspring of individual **P** of MGA is the following:

$$\mathbf{P}' = \mathbf{Q}^{-1}(\mathbf{e} + ((\mathbf{QP} - \mathbf{e}) - \alpha\mathbf{P})_+). \tag{15}$$

Select better existing individuals to form a new population. In order to avoid premature, ranking selection is adopted by both algorithms.

C. When evolution finished, the best individual of the final population is then the optimum **u**.

4. Synthetic Numerical Implementation

In this section the effectiveness of proposed algorithms are tested using synthetic data. In order to simplify description, SGA optimized Lagrangian support vector machine is abbreviated as SGA_LSVM, MGA optimized Lagrangian support vector machine is abbreviated as MGA_LSVM.

All of our experiments are run on a personal computer, which utilizes a 3.2 GHz dual-core processor and a maximum of 2GB of memory.

First we try the linear case. For linear case, synthetic data are generated to test our algorithm, shown in Fig. 1. These data points are divided into two data sets: half for building the classifier, the other half for testing the classifier. The parameters for ideal separating plane should be $\mathbf{w} = [1, 1]$ and $r = 1$ Table 1 shows the results.

If we set $r = 1$, then **w** obtained by SGA_LSVM is [0.9318, 0.9518], **w** of MGA_LSVM is [1.0364, 1.0163]. **w** and r we got are very close to the ideal values. The training correctness and testing correctness reveal the effectiveness of our algorithms as classifier.

Then nonlinear classification ability of our algorithms is tested. Fig. 2 shows the data points for the nonlinear case. Table 2 shows the outputs of our algorithms as classifier with RBF kernel.

The results show the feasibility of our algorithms.

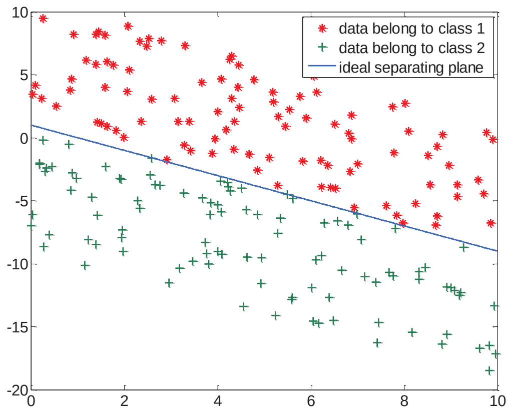


Fig. 1. Synthetic data for linear case

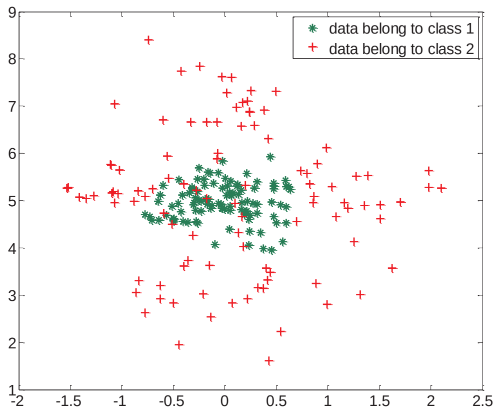


Fig. 2. Data points for nonlinear case

Table 1. Results of genetic algorithm based support vector machine

Classifier	Training correctness (%)	Testing correctness (%)	Computing time (sec)	**w**	r
SGA_LSVM	97	98	1.823	[2.4776, 2.5307]	2.6587
MGA_LSVM	99	100	0.470	[0.5465, 0.5358]	0.5272

Table 2. Results of our classifier with RBF kernel

	Training correctness (%)	Testing correctness (%)	Computing time (sec)
SGA_LSVM	89.1	87.2	1.549
MGA_LSVM	90.0	87.8	0.418

Table 1 and Table 2 reveal that, for the same problem, MGA_LSVM requires shorter computing time than SGA_LSVM. That means the convergence speed of MGA is higher than that of SGA. The reason lies in the difference of mutation operation. As mentioned above, in MGA, some knowledge of the function being optimized is put into the operation design, thus the convergence speed of MGA is greatly accelerated.

5. Diagnostic Practice

In this section, our algorithm is applied to real world problem of rolling bearing faults diagnosis. In order to test the classification ability of our algorithm, backpropagation (BP) network and standard LSVM classifier are also applied to solve the same problem: recognize three types of rolling bearing faults from the normal condition.

In order to verify the performance of proposed method, experiment is conducted on the rolling bearing test rig (Fig. 3). Key phase signal is collected from the input axial with photoelectric sensor. Vibration signals are collected from top and side position of testing bearing. Sampling frequency is set to be 12000 Hz in the experiment. The range of speed fluctuation of the input shaft is 800 rpm~1500 rpm during the experiment.

The four working conditions of rolling bearing are: normal condition, inner race flake, outer race flake and rolling element flake. Skewness factor and shape factor are selected as classification features. Fig. 4 – Fig. 6 show the training data and testing data under different working conditions. It is apparently a nonlinear classification problem.

Table 3 – Table 5 show the experimental results obtained by different classifiers. BP is the worst classifier with respect to classification correctness as well as computing time. It performs especially badly when there is a serious aliasing between two classes, such as the case of

classifying normal condition and rolling element flake, in which the classification correctness is less than 50 %.

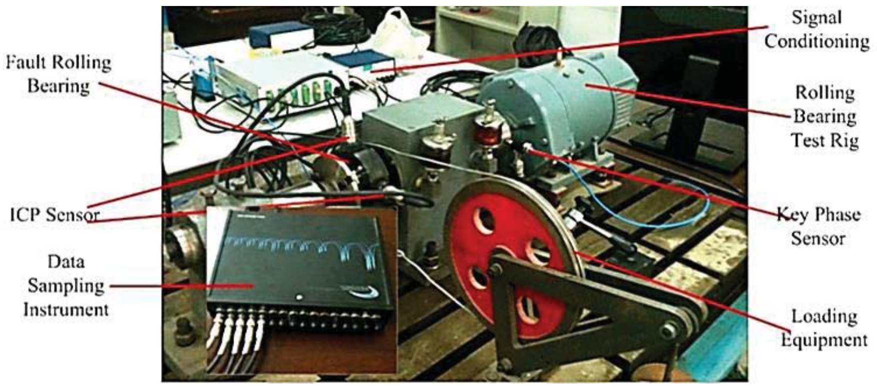


Fig. 3. Test rig for rolling bearing experiment

Table 3. Comparison of recognition ability and computing time of different classifiers
Condition: normal, outer race flake

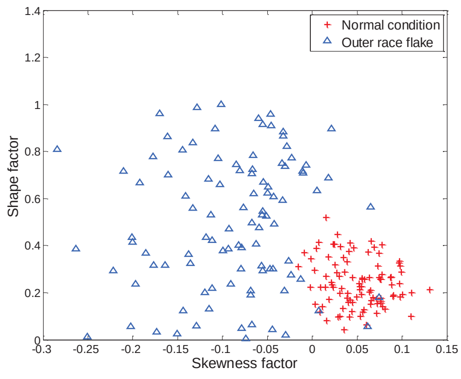
	SGA_LSVM	MGA_LSVM	LSVM	BP
Training correctness (%)	92.5	97.5	98.5	86.5
Testing correctness (%)	91.5	94.5	92.5	85.0
Computing time (sec)	6.213	1.236	9.591	73.955

Table 4. Comparison of recognition ability and computing time of different classifiers
Condition: normal, inner race flake

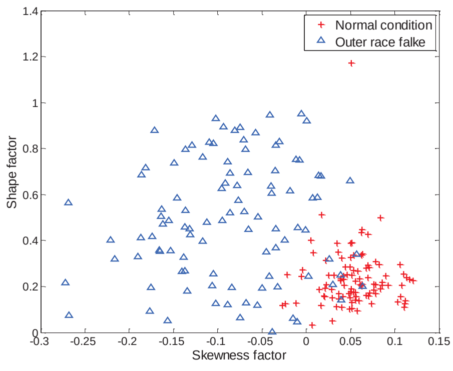
	SGA_LSVM	MGA_LSVM	LSVM	BP
Training correctness (%)	85.5	89.0	90.0	64.0
Testing correctness (%)	84.5	86.5	86.0	62.0
Computing time (sec)	6.108	1.326	10.188	68.450

Table 5. Comparison of recognition ability and computing time of different classifiers
Condition: normal, rolling element flake

	SGA_LSVM	MGA_LSVM	LSVM	BP
Training correctness (%)	74.5	78.5	81.0	28.0
Testing correctness (%)	75.5	78.5	74.0	32.5
Computing time (sec)	5.982	1.452	10.235	69.658

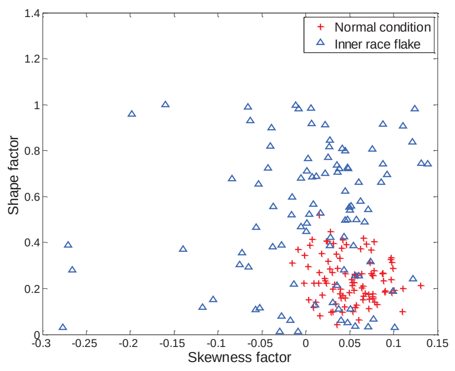


(a) training data

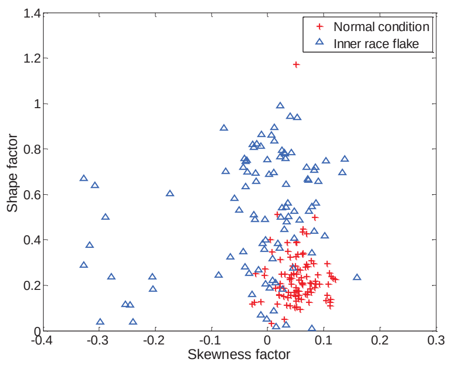


(b) testing data

Fig. 4. Experimental data for distinguishing between normal condition and outer race flake

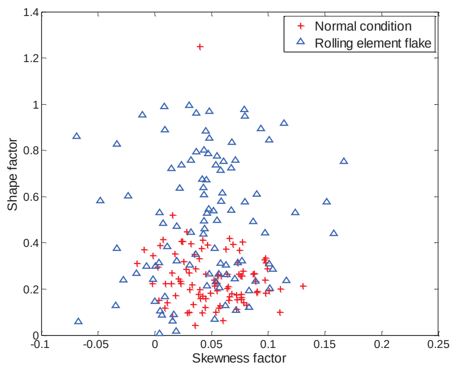


(a) training data

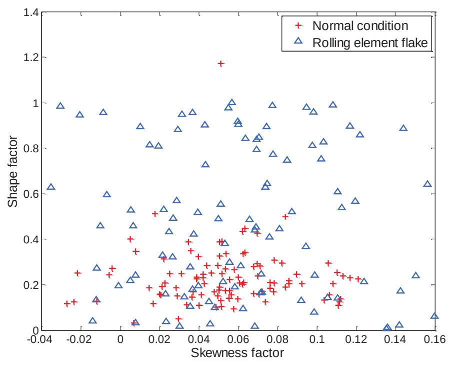


(b) testing data

Fig. 5. Experimental data for distinguishing between normal condition and inner race flake



(a) training data



(b) testing data

Fig. 6. Experimental data for distinguishing between normal condition and rolling element flake

For a classifier, we care more about testing correctness which represent generalization ability of a classifier than training correctness. For the other three classifiers, the classification ability of MGA_LSVM classifier is slightly better than that of standard LSVM, while that of SGA_LSVM is relatively worse. As for computing time, MGA_LSVM requires the shortest computing time, and then the SGA_LSVM, standard LSVM requires the longest time. These observations show that MGA _LSVM performs the best compared with three other classifiers.

6. Conclusion

In this article genetic algorithm is introduced to solve the optimization of Lagrangian support vector machine, and is applied to practice diagnostic problem. Two kinds of genetic algorithm are designed, denoted by SGA and MGA respectively. The operation is described in detail. Synthetic numerical experiments revealed the effectiveness of these two GA optimized LSVM classifiers, it also show that the convergent speed of MGA is faster than that of SGA. Then these classifiers are applied to recognize fault bearings from normal ones. Their performance is compared with that of neural network and standard Lagrangian support vector machine. Experimental results show that MGA_LSVM is the best classifier compared with three other classifiers with respect to both recognition ability and computing time.

Acknowledgements

This project was accomplished thanks to the funds of National Natural Science Foundation of China (No. 51175049) and Special Fund for Basic Scientific Research of Central Colleges (No. CHD2011JC025).

References

[1] **V. Vapnik** The Nature of Statistical Learning Theory. Springer Verlag, New York, Inc., USA, 1995.

[2] **Javier M., Alberto M.** Support vector machines with applications. Statistical Science, Vol. 21, Issue 3, 2006, p. 322-336.

[3] **Yani Z., Jiayou D., Jiatao S.** Expression recognition based on genetic algorithm and SVM. Fuzzy Information and Engineering, Vol. 62, 2009, p. 743-751.

[4] **Taiyue W., Hueimin C.** Fuzzy support vector machine for multi-class categorization. Information Processing & Management, Vol. 43, Issue 4, 2007, p. 914-929.

[5] **Achmad W., Bo S. Y.** Support vector machine in machine condition monitoring and fault diagnosis. Mechanical Systems and Signal Processing, Vol. 21, Issue 6, 2007, p. 2560-2574.

[6] **Junhong Z., Yu L., Feng R. B.** Diesel engine valve fault diagnosis based on LMD and SVM. Transactions of CSICE, Vol. 30, Issue 5, 2012, p. 469-473.

[7] **Jian C., Kong Q. W.** Active learning for image retrieval with Co-SVM. Pattern Recognition, Vol. 40, Issue 1, 2007, p. 330-334.

[8] **Xiaohong S., Jin D., Peijun M.** Image retrieval by convex hulls of interest points and SVM-based weighted feedback. Chinese Journal of Computers, Vol. 32, Issue 11, 2009, p. 2221-2228.

[9] **Mangasarian O. L., David R.** Lagrangian support vector machines. Journal of Machine Learning Research, Vol. 9, Issue 1, 2001, p. 161-177.

[10] **Jae Pil H., Seongkeun P., Euntai K.** Dual margin approach on a Lagrangian support vector machine. International Journal of Computer Mathematics, Vol. 88, Issue 4, 2011, p. 695-708.

[11] **Xiaowei Y., Lei S., Zhifeng H.** An extended Lagrangian support vector machine for classifications. Progress in Natural Science, Vol. 14, Issue 6, 2004, p. 519-523.

[12] **Kristin P., Colin C.** Support vector machines: Hype or Hallelujah?. SIGKDD Explorations, Vol. 2, Issue 2, 2000, p. 1-13.

[13] **Joachims T.** SVMlight. http://svmlight.joachims.org/, 2010.

[14] **Chih C. C., Chih J. L.** http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/, 2001.

[15] **Randy L. H., Sue E. H.** Practical Genetic Algorithm. Wiley-Interscience, 2004.

[16] **Agoston E. E., Smith J. E.** Introduction to Evolutionary Computing. Springer, 2008.