Faults Diagnosis based on Support Vector Machines and Particle Swarm Optimization

Chenghua SHI, Yapeng WANG, Honglei ZHANG

School of Economics and Management, Hebei University of Engineering, Handan 056038, China, chenghuashi@hebeu.edu.cn

College of Economics & Management, Huazhong Agricultural University, Wuhan 430070, China, hzndwangyapeng@163.com
doi : 10.4256/ijact.vol3.issue5.8

Abstract

Faults diagnosis is essentially one of pattern recognition problems. It has been gaining more and more attention to develop methods for improving the accuracy and effectiveness of pattern recognition. Support vector machine (SVM) is a powerful technique for the classification problems with small sampling, nonlinear and high dimension. However, one important problem encountered in setting up SVM models is how to determine the values of their parameters. The paper examined the diagnosis effects of SVMs with default and chosen parameters on the Steel Plates Faults Data Set, showing that different parameters may produce different diagnosis results. Particle swarm optimization (PSO), which is a heuristic method that optimizes a problem by iteratively trying to improve the candidate solution, was applied to optimize the parameters of SVMs, which enhanced the diagnosis accuracy.

Keywords: Faults Diagnosis, Support Vector Machines, Parameters Optimization, Particle Swarm Optimization

1. Introduction


Support vector machines (SVMs), invented by Vladimir Vapnik [9], is a novel machine learning method based on statistical learning theory, which is powerful for the problem with small sampling, nonlinear and high dimension. Owing to the advantages of SVMs, the technology has been used in various fields [10-12], including faults diagnosis. Lv et al [13] applied a multi-layer SVM classifier to diagnose faults of power transformer and test results showed that the classifier had an excellent performance on training speed and reliability. Yélamos et al [14], Cui and Wang [15], Saimurugan et al [16] and other researchers also contributed their efforts to the use of SVMs in faults diagnosis.

However, one important problem encountered in setting up SVM models is how to determine the values of their parameters and inappropriate parameter settings may lead to poor classification results [17]. Limited literatures have focused on the parameters optimization of SVMs. The paper tries to do some efforts for the issue. The remainder of this paper is organized as follows. Section 2 briefly introduces related knowledge about SVMs. Section 3 develops the SVM approach for faults diagnosis on Steel Plates Faults without parameters optimization. Section 4 applies particle swarm optimization (PSO) to optimize the parameters of SVMs in the faults diagnosis. Conclusions are finally drawn in
Section 5, along with recommendations for future research.

2. Support vector machines

SVM is a learning machine based on the statistical learning theory which embodies the Structural Risk Minimization (SRM) principle shown superior to the Empirical Risk Minimization (ERM) principle [9]. The machine was first proposed for two-group classification problem, and has been gaining much popularity in various fields due to its attractive features.

For the classification problems, the realization process of SVM is that input vectors are firstly mapped into a high-dimensional space through a nonlinear mapping, and then a hyperplane is constructed and is moved until a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class.

The classification problem can be restricted to the two-class classification problem without loss of generality. Here the goal is to separate the two classes by a function which is induced from available samples. Given that \( T=\{(x_1,y_1),\ldots,(x_n,y_n)\} \) is the training set; \( x_j \in \mathbb{R}^d \) are input vectors; \( y \in \{-1,1\} \) are classification labels; \( <\omega,x>+b=0 \) embodied a hyperplane. SVMs require the solution of the following optimization problem:

\[
\min_{\omega,b,\xi} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{n} \xi_i \quad (1)
\]

\[
\text{S.T.} \quad y_i(\omega \cdot x_i) + b \geq 1, i=1,2,\ldots,n \quad (2)
\]

where \( C>0 \) is the penalty parameter; \( \xi=(\xi_1,\xi_2,\ldots,\xi_n) \) represents the classification errors for the training dataset. The corresponding dual problem of (1)-(2) is as follows:

\[
\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{n} \alpha_i \quad (3)
\]

\[
\text{S.T.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \quad (4)
\]

\[
0 \leq \alpha_i \leq C, i=1,2,\ldots,n \quad (5)
\]

where \( \alpha_i \geq 0 \) represents the Lagrange Multipliers.

Whereas the original problem may be stated in a finite dimensional space, it often happens that in that space the datasets to be classified are linearly inseparable. For this reason it was proposed that the original finite dimensional space be mapped into a much higher dimensional space presumably making the separation easier in that space. Kernel functions \( K(x_i,x_j) = \phi(x_i) \cdot \phi(x_j) \) are defined to suit the problem. Though various kernel functions have been proposed by researchers, the most widely used ones are the following four kinds:

1. Linear kernel function \( K(x_i,x_j) = x_i \cdot x_j^T \);
2. Polynomial kernel function \( K(x_i,x_j) = (x_i \cdot x_j + r)^\gamma, \gamma > 0, r = 0 or 1 \);
3. Radial basis function \( K(x_i,x_j) = \exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right) \);
(4) Sigmoid kernel function \( K(x_i, x_j) = \tanh(\gamma x_i \cdot x_j^T + r) \).

where \( \gamma \), \( r \), \( \sigma \) and \( d \) are kernel parameters.

Once the kernel function \( K(x_i, x_j) \) is selected, the optimization problem (3)-(5) is transformed into the following model:

\[
\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i \cdot x_j) - \sum_{i=1}^{n} \alpha_i
\]

Subject to:

\[
\sum_{j=1}^{n} y_j \alpha_j = 0
\]

\[
0 \leq \alpha_i \leq C, i = 1,2,\ldots,n
\]

After solving the above problem, we can gain the value of \( \alpha_i^* \), and the decision-making function can be:

\[
f(x) = \text{sgn} \left( \sum_{i=1}^{n} y_i \alpha_i^* K(x \cdot x_i) + b^* \right)
\]

where \( b^* = y_j - \sum_{i=1}^{n} y_i \alpha_i^* K(x_i, x_j) \) \( j \in \{ j | 0 < \alpha_j^* < C \} \).

3. Faults diagnosis based on SVMs

3.1. Description of the faults diagnosis dataset

The Steel Plates Faults Data Set used in the paper comes from the UCI Machine Learning Repository [18]. Steel Plates Faults Data Set is one of the datasets in the Repository, which classifies steel plates’ faults into 7 different types: Pastry, Z_Scratch, K_Scatch, Stains, Dirtiness, Bumps and Other_Faults. The goal was to train machine learning for automatic pattern recognition. The dataset includes 1941 instances, which have been labeled by different fault types. The detailed information and the whole dataset can be accessed from http://archive.ics.uci.edu/ml/datasets/Steel+Plates+Faults. The dataset was donated by Semeion, Research Center of Sciences of Communication, Via Sersale 117, 00128, Rome, Italy. Buscema et al first used the dataset in July 2010 [19]. Each instance of the dataset owns 27 independent variables and one fault type.

3.2. Faults diagnosis modeling based on SVM

According to SVM theory, we can conclude the basic process of faults diagnosis modeling based on SVM, shown as Figure 1.
3.2.1. Selecting the training dataset and testing dataset

<table>
<thead>
<tr>
<th>Types of Faults</th>
<th>Labels of Faults</th>
<th>Instances distribution in the original dataset</th>
<th>972 instances for the training dataset</th>
<th>969 instances for the testing dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pastry</td>
<td>1</td>
<td>1–158</td>
<td>1–79</td>
<td>80–158</td>
</tr>
<tr>
<td>Z_Scratch</td>
<td>2</td>
<td>159–348</td>
<td>159–253</td>
<td>254–348</td>
</tr>
<tr>
<td>K_Scatch</td>
<td>3</td>
<td>349–739</td>
<td>349–544</td>
<td>545–739</td>
</tr>
<tr>
<td>Stains</td>
<td>4</td>
<td>740–811</td>
<td>740–775</td>
<td>776–811</td>
</tr>
<tr>
<td>Dirtiness</td>
<td>5</td>
<td>812–866</td>
<td>812–839</td>
<td>840–866</td>
</tr>
<tr>
<td>Bumps</td>
<td>6</td>
<td>867–1268</td>
<td>867–1067</td>
<td>1068–1268</td>
</tr>
<tr>
<td>Other_Faults</td>
<td>7</td>
<td>1269–1941</td>
<td>1269–1605</td>
<td>1606–1941</td>
</tr>
</tbody>
</table>

The 1941 instances in the Steel Plates Faults Data Set are classified into seven faults: Pastry, Z_Scratch, K_Scatch, Stains, Dirtiness, Bumps and Other_Faults, which are labeled by 1, 2, 3, 4, 5, 6 and 7 in the paper. The Instances distribution in the original dataset is shown as Table 1. The paper chose 972 instances for the training dataset and 969 instances for the testing dataset.

3.2.2. Data preprocessing

The paper mainly normalized the training dataset before inputting them into SVMs. The normalization mapping is as follows:

\[ f : x \rightarrow y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{10} \]

where \( x \) is the value of some variable, \( x_{\min} = \min(x) \), \( x_{\max} = \max(x) \) and \( y \) is the normalized value which belongs to \([0, 1]\).

3.2.3. Training SVM model

After being preprocessed, the normalized values can be input into the SVMs to train the classifier. The SVM toolbox that the paper used is the Libsvm toolbox developed by Chang and Lin [20]. Libsvm is integrated software for support vector classification (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification. Although inappropriate parameter settings may lead to poor classification results, there are no effective methods to setup the parameters rationally. However, the Libsvm toolbox has provided many default parameters which may deal with some problems. So the paper first applied default parameters to train the SVM and then randomly chose some parameter values to validate the effects of the parameters. The kernel function of SVM applied in the paper is the Radial Basis Function. Input the normalized values of the 972 instances in the training dataset, we can train the SVM model.
3.2.4. Faults diagnosis based on the trained SVM

By train the SVM, we can get the faults diagnosis model which can be used to diagnose the fault of each testing instance. The output result is shown as Figure 2, compared with actual faults. 520 of 969 instances are identified right, which means the accuracy is 53.6636%.

![Figure 2. Actual faults and diagnosed faults of the testing dataset with default parameters](image)

3.3. Results analysis

The results shown in the Figure 3 are produced with default parameters. Now we choose some different parameter values to see the diagnosis results. There are four kernel functions: Linear, Polynomial, Radial basis function and Sigmoid mentioned in the Session 2, and the key parameters \(c\) and \(g\) also have important effects on the classifier. Table 2 gives the recognition results with different parameters. As we can see, different kernel functions may result in different results with the same \(c\) and \(g\), and different \(c\)s and \(g\)s may also produce different diagnosis results with the same kernel function.

<table>
<thead>
<tr>
<th>Table 2. Results with different parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The kernel function</strong></td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Linear</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Polynomial</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Radial basis function</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

3.4. Parameters optimization of faults diagnosis SVM based on cross validation

As Table 2 shows, SVMs’ parameters are crucial for their diagnosis accuracies. The diagnosis results SVMs produce with default parameters may be satisfactory, but relevant parameters need to be adjusted to get more satisfactory accuracies. Then, how to choose the best parameters is the issue the following sections try to solve. In the part, the paper uses Cross Validation (CV) method to optimize the parameters (mainly the penalty parameter \(c\) and the kernel function parameter \(g\)) of RBF-SVM.
Faults Diagnosis based on Support Vector Machines and Particle Swarm Optimization

Chenhua SHI, Yapeng WANG, Honglei ZHANG

International Journal of Advancements in Computing Technology Volume 3, Number 5, June 2011

Figure 3. The whole process of faults diagnosis based on SVM with parameters optimization by CV

CV is a statistical analysis method used to verify the performance of classifiers. The basic idea is that the original dataset is divided into training datasets which are used for training classifiers, and validation datasets for testing the trained models to obtain the classification accuracy as the evaluation performance of classifiers. Several common CV methods are Hold-Out Method, K-fold Cross Validation and Leave-One-Out Cross Validation. The paper mainly applies the K-CV to optimize the parameters of SVMs. The training dataset and testing dataset is the same as the datasets mentioned in 3.1. The whole process of faults diagnosis based on SVMs and Cross Validation is shown in Figure 3.

First, in order to cover the optimal parameters, a rough range is chose. The ranges of $c$ and $g$ are from $2^{-10}$ to $2^{10}$. The selection results are shown as Figure 4. The best CV accuracy is 75.7202% with the best $c$ 12.1257 and $g$ 0.25. It took 984.21 seconds in all.

According to the contour map and 3D view in Figure 4, the paper contracted the ranges of $c$ into from $2^0$ to $2^{10}$, and $g$ from $2^{-6}$ to $2^{2}$. Fine selection results are shown as Figure 5 with the run time being 132.93 seconds. Best cross validation accuracy equals 75.823%, the best $c$ 32 and the best $g$ 0.176777.
After getting the best $c$ and $g$, we can use them to diagnose the faults of the instances in the testing dataset. The diagnosis results are shown as Figure 6. The faults diagnosis accuracy is 60.7843% (589/969), which is higher than any of those in the Table 2. It can be seen that the CV thoughts can attain the optimal parameters in some sense, which may effectively avoid the overlearning and underlearning. A better accuracy can be achieved for the testing dataset. Example results show that the trained SVM models with the selected parameters by CV methods are more effective than the models trained with randomly selected parameters.

4. Parameters optimization of faults diagnosis SVM based on particle swarm optimization

Although CV methods can find the best $c$ and $g$ with the highest classification accuracy by the grid search function, sometimes it will be very time-consuming if you want to find the best parameters in a wider range. Heuristic algorithms need not traverse all the parameter values within the grid to find the global optimal solutions.

4.1. Particle swarm optimization

Particle Swarm Optimization (PSO) is a heuristic method that optimizes a problem by iteratively trying to improve a candidate solution. PSO algorithm first initializes a population of particles in the solution space. Each particle represents a candidate solution of the optimization problem and is characterized by the position, velocity and fitness value. The fitness values are the merits of the particles. These particles are moved around in the search-space. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm’s best known
position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Given the search-space is \( D \) dimensions and a population \( X=\{X_1, X_2, \ldots, X_n\} \) is composed by \( n \) particles. \( X_i=(X_{i1}, X_{i2}, \ldots, X_{id}) \), \( i=1,2,\ldots,n \) is a \( d \)-dimensional vector, representing a candidate solution. According to the fitness function, the fitness value of \( X_i \) can be calculated. \( V_i=(V_{i1}, V_{i2}, \ldots, V_{id}) \) is the velocity of the \( i \)-th particle. \( P_i=(P_{i1}, P_{i2}, \ldots, P_{id}) \) represents the particle's best known position and \( P_g=(P_{g1}, P_{g2}, \ldots, P_{gd}) \) is the swarm's best known position. In each iteration, each particle updates its velocity \( V_i \) and position \( X_i \) according to the following formula:

\[
V_{id}^{k+1} = \omega V_{id}^k + c_1 r_1 (P_{id}^k - X_{id}^k) + c_2 r_2 (P_g^k - X_{id}^k)
\]

\[
X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1}
\]

where \( \omega \) is the inertia weight; \( d=1,2,\ldots,D \); \( i=1,2,\ldots,n \); \( k \) is the current iteration; \( V_{id} \) is the particle velocity; non-negative constants \( c_1 \) and \( c_2 \) are the acceleration parameters; \( r_1 \) and \( r_2 \) are two random numbers between 0 and 1.

**Figure 7.** The whole process of faults diagnosis based on SVM with parameters optimization by PSO

### 4.2. Algorithm design of optimizing parameters of SVMs based on PSO

According to the algorithm of PSO, the paper designed the whole process of faults diagnosis based on SVM with parameters optimization by PSO, as Figure 7 shows. The accuracy of the training instance by CV is taken as the fitness function of PSO. Related parameters settings of PSO and SVM are shown as Table 3.
Faults Diagnosis based on Support Vector Machines and Particle Swarm Optimization
Chenghua SHI, Yapeng WANG, Honglei ZHANG
International Journal of Advancements in Computing Technology Volume 3, Number 5, June 2011

Table 3. Related parameters settings of PSO and SVM

<table>
<thead>
<tr>
<th>Parameters settings of PSO</th>
<th>Parameters settings of SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>Initial values</td>
</tr>
<tr>
<td>$c_1$</td>
<td>1.5</td>
</tr>
<tr>
<td>$c_2$</td>
<td>1.7</td>
</tr>
<tr>
<td>maxgen</td>
<td>200</td>
</tr>
<tr>
<td>sizepop</td>
<td>20</td>
</tr>
<tr>
<td>$k$</td>
<td>0.6</td>
</tr>
<tr>
<td>$w$</td>
<td>1</td>
</tr>
</tbody>
</table>

4.3. Results analysis
The parameters optimization process and results are shown as Figure 8. Best validation accuracy equals 74.7942\%, best $c$ 4.76824, and best $g$ 1.41338 after 112.3 seconds. The faults diagnosis accuracy of the testing dataset is 60.7843% (589/969), which is equal to the optimal accuracy attained by CV method, but with smaller $c$ and $g$.

4.5. Conclusions
Faults diagnosis, which is essentially a pattern recognition problem, plays an important role in the operation and maintenance of mechanical equipment. Support vector machine provide a good technique for enhancing the accuracy of faults identification. However, the largest problem encountered in setting up SVM models is how to select their kernel functions and the values of related parameters. The paper examined the diagnosis effects of SVMs with default and chosen parameters on the Steel Plates Faults Data Set. Support vector machines with appropriate parameters can provide a good tool for enhancing the diagnosis accuracy. Different kernel functions may result in different results with the same parameter, and different parameters may also produce different diagnosis results with the same kernel function.

Particle Swarm Optimization (PSO) is a heuristic method that optimizes a problem by iteratively trying to improve a candidate solution. PSO does not require for the optimization problem to be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. The best parameters of faults diagnosis SVM was achieved by PSO, which yielded the same accuracy with CV method. Although the paper produced satisfactory diagnosis results by applying PSO to select the best parameters of SVM, the choice of PSO parameters is not well considered, which may have a large impact on optimization performance. How to select the appropriate parameters of PSO when it is used to optimize the SVMs’ models needs further works.
6. References