SVM Parameters Tuning with Quantum Particles Swarm Optimization

Zhiyong Luo[1], Wenfeng Zhang[2], Yuxia Li[3] and Min Xiang[1]
1. School of Automation, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
2. School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China
3. School of Automation Engineering, University of Electric Science and Technology of China, Chengdu, 610054, China
luozhyong@163.com

***Abstract*—Common used parameters selection method for support vector machines (SVM) is cross-validation, which is complicated calculation and takes a very long time. In this paper, a novel regularization parameter and kernel parameter tuning approach of SVM is presented based on quantum particle swarm optimization algorithm (QPSO). QPSO is a particle swarm optimization (PSO) with quantum individual that has better global search capacity. The parameters of least squares support vector machines (LS-SVM) can be adjusted using QPSO. Classification and function estimation are studied using LS-SVM with wavelet kernel and Gaussian kernel. The simulation results show that the proposed approach can effectively tune the parameters of LS-SVM, and improved LS-SVM with wavelet kernel can provide better precision.**

***Keywords*—quantum particle swarm optimization algorithm (QPSO), parameters tuning, support vector machines (SVM), least squares support vector machines (LS-SVM)**

I. INTRODUCTION

As a machine learning method, support vector machines (SVM) originally introduced by Vapnik [1] within the area of the statistical theory and structural risk minimization has emerged as one powerful tool for data analysis. It has been widely used for many applications, such as regression and pattern recognition [3], [4].

It is well known that SVM generalization performance depends on a good setting of regularization parameter and the kernel parameter. To minimize the generalization error, these parameters should be properly selected. There are roughly several methods for SVM to select parameter: cross-validation, Vapnik-Chervonenkis (VC) bound and the Bayesian evidence framework method [5], [6], [7]. Cross-validation is the most commonly used approach due to its high accuracy, but it is complex calculation and consumes more computation time.

Particle swarm optimization algorithm (PSO) [8] [9], put forward by Kennery and Eberhart, is a swarm intelligent optimization method. PSO begins with a random population and searches for optima by updating the population. The quantum particle swarm optimization algorithm (QPSO) was presented in [10] [11] as an algorithm with good performance-complexity trade-off. Based on the quantum bit, the best chromosome's guidance is used to draw close to the optimum step by step, QPSO can find solution quickly in global space. In this paper, QPSO is applied to tune parameters of least squares support vector machines (LS-SVM) [12], which are used in pattern recognition and function estimator. Simulations conducted on benchmark data set and standard approximated functions demonstrate the effectiveness and efficiency of the proposed method. The better performances and easier of implementing parameters selection with simple calculation is compared with cross-validation method to tune LS-SVM with wavelet kernel and Gaussian kernel.

The rest of this paper is organized as follows: In Section II, we review LS-SVM. The proposed approach is given in Section III. Experimental results are shown in Section IV and some conclusions will be given in Section V.

II. LEAST SQUARES SUPPORT VECTOR MACHINES

Let $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_l,y_l)\}$ be a training set with input data $x_k \in R^d$ and corresponding output data $y_k \in R$. Learning from the training data can be viewed as a multivariate function f approximation that represents the relation between the input data and output data [1], [2]. In the general case, the input data are mapped into a feature space by nonlinear function $\Phi(x)$, the SVM function $f(x)$ can be expressed as:

$$f(x)=\boldsymbol{\omega}^{\mathrm{T}}\Phi(x)+b \tag{1}$$

where $\boldsymbol{\omega}^T$ is a m-dimensional vector, and b is a scalar.

A. LS-SVM for function estimation [10]

In LS-SVM for function estimation, one defines the following optimization problem:

$$\min_{\boldsymbol{\omega},b,\mathbf{e}} J(\boldsymbol{\omega},b,\mathbf{e})=\frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}+\gamma\frac{1}{2}\sum_{k=1}^{l}e_k^2 \tag{2}$$

s.t.

$$y_k=\boldsymbol{\omega}^{\mathrm{T}}\Phi(x_k)+b+e_k,\ k=1,\cdots,l \tag{3}$$

where $\mathbf{e}=[e_1,e_2,\cdots,e_l]^{\mathrm{T}}$, $e_k \in R^{1\times 1}$ denotes the error vector, γ is regularization parameter.

Using the optimization theory can solve this problem. One can define the Lagrangian for this problem as follows

$$L(\boldsymbol{\omega}, b, \mathbf{e}; \boldsymbol{\alpha}) = J(\boldsymbol{\omega}, b, \mathbf{e}) - \sum_{k=1}^{l} \alpha_k \left\{ \boldsymbol{\omega}^{\mathrm{T}} \Phi(x_k) + b + e_k - y_k \right\} \tag{4}$$

where α_k $(k = 1, \cdots, l)$ are Lagrange multipliers.

The conditions for optimality lead to a set of linear equations:

$$\begin{bmatrix} \mathbf{0} & \mathbf{1}^{\mathrm{T}} \\ \mathbf{1} & \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \tag{5}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_l]^{\mathrm{T}}$, $\mathbf{1} = [1, 1, \cdots, 1]^{\mathrm{T}}_{1\times l}$, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_l]^{\mathrm{T}}$ and $\boldsymbol{\Omega}_{mn} = \boldsymbol{\Phi}(x_m)^{\mathrm{T}} \boldsymbol{\Phi}(x_n)$ for $m, n = 1, \cdots, l$. According to Mercer's condition, there exists kernel function $K(x_m, x_n) = \boldsymbol{\Phi}(x_m)^{\mathrm{T}} \boldsymbol{\Phi}(x_n)$.

The resulting LS-SVM model for function estimation becomes:

$$f(x) = \sum_{k=1}^{l} \alpha_k K(x, x_k) + b \tag{6}$$

where α_k, b are the solution to the linear system (5).

B. LS-SVM for classification [10]

Considering $y_k(\boldsymbol{\omega}^{\mathrm{T}} \Phi(x_k) + b) = 1 - e_k$, $y_k \in \{+1, -1\}$ for classification, similar to function estimation, the solutions lead to a set of linear equations:

$$\begin{bmatrix} \mathbf{0} & \mathbf{y}^{\mathrm{T}} \\ \mathbf{y} & \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \tag{7}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_l]^{\mathrm{T}}$, $\mathbf{1} = [1, 1, \cdots, 1]^{\mathrm{T}}_{1\times l}$, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_l]^{\mathrm{T}}$ and $\boldsymbol{\Omega}_{mn} = y_m y_n \boldsymbol{\Phi}(x_m)^{\mathrm{T}} \boldsymbol{\Phi}(x_n)$.

The resulting LS-SVM model for classification is:

$$f(x) = \operatorname{sgn}\left(\sum_{k=1}^{l} \alpha_k y_k K(x, x_k) + b\right) \tag{8}$$

where α_k, b are the solution to the linear system (7).

III. Quantum Particles Swarm Optimization Tuning

Parameters selection for SVM is very complex and quite hard to solve by conventional optimization techniques. Here quantum particles swarm optimization algorithm (QPSO) is adopted to tune the parameters of LS-SVM model. The QPSO was presented in [10] [11] as an algorithm with good performance-complexity trade-off. Based on the quantum bit, the quantum bit has the advantage that it can represent a linear superposition of states in search space probabilistically, and the best chromosome's guidance is used to draw close to the optimum step by step, QPSO can find solution quickly in global space.

In the quantum theory, the minimum unit that carries information is a qubit, which can be in any superposition of state 0 and 1. We define such a quantum particle vector $Q(t) = q_1^t, \cdots, q_n^t$ at generation t, where n is the size of the population, and q_j^t is the j particle with quantum energy which is defined as:

$$q_j^t = \left[q_j^1(1), q_j^1(2), \cdots, q_j^1(m) \right] \tag{9}$$

where $1 \le q_j^t(i) \le (i = 1, \cdots, m)$, which represent a bit of quantum particle, m is the particle's length. $q_j^t(i)$ gives the appearing probabilities of the state "0". Then the QPSO algorithm can be described as:

```
begin
    t ← 0
    initialize Q(t)
    observe Q(t) to get P(t)
    evaluate P(t)
    store the best solution among P(t)
    while (not termination-condition) do
    begin
        t ← t + 1
        observe Q(t − 1) to get P(t)
        evaluate P(t)
        update Q(t)
        store the best solution among P(t)
    end
end
```

where $P(t)$ are are particle vector, $P(t) = [p_1^t, \cdots, p_n^t]$, p_j^t is the j th particle.

In the "initialize $Q(t)$", all $q_j^t(i)$ $(i = 1, \cdots, m)$ of quantum chromosome q_j^t $(j = 1, \cdots, n)$ are initialized as $1/\sqrt{2}$, which means that all the possible linear superposition of states appear in the same probability. Particle vector $P(t)$ is formed by observing $Q(t)$ in the next step, the procedure can be described as:

$$p_j^t(i) = \begin{cases} 1 & q_j^t < \text{rand}[0, 1] \\ 0 & q_j^t > \text{rand}[0, 1] \end{cases} \tag{10}$$

where $p_j^t(i)$ represents the i th bit of p_j^t $(j = 1, \cdots, n)$. Then "evaluate $P(t)$" step is performed with evaluation function, and the best solution among initial $P(t)$ is stored.

In "while $(\cdots)$ do" cycles, Particle vector $P(t)$ is formed by observing $Q(t-1)$ and "evaluate $P(t)$ " step is implemented with above-mentioned method. A "update $Q(t)$" step is added, like the PSO the QPSO algorithm has memory in order to store the best position values already found for each particle (p^t_{jbest}) and the best global position (p^t_{gbest}). From these positions, the best global and individual quantum energy values are calculated in order to generate changes in the particle positions.

$$q^t_{gbest} = \alpha \times p^t_{gbest} + \beta \times (1 - p^t_{gbest}) \tag{11}$$

$$q^t_{jbest} = \alpha \times p^t_{jbest} + \beta \times (1 - p^t_{jbest}) \tag{12}$$

where $\alpha + \beta = 1$, $0 < \alpha, \beta < 1$ are called the control parameters which represent the control degree of $Q(t)$. The smaller of α , the bigger of the appear probability of the desired item

$$q^{t+1}_j = c_1 \times q^t_j + c_2 \times q^t_{jbest} + c_3 \times q^t_{gbest} \tag{13}$$

where $c_1 + c_2 + c_3 = 1$, $0 < c_1, c_2, c_3 < 1$ represent the degree of the belief on oneself, local maximum and global maximum, respectively.

After "update $Q(t)$" step in the cycles, the best particle solution among $P(t)$ is selected, if it is better than the stored best particle solution, then it will be stored. When the cycles are completed, the stored best particle solution is the expected particle solution with quantum particles swarm optimization.

In order to speed up quantum particles swarm optimization, we adopt two-step scheme to apply QPSO on the parameters selection. Because the range of possible parameters is wide, reducing the search range can improve efficiency of optimization, so the first step is applied to rough adjust the parameters, and the second step to tune the parameters within the selected range which is confirmed by the first step's obtained parameters, that is shown as:

$$v^1_r = \lambda^*_r - \eta(u^0_r - v^0_r) \tag{14}$$

$$u^1_r = \lambda^*_r + \eta(u^0_r - v^0_r) \tag{15}$$

where $v^0_r \ (r = 1, 2)$ are original minimum of regularization parameter γ range and kernel parameter σ range, and $u^0_r \ (r = 1, 2)$ are original maximum of parameters range. $\lambda^*_r \ (r = 1, 2)$ represent the first step optimized value of γ and σ, v^1_r and u^1_r are the second step minimum and maximum of parameters range, respectively. The coefficient $\eta \in [0, 0.5]$ is used to adjust the second step parameters search range according to the optimized result of the first step. For preventing the new range to get across original range, we make the following rules: if $v^1_r < v^0_r$, then $v^1_r = v^0_r$, and if $u^1_r > u^0_r$, then $u^1_r = u^0_r$.

To apply QPSO on the parameters tuning, we utilize the value range of the regularization parameter and the kernel parameter to decide the bit number of a particle solution p^t_j, then the corresponding bit number of a quantum chromosome q^t_j can be confirmed. For Gaussian kernel of LS-SVM model, the initial value range of the regularization parameter γ and the kernel parameter σ are $[0.1, 1000]$ and $[0.1, 100]$, respectively.

The evaluation function, which defines the selection criteria, affects the performance of the model parameters selection. Here, QPSO shows its flexibility to implement various criteria according to applications. One choice of the evaluation is to use the overall classification rate on a test set. The evaluation function of classification is defined as:

$$Evaluation_s = \ Accuracy_s \tag{16}$$

where $Accuracy_s$ is the classifier's s th generation testing accuracy.

Another choice of the evaluation is to use the function approximation precision on a test set. The evaluation function of function estimation is defined as:

$$Evaluation_t = \sqrt{\frac{\sum_{k=1}^{z} (y_k - f_k)^2}{z}} \tag{17}$$

where y_k and f_k denote the desired output and the approximation output for the test set's k th input data, respectively, z is the input data number of the test set, and $Evaluation_t$ is t th generation evaluation to present function approximation accuracy . According to LS-SVM model, the evaluation function is chosen to optimize the parameters by using QPSO.

In general quantum particle swarm optimization, the size of population is 20. 20 generations optimization is implemented in the first step, the program is terminated when the quantum particles swarm optimization has been completed. In second step, the program is terminated when the best evaluation has not changed more than a very small value, i.e. 10^{-4} for classification and function approximation over the last generation. When quantum particle swarm optimization is accomplished, the optimized best particle solution is converted to real value, the best value of regularization parameter γ and kernel parameter σ can be obtained.

IV. EXPERIMENTS

In this section, we evaluate the proposed SVM parameters tuning method with three numerical experiments, the classification of two-class benchmark problem, approximation of a single-variable function and two-variable function.

For comparison, the cross-validation (CV) and QPSO method are performed to tune the parameters of LS-SVM model with wavelet kernel and Gaussian kernel, respectively. The kernel function of wavelet kernel and Gaussian kernel are

$K(x,x_k)=\prod_{i=1}^{d}(1-\|x^i-x_k^i\|^2/a^2)\exp(-\|x^i-x_k^i\|^2/2a^2)$ and $K(x,x_k)=\exp(-\|x-x_k\|^2/(2\sigma^2))$, where the x_k^i denotes the i th component of the k th training samples, a is wavelet dilation coefficient [13].

A. Classification on benchmark problem

For classification, we use the UCI binary classification benchmark repository [14]: the Johns Hopkins university ionosphere (ion) and the sonar (snr) with input dimension n equal to 33 and 60, and total number of patterns 351 and 208, respectively. Each component of the input data is normalized to zero mean and unit standard deviation. We extract randomly 2/3 of the data as train set, and the rest as test set. The parameters of LS-SVM classifiers with wavelet kernel and Gaussian kernel are (γ,a) and (γ,σ), which are adjusted by the cross-validation (CV) and QPSO optimization, respectively. The parameters of PSO are $c_1=c_2=0.2$, $c_3=0.6$, $\alpha=0.1$ and $\beta=0.9$. The tuned parameters and results of classification are shown in Table I.

TABLE I. TUNED PARAMETERS AND RESULTS OF CLASSIFICATION

Method	Data	Kernel	γ	a or σ	Train (%)	Test (%)
CV	ion	Wavelet	3.16	4.25	100.00	96.8
		Gaussian	4.27	2.33	100.00	96.00
	snr	Wavelet	8.96	16.55	87.12	78.27
		Gaussian	7.24	23.33	86.10	77.15
QPSO	ion	Wavelet	3.96	0.25	100.00	97.17
		Gaussian	4.88	4.68	100.00	96.45
	snr	Wavelet	29.25	19.54	87.86	78.69
		Gaussian	27.76	24.56	86.55	77.68

B. Approximation of single-variable function

In this experiment, we approximate the following single-variable function[13]

$$f(x)=\begin{cases}-2.186x-12.864 & -10\le x<-2\\ 4.246x & -2\le x<0\\ 10e^{-0.05x-0.5} & \\ \quad\cdot\sin[(0.03x+0.7)x] & 0\le x\le 10\end{cases}$$

We have uniformly sampled examples of 200 points, 100 points of which are taken as training examples and others testing examples. The parameters of LS-SVM estimator with wavelet kernel and Gaussian kernel are adjusted with above two tuning parameters methods. The parameters of QPSO are $c_1=c_2=0.1$, $c_3=0.8$, $\alpha=0.1$ and $\beta=0.9$. The normalized root of mean-square-error (NRMSE) is served as one criteria for assessing the extrapolation ability of our procedure. Table II lists the tuned parameters and approximation errors. The approximation results are plotted in Figs. 1 and 2, respectively.

TABLE II. TUNED PARAMETERS AND RESULTS OF APPROXIMATION

Method	Kernel	γ	a or σ	NRMSE (Train)	NRMSE (Test)
CV	Wavelet	910.86	0.14	0.0022	0.0034
	Gaussian	900.68	0.47	0.0203	0.0212
QPSO	Wavelet	992.86	0.22	0.0019	0.0027
	Gaussian	865.68	0.36	0.0086	0.0098

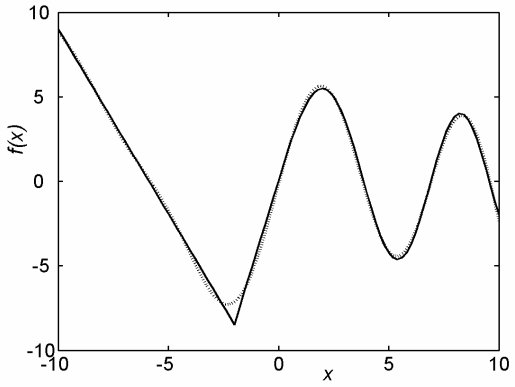

Figure 1. Original function (solid line) and resulting approximation by LS-SVM with Gaussian kernel (dotted line)

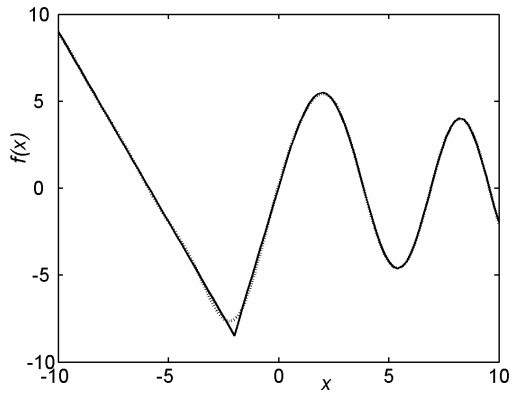

Figure 2. Original function (solid line) and resulting approximation by LS-SVM with Wavelet kernel (dotted line)

C. Approximation of two-variable function

This experiment is to approximate a two-variable function[13]

$$f(x,y)=(x^2-y^2)\sin(0.5x)$$

We take uniformly 81 sampled points as the training examples, and 1600 points as the testing examples. The

parameters of QPSO are $c_1 = c_2 = 0.1$, $c_3 = 0.8$, $\alpha = 0.05$ and $\beta = 0.95$. Table III lists the tuned parameters and approximation results of LS-SVM with wavelet kernel and Gaussian kernel, respectively. Fig. 3 shows the original function $f(x, y)$, and Figs. 4 and 5 show the approximation results.

TABLE III. TUNED PARAMETERS AND RESULTS OF APPROXIMATION

Method	Kernel	γ	a or σ	NRMSE	
				(Train)	(Test)
CV	Wavelet	268.53	0.88	0.0169	0.0352
	Gaussian	566.71	0.54	0.0188	0.0432
QPSO	Wavelet	962.65	0.56	0.0122	0.0248
	Gaussian	978.26	0.69	0.0144	0.0289

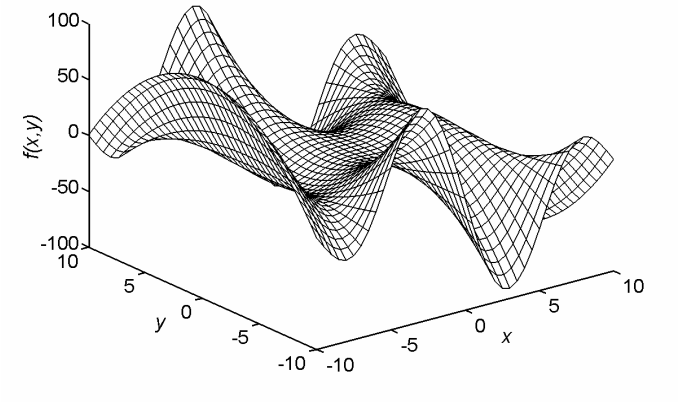

Figure 3. Original function

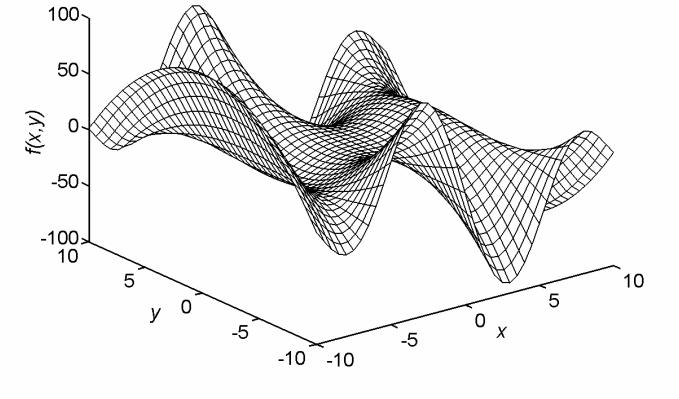

Figure 4. Resulting approximation by LS-SVM with Gaussian kernel

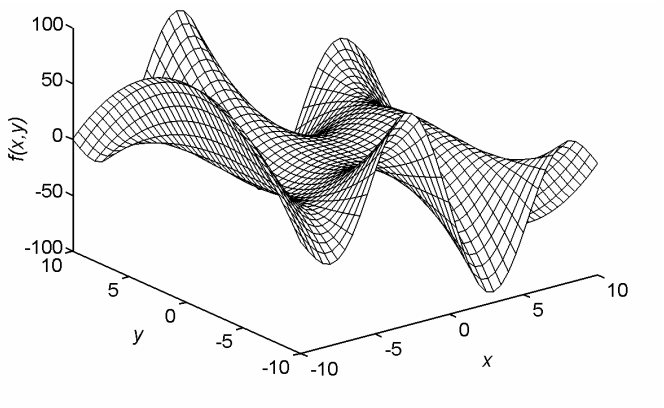

Fig. 5. Resulting approximation by LS-SVM with Wavelet kernel

We have compared the classification and function approximation results obtained by wavelet kernel and Gaussian kernel, whose parameters are tuned with the cross-validation and QPSO, respectively. To summarize, the QPSO method is better than the cross-validation to adjust the parameters of LS-SVM models in these three experiments, and the wavelet kernel has better performance than Gaussian kernel.

V. CONCLUSION

In this paper, we discuss a practical way to tune the regularization parameter and the kernel parameter with quantum particle swarm optimization (QPSO), which takes full advantage of particle swarm optimization (PSO) and update with quantum individual. This work provides a new adjusting parameters of SVM approach. Three simulations of LS-SVM model with wavelet kernel and Gaussian kernel show that the proposed method is effective and efficient, and enhanced LS-SVM with wavelet kernel shows good generalization ability on classification problem and gives better approximation on function estimation. The main attraction of the proposed tuning method is that it is simple calculation of implementation with better performances in comparison with the cross-validate method.

ACKNOWLEDGMENT

The first author would like to thank Prof. Ping Wang and Dr. Min Xiang of Chongqing University of Posts and Telecommunications of China for their help. This work was Supported by National High-Tech R&D Program of China (No. 2006AA040302-1) and Scientific Research Foundation for PhD of Chongqing University of Posts and Telecommunications (No. A2007-15).

REFERENCES

[1] V.N. Vapnik, The Nature of Statistical Learning Theory (the second edition), New York: Springer-Verlag, 1998.

[2] T. Poggio, R. Rifkin, S. Mukherjee, P. Niyogi, "General Conditions for Predictivity in Learning Theory," Nature, vol. 428, pp. 419–422, Mar.2004.

[3] M. Carozza, S. Rampone, "Towards an incremental SVM for regression," Proc. International Joint Conference on Neural Networks, Como, Italy, Jul. 2000, pp. 405–410.

[4] Z. Y. Luo, Z. k. Shi, "Fault diagnosis of analog circuits using SVM with Bayesian Framework," Journal of System Simulation, vol. 19, pp. 3009–3013, July. 2007.

[5] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," Machine Learning, vol. 46, pp. 131–159, Jan. 2002.

[6] O. Chapelle, V. Vapnik, " Model selection for support vector machines," in Handbook of Advances in Neural information Processing Systems, Cambridge: Mass MIT Press, 2000.

[7] J. T. Kowk, "The evidence framework applied to support vector machines," IEEE Transaction on Neural Network, vol. 11, pp. 1162–1173, Sep./Oct. 2000.

[8] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization," Proc. IEEE International Conference on neural nerworks, Nagoya, Japan, 1995, pp. 1942–1948.

[9] R. C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," Proc. of the sixth international symposium on micro machine and human science, Nagoya, Japan, 1995, pp. 39–43.

[10] S. Y. Yang, M. Wang, L. C. Jiao, "A Quantum Particle Swarm Optimization," Proc. IEEE Congress on Evolutionary Computation, CEC'04, Portland, Oregon, USA, June 2004, pp. 320–324.

[11] L. D. Oliveira, F. Ciriaco, T. Abr˜ao, P. J. E. Jeszensky, "Particle Swarm and Quantum Particle Swarm Optimization Applied to DS/CDMA Multiuser Detection in Flat Rayleigh Channels," Proc. IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications, Manaus, Amazon, Brazil, Aug. 2006, p133–137.

[12] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, Least squares support vector machines, Singapore: World Scientific press, 2002.

[13] L. Zhang, W. D. Zhou, L. C. Jiao, "Wavelet Support Vector Machine," IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 34, pp. 34–39, Jan./Feb. 2004.

[14] C. L. Blake, C. J. Merz, UCI Repository of machine learning databases, Irvine, CA: University of California, Dept. of Information and Computer Science, 1998.