Double-layer Support Vector Machine for Robust and Efficient Classification

Yuting Wu1, Yigang He1,2*, Luqiang Shi1, Yongbo Sui1, Yuan Huang1 and Tongtong Cheng1

1 School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, 230009, China
2 The School of Electrical Engineering, Wuhan University, Wuhan, 430072, China
Email: 18655136887@163.com
*Corresponding Author

Abstract. In this paper, double-layer support vector machine (DLSVM) for binary classification is proposed that can expeditiously learn a classification model under the premise of ensuring accuracy. In the first layer, least squares support vector machine (LSSVM) is used to evaluate all samples in the data set. Support vector machine (SVM) utilize a sparse data set to train the binary classifier in the second layer. In order to obtain excellent sparse data set for SVM, the normalized norm sequence and the normalized Lagrange multiplier Shannon entropy sequence are introduced. DLSVM provides low computation cost because the two-layer structure reduces lots of iterative operations. Experimental results show that the proposed method not only has similar classification accuracy compared to SVM but also has higher efficiency.

1. Introduction

As one of the most popular supervised classification methods, Support Vector Machine (SVM) has been widely used in different approaches such as fault diagnosis[1]-[2], image and video processing [3]-[4] and information retrieval[5] etc. Traditional classification methods minimize loss function to calculate the parameters of the separating hyperplane. SVM tries to maximize margin to find best-fit separating hyperplane. The points closest to the separating hyperplane are known as support vectors and the distance between support vectors and the separating hyperplane is known as margin. Hence, the SVM can achieve greater prediction accuracy than other traditional supervised classifiers with a small training set [6]. The optimization problem of SVM classifier involves inequality constraints, and the sequential minimal optimization (SMO) algorithm [7] is used to solve the optimization problem. In SMO based method, the Lagrange multiplier is calculated iteratively and support vectors is selected with respect to them. All training samples will be sent to train the SVM classifier, however, only the support vectors really work for classification results.

In many applications that deals with large data sets or requires high real-time performance [9]-[10], the classification methods should overcome some computational difficulties. Several works in the literature address the time consuming issue of learning SVM model. They can be divided into two categories:

1) Linear Methods: Different from solving a quadratic programming problem with inequality constraints in standard SVM, the optimization problem is transformed into linear problem with equality constraints.

2) Reduced Methods: Eliminating those less important training samples to generate a smaller
kernel matrix, the thin kernel matrix is used to replace the full kernel matrix. As the most typical linear method, least squares support vector machine [11] has equality constraints and \( L_2 \)-loss function. The solution follows from a linear Karush-Kuhn-Tucker [12] system. The optimization problem of LSSVM can be solved efficiently. However, all the training sample becomes support vectors in LSSVM case. The lack of sparsity leads to a decrease in prediction accuracy. To improve the accuracy of LSSVM, the key is to find optimal support vectors. Pruning those less meaningful training samples and the LSSVM model is re-trained by the remaining data points [12]-[13]. Pruning LSSVM models reduces the time requirements compared to SVM at the cost of accuracy, because those less meaningful data points cannot be precisely pruned. Furthermore, LLSVM (locally linear support vector machines) is approximated by using local coding. The model can be optimized by performing stochastic gradient [14]. The accuracy of LLSVM is better than that of LSSVM based methods, but the optimization process is iterative and time-consuming.

In recent years, reduced SVM methods have been studied by several researchers. In RSVM (reduced support vector machine), a full dataset is replaced to a uniform random subset to generate a thin rectangular kernel matrix [15]. The computational time of RSVM is much less than that of a standard SVM with the full kernel matrix. However, the performance of RSVM is not stable for its randomness. RW-SVM (reduced weighted support vector machine) reduces the training set to a smaller size via clustering method. In subsequent supervised learning, the loss function is modified by weighted training samples [16]. The data points of the smaller training set may not belong to the original data set, which will affect the final classifier performance.

In this paper, we propose an improved SVM model with double layered structure for robust and efficient classification. In the first layer, IWLSSVM is used to evaluate all data points. In IWLSSVM, an improved RBF kernel matrix with the normalized \( L_2 \) sample distance is used to adapt to values that lie in different ranges. In the second layer, the normalized Lagrange multiplier Shannon entropy is used to eliminate those less important data points and the sparse training set is utilized to train SVM model. DL-SVM inherits both merits of linear methods and reduced methods. The proposed classification model is evaluated on multiple binary datasets from the UCI repositories. The results show that the classification accuracy of DL-SVM not inferior to SVM while maintaining higher efficiency.

2. Review of SVM and LSSVM

In this section, we briefly review the SVM model with inequality constraints and the LSSVM model with equality constraints.

2.1. Support Vector Machine

Given that the data set is \( \{(x_i, y_i)\}_{i=1}^N \), where \( x_i \in \mathbb{R}^n \) represents the input vector of the training sample and \( y_i \in \mathbb{R} \) is the corresponding sample label. The separating hyperplane is formed as:

\[
f(x) = w^T \phi(x) + b
\]

where \( w \) is the weight vector and \( b \) is the bias. In order to find the \( w \) and \( b \) values, we must find the points closest to the separating hyperplane and maximize that margin. The optimization problem can be written as:

\[
\max_{w,b} \left\{ \min_{i} (y_i \cdot (w^T \phi(x_i) + b)) - \frac{1}{\|w\|} \right\}, i = 1, 2, \ldots, N
\]

The above optimization function is equivalent to the following one with inequality constraints [9].
\[
\begin{align*}
\min \| y - w^T \phi(x) + b \| + C \sum_i \xi_i \\
\text{s.t.} \quad y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, i = 1, 2, \ldots, N \\
\xi_i \geq 0, i = 1, 2, \ldots, N
\end{align*}
\]  
(3)

where \( C \) is a tradeoff parameter and \( \xi_i \) is a slack variable. By using Lagrange dual algorithm, the previous problem can be transformed into an unconstrained optimization problem as follows:

\[
\max \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \phi(x_i) \phi(x_j) \\
\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0
\]

(4)

where \( \phi \) is kernel function and \( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N) \) is the Lagrange multiplier vector. In this paper the RBF kernel is used.

\[
K(x, x_i) = \exp \left( -\frac{\| x - x_i \|^2}{2\sigma^2} \right)
\]

(5)

where \( \sigma \) is the kernel parameter. The sequential minimal optimization (SMO) algorithm is used to solve Equation (4). The predictive function can be written as

\[
g(x) = \sum_i \alpha_i y_i K(x, x_i) + b
\]

(6)

Solving the optimization problem of SVM model by using SMO algorithm is iterative process. However, only those support vectors play real role in training SVM classifier.

### 2.2. Least Squares Support Vector Machine

Different from SVM, Equation (2) is converted to the optimization problem with an equality constraint as follows [10]:

\[
\begin{align*}
\min \sum_i e_i^2 + C \sum_i \xi_i \\
\text{s.t.} \quad y_i = w^T \phi(x_i) + b + e_i, i = 1, 2, \ldots, N
\end{align*}
\]  
(7)

where \( C \) is a tradeoff parameter and \( e_i \in R \) is the sample error. The Lagrangian function with Lagrange multipliers \( \alpha_i \) is defined as follows:

\[
L(w, b, \alpha, \varepsilon) = \frac{1}{2} w^T w + \frac{C}{2} \sum_i \varepsilon_i^2 + \sum_i \alpha_i (y_i - w^T \phi(x_i) - b - \varepsilon_i)
\]

(8)

The optimal parameters can be calculated by

\[
\begin{align*}
\frac{\partial L}{\partial w} = 0 & \quad \Rightarrow \quad \sum_i \alpha_i \phi(x_i) = 0 \\
\frac{\partial L}{\partial b} = 0 & \quad \Rightarrow \quad \sum_i \alpha_i = 0 \\
\frac{\partial L}{\partial \varepsilon_i} = 0 & \quad \Rightarrow \quad \alpha_i = C e_i \\
\frac{\partial L}{\partial \alpha_i} = 0 & \quad \Rightarrow \quad w^T \phi(x_i) + b + \varepsilon_i - y_i = 0
\end{align*}
\]

(9)

Eliminating the variables \( w \) and \( e_i \), the following linear equations can be got:

\[
\begin{bmatrix}
0 \\
I
\end{bmatrix} \Omega + \frac{1}{C} \begin{bmatrix}
I \\
\Omega
\end{bmatrix} \begin{bmatrix}
b \\
\alpha
\end{bmatrix} = \begin{bmatrix}
0 \\
y
\end{bmatrix}
\]

(10)
where $\mathbf{t} = [t_1, \ldots, t_N]^T$, $\mathbf{a} = [a_1, \ldots, a_N]^T$, $\mathbf{y} = [y_1, \ldots, y_N]^T$, $\mathbf{I}$ is a unit matrix, $\Omega$ is a $N \times N$ matrix and $\Omega_{ij} = K(x_i, x_j), k, l = 1, 2, \cdots, N$. The predictive function of LSSVM model is same as Equation (6).

3. Proposed Approach : DL-SVM

As shown in Fig.1, LSSVM in the first layer of DL-SVM is utilized to estimate all sample in original dataset and the sparse dataset is sent to training SVM model in the second layer.

![Figure1. Scheme of DL-SVM](image)

3.1. Improved Weights Least Squares Support Vector Machine

If the training dataset is a balanced binary dataset, the estimation of the support values is optimal when the error variable distribution is Gaussian. WLSSVM (weights least squares support vector machine) use weighting factor $\beta_i$ to weight the error variable $e_i = \alpha_i / C$ [17]. The optimization problem is expressed as follows.

$$
\min_{w,b,a} J(w, e) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{N} \beta_i e_i^2
$$

(11)

The Lagrangian function with Lagrange multipliers becomes

$$
L(w, b, e, a) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{N} \beta_i e_i^2 + \sum_{i=1}^{N} a_i (y_i - w^T \phi(x_i) - b - e_i)
$$

(12)

The KKT system becomes

$$
\begin{bmatrix}
0 & \mathbf{I}^T \\
\mathbf{I} & \Omega + \frac{1}{C} \mathbf{\boldsymbol{\beta}}
\end{bmatrix}
\begin{bmatrix}
\mathbf{a} \\
\mathbf{y}
\end{bmatrix}
= \begin{bmatrix}
0 \\
\mathbf{y}
\end{bmatrix}
$$

(13)

where $\mathbf{\boldsymbol{\beta}}$ is a diagonal matrix.

$$
\mathbf{\boldsymbol{\beta}} = \text{diag} \{ \beta_1, \beta_2, \cdots, \beta_N \}
$$

(14)

The value of weighting factor $\beta_i$ can be obtained by taking

$$
\beta_i = \begin{cases}
1 & \text{if } |e_i / \delta| \leq c_1 \\
\frac{c_2 - e_i / \delta}{c_2 - c_1} & \text{if } c_1 \leq |e_i / \delta| \leq c_2 \\
10^{-4} & \text{otherwise}
\end{cases}
$$

(15)

where $\delta = \frac{\text{IQR}}{2 \times 0.6745}$ [18], IQR (interquartile range) is the difference between the 75th percentile and the 25th percentile of error variable distribution in LSSVM case. $c_1$ and $c_2$ are set to 2.5 and 3 respectively.

In WLSSVM model, the RBF kernel matrix is written as
\[
\Omega = \begin{bmatrix}
K(x_i, x_i) & \cdots & K(x_i, x_N) \\
\vdots & \ddots & \vdots \\
K(x_N, x_i) & \cdots & K(x_N, x_N)
\end{bmatrix}
\]

where \(K(x_i, x_j)\) is the kernel function of \(x_i\) and \(x_j\). If the distance between two different samples is large, the kernel matrix tends to a unit matrix. Under these circumstances, Equation (10) can be written as

\[
\begin{bmatrix}
0 \\
I
\end{bmatrix}
= \begin{bmatrix}
\frac{1}{C} \mathbf{x} \\
[\mathbf{w}^T, \mathbf{y}^T]
\end{bmatrix}
\]

The Lagrange multiplier \(\alpha_i\) can be solved by the follow equation

\[b + \frac{C + 1}{C} \alpha_i = y_i\]  \hspace{2cm} (18)

\(b\) is a constant and the value of label \(y_i\) is \(+1\) or \(-1\) in binary LSSVM classifier, so the value of \(\alpha_i\) is \(\frac{(1-b)C}{C+1}\) or \(\frac{-(1+b)C}{C+1}\).

In order to adapt to sample values that lie in different ranges, an improved RBF kernel matrix with the normalized \(L_2\) sample distance is proposed. \(i\)-th columns of \(\Omega\) represents the RBF kernel value sequence between \(i\)-th sample and all samples, which corresponds to a \(L_2\) sample distance sequence \([d_{i1}, d_{i2}, \ldots, d_{iN}] = [||x_i - x_1||, ||x_i - x_2||, \ldots, ||x_i - x_N||]\).

\[
\begin{align*}
\hat{d}_{i\text{max}} &= \max[d_{i1}, d_{i2}, \ldots, d_{iN}] \\
\hat{d}_{i\text{min}} &= \min[d_{i1}, d_{i2}, \ldots, d_{iN}] \\
\hat{R}_d &= \hat{d}_{i\text{max}} - \hat{d}_{i\text{min}}
\end{align*}
\]

where \(\hat{d}_{i\text{max}}, \hat{d}_{i\text{min}}\) and \(\hat{R}_d\) are the maximum element, minimum element and the range in \([d_{i1}, d_{i2}, \ldots, d_{iN}]\) respectively. The normalized \(L_2\) sample distance sequence is defined as follows:

\[
D_i = [d_{i1}, d_{i2}, \ldots, d_{iN}] / \hat{R}_d
\]

The normalized \(L_2\) sample distance matrix is shown as follows:

\[
D = [D_1', D_2', \ldots, D_N']
\]  \hspace{2cm} (21)

**Algorithm 1. IWLSSVM**

**Inputs:**

- The training dataset \(\{(x_i, y_i)\}_{i=1}^N\)
- The tradeoff parameter \(C\)
- The kernel parameter \(\sigma\)

**Outputs:**

- Lagrange multiplier sequence \([\alpha_1, \alpha_2, \ldots, \alpha_N]\)

**Process:**

1. Calculate the normalized \(L_2\) sample distance matrix \(D\) and the corresponding RBF kernel matrix \(\Omega\).
2. Solve the linear system of LSSVM and obtain the Lagrange multiplier sequence of LSSVM model.
3. Calculate the error variables of LSSVM model.
4. Calculate \(\hat{s}\) from the error variable distribution.
5. Determine the weighting factor \(\beta_i\) according to Equation (15).
6. Solve IWLSSVM model according to Equation (13), giving the Lagrange multiplier sequence \([\alpha_1, \alpha_2, \ldots, \alpha_N]\).
3.2. Normalized Lagrange Multiplier Shannon Entropy to Reduce Training Dataset

As a concept of information theory, entropy is employed to measure information content [11]. The entropy of i-th Lagrange multiplier $\alpha_i$ is defined as

$$E_i = -\left(\frac{\text{abs}(\alpha_i)}{\sum_{i=1}^{N} \text{abs}(\alpha_i)} \right) \log_2 \left(\frac{\text{abs}(\alpha_i)}{\sum_{i=1}^{N} \text{abs}(\alpha_i)} \right)$$

(22)

where $\text{abs}(\cdot)$ represents the absolute value of a number. The entropy sequence of Lagrange multipliers $[E_1, E_2, \cdots, E_N]$ is normalized as $[\text{norm}E_1, \text{norm}E_2, \cdots, \text{norm}E_N]$.

$$\text{norm}E_i = \frac{E_i}{E_{\text{max}} - E_{\text{min}}}$$

(23)

where $E_{\text{max}}$ and $E_{\text{min}}$ are the maximum element and the minimum element in $[E_1, E_2, \cdots, E_N]$. A judgment condition is defined as

$$\text{norm}E_i \geq \theta, \theta \in [0, 1]$$

(24)

where $\theta$ is a judgment parameter. If the judgment condition is satisfied, the corresponding training sample will be added in the sparse training set of SVM, otherwise it will be eliminated. By adapting this method, the solution of SVM in the second layer is sparse.

**Algorithm 2. DL-SVM**

**Inputs:**
- The training dataset $\{(x_i, y_i)\}_{i=1}^{N}$
- The testing sample $x^*$
- The tradeoff parameter $C$
- The kernel parameter $\sigma$
- The judgment parameter $\theta \in [0, 1]$

**Outputs:**
- The predicted class label for $x^*$

**Process:**
1. Calculate the Lagrange multiplier sequence $[\alpha_1, \alpha_2, \cdots, \alpha_N]$ of IWLSSVM model.
2. while $i < N + 1$ do
3. Calculate the normalized Lagrange multiplier Shannon entropy $\text{norm}E_i$ of $\alpha_i$
4. if $\text{norm}E_i \geq \theta$ then
5. The corresponding training sample is sent to a sparse set.
6. end if
7. end while
8. Train SVM model with the sparse dataset and obtain support vectors (SVs)
9. return the predicted class label $y^*$ according to Equation (6)

4. Experimental Results

In this section, the effectiveness of the proposed DL-SVM model is experimentally evaluated compared to several related methods. All results are obtained by using Dell laptop (2.7GHz CPU and 8GB memory) and all models are implemented by using Python 3.6.

4.1. Experimental Setup

In the experiments, seven binary benchmark datasets or binary subsets of benchmark datasets from the UCI repositories [19] are used to display the experimental results. Wine-1, Wine-2, Iris-1, Iris-2, Ionosphere, Abalone-1 and Abalone-2. Wine-1 is a binary subset of Wine dataset, samples of class 1
and class 2 are used and the labels of class 1 and class 2 are set to $+1$ and $-1$ respectively. The first 80 samples are selected as training set and the rest 40 as testing set. Wine-2 is a subset of Wine, which contains samples of class 2 and class 3 in Wine dataset. The first 81 samples are used for training and the rest 40 for testing. Iris-1 and Iris-2 are subsets of Iris dataset. Iris-1 contains samples of class ‘setosa’ and class ‘versicolor’, we use the first 35 ‘setosa’ samples and the first 35 ‘versicolor’ samples as training instances and the rest for testing. Iris-2 contains samples of class ‘versicolor’ and class ‘virginica’, the dataset is divided in the same way as Iris-1. For Ionosphere, we use the first 301 samples for training and the rest 50 for testing. Abalone-1 contains samples of class 6 and class 7 in Abalone dataset, the first 500 samples are used for training and the rest 150 for testing. Abalone-2 contains samples of class 14 and class 15 in Abalone dataset, the first 179 samples are used for training and the rest 50 for testing.

DL-SVM and IWLSSVM are compared to three baseline methods: the standard SVM, LSSVM and LSSVM with normalized L$_2$ sample distance. Every experimental process is repeated 10 times, the average result are recorded. There three parameters need to set. $\theta$ is set to 0.3, 0.5, 0.7, respectively. $C$ and $\sigma$ are optimized by cross validation method and the ranges of $C$ and $\sigma$ are set to

$$\{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}.$$

4.2. Error Variable Distribution of IWLSSVM

We first compare the error Variable distribution of IWLSSVM to LSSVM and LSSVM with normalized L$_2$ sample distance in Wine-1 dataset and Ionosphere dataset, with results shown in Fig.2 and Fig.3. In LSSVM model, the RBF kernel matrix is written as Equation (16), if the distance between two different samples is large, the kernel matrix tends to a unit matrix and the error variable distributes over two constants. LSSVM model with normalized L$_2$ sample distance can solve the above problem to some extent, but the distribution of error variable is discrete. The error variable distribution in IWLSSVM model approximate Gaussian and the support vectors can be evaluated from the Lagrange multiplier sequence. The bigger the absolute value of Lagrange multiplier, the greater the probability that the sample is a support vector.

**Figure 2.** Histogram of Lagrange multipliers for Wine-1 dataset. (a) Pure LSSVM model (b) LSSVM model with normalized L2 sample distance (c) IWLSSVM

**Figure 3.** Histogram of Lagrange multipliers for Ionosphere dataset. (a) Pure LSSVM model (b) LSSVM model with normalized L2 sample distance (c) IWLSSVM
4.3. Performance of DL-LSSVM

The testing and training performance of pure SVM and DLSVM with different $\theta$ are compared in this section. The average training time and the correct recognition rate in test set are presented in Table 1. For the first datasets, we can see that DLSVM with $\theta = 0.3$ can achieve 100% correct recognition rate in test phase and the average training time is smaller that of pure SVM. For Iris-1 dataset, the average training time of DLSVM with $\theta = 0.5$ is only 0.267s and the correct recognition rate reaches 100%. For Ionosphere dataset, DLSVM with $\theta = 0.3$ has smaller average training time than SVM, the correct recognition rate is 100%. For the last two datasets, DLSVM has larger correct recognition rate compare to the pure SVM when $\theta = 0.5$. We can know that DLSVM has similar or even better performance with lower computation cost, due to DLSVM can provide a sparse training dataset for SVM in the second layer. By adjusting the value of $\theta$, we can get different DLSVM model to adapt to different application scenarios.

Table 1. Performance comparison between DLSVM and SVM

|                  | DL-SVM($\theta=0.1$) | DL-SVM($\theta=0.3$) | DL-SVM($\theta=0.5$) | SVM |
|------------------|-----------------------|-----------------------|-----------------------|-----|
|                  | Average training time(s) | Correct Recognition rates in test | Average training time(s) | Correct Recognition rates in test | Average training time(s) | Correct Recognition rates in test |
| Wine-1           | 0.583                 | 100%                  | 0.303                 | 100%                  | 0.262                 | 90%                  | 0.712                 | 95%                  |
| Wine-2           | 0.726                 | 100%                  | 0.712                 | 100%                  | 0.436                 | 95%                  | 0.574                 | 100%                  |
| Iris-1           | 0.468                 | 100%                  | 0.321                 | 100%                  | 0.267                 | 100%                  | 0.395                 | 100%                  |
| Iris-2           | 0.430                 | 100%                  | 0.336                 | 100%                  | 0.213                 | 100%                  | 0.395                 | 100%                  |
| Ionosphere       | 7.971                 | 100%                  | 6.152                 | 100%                  | 3.114                 | 98%                  | 6.241                 | 98%                  |
| Abalone-1        | 30.281                | 64%                   | 27.258                | 57.4%                 | 17.912                | 50%                  | 21.187                | 46%                  |
| Abalone-2        | 4.058                 | 54%                   | 3.126                 | 56%                   | 2.786                 | 60%                  | 2.945                 | 54%                  |

Table 2. Size of sparse training set and support vectors in DL-SVM

|                  | $\theta=0.1$ | $\theta=0.3$ | $\theta=0.5$ |
|------------------|------------|------------|------------|
|                  | sparse training set | support vectors | sparse training set | support vectors | sparse training set | support vectors |
| Wine-1(80)       | 50         | 25         | 15         | 12         |
| Wine-2(81)       | 68         | 42         | 24         | 24         |
| Iris-1(70)       | 64         | 51         | 59         | 56         |
| Iris-2(70)       | 67         | 67         | 64         | 62         |
| Ionosphere(301)  | 288        | 268        | 255        | 200        |
| Abalone-1(1500)  | 460        | 142        | 109        | 99         |
| Abalone-2(179)   | 172        | 123        | 82         | 73         |

Figure 4. Training time and training error of DLSVM with different values of $\theta$ in Wine_1 dataset

The size of sparse training set and support vectors in the second layer of DL-SVM with different values of $\theta$ are presented in Table 2. We can see the size of sparse training set and support vectors decreases as the value of $\theta$ increases. Those less important support vectors are deleted. The training time of DL-SVM is equal to the sum of the training time of IWLSSVM and the training time of SVM with the sparse training set, so the training time decreases as the value of $\theta$ increases.
5. Conclusions
In this paper, a novel SVM model named DL-SVM is proposed. DL-SVM is a double layer structure. IWLSSVM is used to estimate all samples in the first layer and a sparse dataset is utilized by SVM in the second layer. In IWLSSVM, a normalized $L_2$ sample distance matrix is used to calculate Lagrange multiplier sequence and the Lagrange multiplier distribution becomes approximate Gaussian. The bigger the absolute value of Lagrange multiplier, the greater the probability that the sample is a support vector. A judgment parameter $\theta$ is used to select sparse dataset by comparing with entropy sequence of Lagrange multipliers. Experiments on different datasets further testified the effectiveness of DL-SVM.

6. Acknowledgements
This work was supported by the National Natural Science Foundation of China under Grant No. 51577046, the State Key Program of National Natural Science Foundation of China under Grant No. 51637004, the national key research and development plan "important scientific instruments and equipment development" Grant No.2016YFF0102200, Equipment research project in advance Grant No.41402040301.

7. References
[1] Mao, X.; Wang, L.; and Li, C. 2009. SVM Classifier for Analog Fault Diagnosis Using Fractal Features. In International Symposium on Intelligent Information Technology Application. IEEE, 553-557.
[2] Gao, X., and Hou, J. 2016. An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process. Neurocomputing 174:906-911.
[3] Yakhnenko, O.; Verbeek, J.; and Schmid, C. 2011. Region-Based Image Classification with a Latent SVM Model. Machine Learning.
[4] Liu, C.; Wu, X.; and Jia, Y. 2016. Transfer Latent SVM for Joint Recognition and Localization of Actions in Videos. IEEE Transactions on Cybernetics 46(11):2596-2608.
[5] Zhang, X.; Wang, T.; and Chen, H. 2008. Story Link Detection Based on Event Model with Uneven SVM. Asia Information Retrieval Conference on Information Retrieval Technology. Springer-Verlag 436-441.
[6] Kuo, B. C.; Ho, H. H.; Li, C. H.; Hung, C. C.; and Taur, J. S. 2013. A Kernel-Based Feature Selection Method for SVM With RBF Kernel for Hyperspectral Image Classification. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing 7(1):317-326.
[7] Prasad, Y.; Biswas, K. K.; and Singla, P. 2013. Scaling-up quadratic programming feature selection. In AAAI Conference on Late-Breaking Developments in the Field of Artificial Intelligence, 95-97. AAAI Press.
[8] Dong, J. X.; Krzyzak, A.; and Suen, C. Y. 2005. Fast SVM training algorithm with decomposition on very large data sets. IEEE Transactions on Pattern Analysis & Machine Intelligence, 27(4):603-618.
[9] Hsieh, C. J.; Chang, K. W.; Lin, C. J.; et al. 2008. A dual coordinate descent method for large-scale linear SVM. Proceeding International Conference on Machine Learning, 408-415. ACM.
[10] Suykens, J. A. K., and Vandewalle, J. 1999. Least Squares Support Vector Machine Classifiers. Neural Processing Letters, 9(3):293-300.
[11] Wu, C.; Wang, X.; Bai, D.; et al. 2009. Fast Incremental Learning Algorithm of SVM on KKT Conditions. In sixth International Conference on Fuzzy Systems and Knowledge Discovery, 551-554. IEEE Computer Society.
[12] Si, G. Q.; Zhang, G.; and Shi, J. Q. 2016. A robust pruning method for reduced weighted least squares support vector machine. Control and Decision Conference: 3576-3581. IEEE.
[13] Suykens, J. A. K.; Brabanter, J. D.; Lukas, L.; et al. 2008. Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing, 48(1):85-105.
[14] Torr, P. H. S., and Ladicky, L. Locally linear support vector machines. 2011. International Conference on Machine Learning, 985-992.

[15] Lee, Y. J., and Huang, S. Y. 2007. Reduced Support Vector Machines: A Statistical Theory. IEEE Transactions on Neural Networks, 18(1):1-13.

[16] Nguyen, G. H.; Phung, S. L.; and Bouzerdoum, A. 2010. Efficient SVM training with reduced weighted samples. IEEE World Congress on Computational Intelligence, WCCI.

[17] Suykens, J. A. K.; Brabanter, J. D.; and Lukas, L.; et al. 2008. Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing, 48(1):85-105.

[18] Deng, W.; Zheng, Q.; and Chen, L. 2009. Regularized Extreme Learning Machine. IEEE Symposium on Computational Intelligence & Data Mining, 389-395.

[19] A. Frank and A. Asuncion, UCI Machine Learning Repository Sch. Inf. Comput. Sci., Univ. California, Irvine, 2010 [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html