Adaptive and Iterative Least Squares Support Vector Regression Based on Quadratic Renyi Entropy*

Jingqing Jiang [1], Chuyi Song [1], Haiyan Zhao [1], Chunguo Wu [2,3] and Yanchun Liang [2]**

[1] *College of Mathematics and Computer Science, Inner Mongolia University for Nationalities, Tongliao Inner Mongolia 028043, China*

[2] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*

[3] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China*

jiangjingqing@yahoo.com.cn

Abstract

An adaptive and iterative LSSVR algorithm based on quadratic Renyi entropy is presented in this paper. LS-SVM loses the sparseness of support vector which is one of the important advantages of conventional SVM. The proposed algorithm overcomes this drawback. The quadratic Renyi entropy is the evaluating criterion for working set selection, and the size of working set is determined at the process of iteration adaptively. The regression parameters are calculated by incremental learning and the calculation of inversing a large scale matrix is avoided. So the running speed is improved. This algorithm reserves well the sparseness of support vector and improves the learning speed.

1. Introduction

The support vector machine (SVM) is a novel learning method that is constructed based on statistical learning theory. The support vector machine has been studied widely since it was presented in 1995. It has been applied to pattern recognition broadly and its excellent performance has been shown in function regression problems. Training a standard support vector machine requires the solution of a large-scale quadratic programming problem. This is a difficult problem when the number of the samples exceeds a few thousands. Many algorithms for training the SVM have been studied. Suykens [1] suggested a least squares support vector machine (LSSVM) in which the inequality constrains were replaced by equality constrains [1]. By this way, solving a quadratic programming was converted into solving linear equations. So the efficiency of training SVM is improved greatly and the difficulty of training SVM is decreased. But all the training samples are selected as support vectors and the sparseness of support vectors is destroyed in LSSVM. LSSVM involves computing the inverse of a matrix in the process of training. It costs a lot of time and space to computing the inverse matrix for large scale training samples. This restricts LSSVM applied on large scale problems. Some researchers have involved in studying on the sparseness of support vectors and some results have been presented. Suykens[2] proposed a pruning scheme based on support vector spectrum. The basic idea is to sort the support vector according to the corresponding Lagrange multiplier and prune some support vectors with the smaller Lagrange multiplier. Recomputed using the rest support vectors till the performance of the learning machine decreased. This method computed the Lagrange multiplier for all the training samples at beginning so the cost on time and space is large. Wu [3] presented an adaptive and iterative algorithm for LSSVM training. The number of support vectors selected by this method is decided adaptively through the incremental and inverse learning. And it is much smaller than the number of training samples. However, the fitness for function regression is approach to the standard LSSVM. Espinoza [4-5] present a fixed-size LSSVM algorithm and applied it to load forecasting. The size of support vectors was fixed

* The authors are grateful to the support of Inner Mongolia National Science Foundation (Grant No.200711020807) and Inner Mongolia University for Nationalities Scientific Research Project (Grant No. MDB2007132, YB0706, MDK2007032)

** Corresponding author, E-mail: ycliang@jlu.edu.cn forehand and the maximum quadratic Renyi entropy was used to select the support vectors into working set. According to Wu[3] and Espinoza [4-5], combining the Renyi entropy with incremental learning algorithm, an adaptive and iterative LSSVM algorithm for regression (LSSVR) based on Renyi entropy is presented in this paper.

2. Least squares support vector regression

According to [2], let us consider a given training set of l samples $\{x_i, y_i\}_{i=1}^{l}$ with the ith input datum $x_i \in R^n$ and the ith output datum $y_i \in R$. The aim of support vector machine model is to construct the regression function takes the form:

$$f(x,w) = w^T \varphi(x) + b \tag{1}$$

where the nonlinear mapping $\varphi(\cdot)$ maps the input data into a higher dimensional feature space. In least squares support machine for function regression the following optimization problem is formulated

$$\min_{w,e} J(w,e) = \frac{1}{2} w^T w + \gamma \sum_{i=1}^{l} e_i^2 \tag{2}$$

subject to the equality constraints

$$y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1,..., l \tag{3}$$

This corresponds to a form of ridge regression. The Lagrangian is given by

$$L(w,b,e,\alpha) = J(w,e) - \sum_{i=1}^{l} \alpha_i \{w^T \varphi(x_i) + b + e_i - y_i\} \tag{4}$$

with Lagrange multipliers α_k. The conditions for the optimality are

$$\begin{cases} \dfrac{\partial L}{\partial W} = 0 \rightarrow w = \sum\limits_{i=1}^{l} \alpha_i \varphi(x_i) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum\limits_{i=1}^{l} \alpha_i = 0 \\ \dfrac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i \\ \dfrac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i = 0 \end{cases} \tag{5}$$

for $i = 1,...,l$. After eliminating e_i and w, we could have the solution by the following linear equations

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{6}$$

where $y = [y_1,..., y_l]^T, \vec{1} = [1,...,1]^T, \alpha = [\alpha_1,...,\alpha_l]^T$ and the Mercer condition

$$\Omega_{kj} = \varphi(x_k)^T \varphi(x_j) = \psi(x_k, x_j) \qquad k, j = 1,...,l \tag{7}$$

is applied.

Set $A = \Omega + \gamma^{-1} I$. For A is a symmetric and positive-definite matrix, A^{-1} exists. Solving the linear equations (6) we obtain the solution

$$\alpha = A^{-1}(y - b\vec{1}) \qquad b = \frac{\vec{1}^T A^{-1} y}{\vec{1}^T A^{-1} \vec{1}} \tag{8}$$

Substituting w in Eq. (1) with the first equation of Eqs. (5) and using Eq. (7) we have

$$f(x,w) = y(x) = \sum_{i=1}^{l} \alpha_i \psi(x, x_i) + b \tag{9}$$

where α_i and b are the solution to Eqs(6). The kernel function $\psi(\cdot)$ can be chosen as

(a) linear function $\psi(x, x_i) = x_i^T x$

(b) polynomial function $\psi(x, x_i) = (x_i^T x + 1)^d$

(c) radial basis function $\psi(x, x_i) = \exp\{-\|x - x_i\|_2^2 / \sigma^2\}$

3. Incremental learning

In order to obtain the regression function in Eq.(9), it needs to compute α and b. And computing A^{-1} is the key to compute them. It costs large on time and space while the number of training sample is large. Liu [6] presented an online LSSVM learning algorithm for function and classification. This algorithm avoids computing the inverse of a large matrix. The increment learning part of the algorithm is used in our paper.

The set which elements (some of training samples) are used to construct the classifier is called working set. Denotes as W. According to Eqs(6), set

$$A_N = \Omega + \gamma^{-1} I \qquad \overline{\alpha}_N = \alpha \qquad \overline{y}_N = y \tag{10}$$

where N is the number of samples in current working set. Eq. (8) can be rewritten as

$$\overline{b}_N = \frac{\vec{1}_N^T A_N^{-1} \overline{y}_N}{\vec{1}_N^T A_N^{-1} \vec{1}_N} \qquad \overline{\alpha}_N = A_N^{-1}(\overline{y}_N - b\vec{1}) \tag{11}$$

where $\vec{1}_N = \underbrace{(1,...,1)}_{N}{}^T$. The regression function is

$$f(x) = \sum_{i=1}^{N} \overline{\alpha}_{N,i} \psi(x, x_i) + \overline{b}_N$$

When a new coming sample (x_{N+1}, y_{N+1}) is added to the current working set, we could calculate the parameters according to Eq. (12)

$$\overline{b}_{N+1} = \frac{\vec{1}_{N+1}^T A_{N+1}^{-1} \overline{y}_{N+1}}{\vec{1}_{N+1}^T A_{N+1}^{-1} \vec{1}_{N+1}}$$

$$\overline{\alpha}_{N+1} = A_{N+1}^{-1}(\overline{y}_{N+1} - \overline{b}_{N+1} \vec{1}_{N+1}) \tag{12}$$

where $\vec{1}_{N+1} = \underbrace{(1,...,1)}_{N+1}{}^T$, $\overline{\alpha}_{N+1} = (\overline{\alpha}_N, \alpha_{N+1})$, $\overline{y}_{N+1} = (\overline{y}_N, y_{N+1})$.

A_{N+1} is constructed by added a line and a column to A_N. That is

$$A_{N+1} = \begin{bmatrix} A_N & q \\ q^T & s \end{bmatrix}$$

where $q=(\Omega_{1,N+1},\Omega_{2,N+1},...,\Omega_{N,N+1})^T$, $s=\Omega_{N+1,N+1}+\gamma^{-1}$. Then the regression function changes to

$$f(x)=\sum_{i=1}^{N+1}\overline{\alpha}_{N+1,i}\psi(x,x_i)+\overline{b}_{N+1}$$

According to the algorithm in reference Liu [6], the matrix A_{N+1}^{-1} in Eq. (12) could be calculated from matrix A_N^{-1} , that is

$$A_{N+1}^{-1}=\begin{bmatrix}A_N^{-1} & 0\\ 0 & 0\end{bmatrix}+\begin{bmatrix}-A_N^{-1}q\\ 1\end{bmatrix}\left[s-q^TA_N^{-1}q\right]^{-1}\begin{bmatrix}-q^TA_N^{-1}\\ 1\end{bmatrix} \quad (13)$$

In this way the calculation for the inverse of a large-scale matrix could be avoided.

4. Renyi Entropy

Entropy is the measure of the degree of the system randomization. It is related to the underlying density distribution of the sample. The larger the entropy is the more information involves in sample and the better the randomization of sample is. Renyi [7-8] gave the definition for density distribution function. The Renyi entropy of order α ($\alpha\geq 0,\alpha\neq 1$) of a continuous probability density functions $p(x)$ is defined in Eq. (14):

$$H_{R\alpha}=\frac{1}{1-\alpha}\log\int p^{\alpha}(x)dx \quad (14)$$

This paper is focus on Renyi's quadratic entropy, $\alpha=2$, because this leads to an important computational simplification obtained for Gaussian kernels. The expression of Renyi quadratic entropy is given by Eq.(15).

$$H_{R2}=-\log\int p^2(x)dx \quad (15)$$

$\int p^2(x)dx$ can be estimated by [9]

$$\int p^2(x)dx\approx\int\hat{p}^2(x)dx=\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\psi(x_i,x_j)=1_N^T\Omega 1_N \quad (16)$$

where each $N\times 1$ vector 1_N has each element equal to 1/*N*. So the quadratic Renyi entropy can be estimated by Eq.(17)

$$H_{R2}\approx-\log(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\psi(x_i,x_j)) \quad (17)$$

5. Adaptive and iterative least squares support vector regression based on Renyi entropy

There is no theory to instruct the number of support vectors which were selected into working set. An iterative algorithm is presented in this paper and the number of support vectors in working set is determined in the process of iteration. The quadratic Renyi entropy is the evaluating criterion for selecting the support vector and the sample that with big quadratic Renyi entropy is selected into working set.

Suppose the training set

$$T=\{s_i\mid s_i=(x_i,y_i),x_i\in R^n,y_i\in R,i=1,2,\cdots,l\}$$

The regression function is

$$f(x)\mid_{\mathbf{W}}=\sum_{i\in\mathbf{W}}\alpha_i\psi(x,x_i)+b$$

where W is the working set whose elements are support vectors used to construct regression function. α and b are decided by the working set.

Firstly, carry out initialization. The first N samples are selected to form the initial working set. Compute α and b using Eq.(8) and compute $f(x)\mid_W$. And then repeat the following operation till the stop criterion is reached: for the samples $s_i=(x_i,y_i)$ which are not in working set, put them into working set respectively and form the temporary working set $\hat{W}_i$. Compute the quadratic Renyi entropy for each temporary working set. The biggest quadratic Renyi entropy is selected and the corresponding sample s_j is put into working set W. Compute α and b using the new W by incremental learning algorithm.

Steps of the proposed algorithm are as follows:
Initialization: Set θ is the precision in training and testing, the precision in stop criterion is ε . Set $W=\{(x_1,y_1),...,(x_N,y_N)\}$ and calculate A^{-1} analytically. Calculate α and b according to Eq.(8) and obtain the regression function $f(x)\mid_W$.

```
(1) while the stop criterion is false do
(2)   for $i=1,...,l$ do
(3)     if $s_i\notin W$ then
(4)       $\hat{W}_i=W\cup\{s_i\}$   //$\hat{W}_i$ is temporary working set
          for added sample $s_i$
(5)       according to Eq.(17), compute the quadratic
          Renyi entropy $H_{R2}(\hat{W}_i)$ for temporary working
          set $\hat{W}_i$
(6)     end if
(7)   end for
(8)   find the maximum Renyi entropy in $H_{R2}(\hat{W}_i)$ ,
      denote $j=\arg\max_i\{H_{R2}(\hat{W}_i)\}$
(9)   $W=W\cup\{s_j\}$
(10)  calculate $\alpha$ and $b$ using the samples in working
      set $W$ by incremental learning
(11)  calculate the current object function $Q$
```

(12) end while

(13) using the samples in working set W and α , b to form the regression function $f(x)|_W$

In this algorithm, the stop criterion is related to the objective function. The objective function is

$$Q(w,e)|_W = \frac{1}{2}\|w\|^T_{s_i \in W} + \gamma \sum_{s_i \in W} e_i^2$$

where $w = \sum_{s_i \in W} \alpha_i y_i \varphi(x_i)$, $e_i = \frac{1}{\gamma}\alpha_i$. The meaning of the defined stop criterion is that the procedure ends when the relative error of objective values in the two adjacent iterations is smaller than a given precision ε

$$\frac{Q_{current} - Q_{last}}{Q_{last}} \le \varepsilon$$

where $Q_{current}, Q_{last}$ are the objection function value for this iteration and last iteration respectively.

6. Numerical Experiment

The experiments are implemented on a DELL PC, which utilizes a 2.8GHz Pentium IV processor with 512MB memory. The OS is Microsoft Windows XP operating system. All the programs are compiled under Microsoft's Visual C++ 6.0.

In order to examine the efficiency of the proposed algorithm and compare with LSSVR algorithm, numerical experiments are performed using three kinds of data sets. The first kind of data set is composed of the simply elementary functions which include

$$f(x) = \sin(x) \quad -15 \le x \le +15$$

$$f(x) = x^2 \quad -1 \le x \le +1$$

$$f(x) = x^3 \quad -1 \le x \le +1$$

These functions are used to test the regression ability for the known function. The second kind of data set is composed of Mackey-Glass (MG) system and sample function $f(x) = \sin c(x)$. The MG system is a blood cell regulation model established in 1977 by Mackey and Glass. It is a chaos system

$$\frac{dx}{dt} = \frac{a \cdot x(t-\tau)}{1 + x^{10}(t-\tau)} - b \cdot x(t)$$

described in [10], where $\tau = 17$ $a = 0.2$ $b = 0.1$ $\Delta t = 1$ $t \in (0,400)$. The embedded dimensions are $n = 4,6,8$ respectively. The sample function is

$$f(x) = \sin c(x) = \begin{cases} 1 & x = 0 \\ \frac{\sin(x)}{x} & x \ne 0 \end{cases} \quad -15 \le x \le 15$$

The third kind of data is spiral function. An RBF kernel function

$$\psi(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / (2\sigma^2))$$

is employed in these three kinds function. The parameters used in this algorithm are showed in Table1. The comparison between LSSVR and our algorithm are showed in Table 2, where the third column is the number of support vectors, the forth column is the seconds for training, and the fifth and seventh columns are the regression accuracy for training and testing, respectively. The regression accuracy is a ratio that is the number of samples whose relative error

$$e = \left| \frac{f(x) - y}{y} \right|$$

is smaller than θ to the number of samples in the working set (testing set). The sixth and eighth columns are the mean square error for training and testing, respectively.

Table 1. Parameters used in algorithm

data	γ	σ	θ	ε
sin	5000	0.5	0.01	0.0001
square	5000	0.1	0.01	0.0001
cube	5000	0.08	0.01	0.000001
sinc	5000	0.6	0.01	0.0001
MG 4	5000	0.3	0.01	0.0001
MG 6	5000	0.3	0.01	0.0001
MG 8	5000	0.3	0.01	0.0001
spiral	5000	1.0	0.01	0.000001

It can be seen from Table 2 that the learning speed of our algorithm is much faster than LSSVR. Moreover, the number of support vectors is less than that obtained by LSSVR for the similar regression accuracy. Table 3 shows the ratio of spending time and selected support vectors for the proposed algorithm to that for LSSVM. It can be seen that the training time spent on the proposed algorithm is 0.59%--15.4% of standard LS-SVM, and the number of support vector is 0.85%--6.54% of training sample. Figure 1-4 show the distribution of selected support vectors on sin, cubic, sinc and spiral function. The black spots denote the support vectors. It can be seen that the distribution of the support vector is homogeneous in the whole sample space and the distribution reflects the property of the sample space.

Table 2. Comparison between the proposed algorithm (RLSSVR) and standard LSSVR

data l*n	Algorithm name	# of SVs	Train time (CPU s)	Accuracy (train%)	MSE (train)	Accuracy (test%)	MSE (test)
sin	LSSVR	3000	303.734	99.93	3.58e-009	99.87	4.19e-009
3000×1	RLSSVR	123	26.0780	99.90	3.22e-008	99.73	3.28e-008
square	LSSVR	3000	303.875	97.10	6.14e-009	98013	7.35e-009
3000×1	RLSSVR	78	8.1870	98.10	1.49e-006	98.67	1.54e-006
cube	LSSVR	3000	300.093	96.6	8.00e-009	94.63	7.69e-009
3000×1	RLSSVR	72	5.4070	93.67	6.85e-006	92.30	5.24e-006
sinc	LSSVR	3000	295.000	99.93	8.23e-011	99.80	1.14e-010
3000×1	RLSSVR	74	8.2660	95.70	4.98e-008	95.37	5.28e-008
MG 4	LSSVR	6000	2960.12	100	1.44e-008	100	1.49e-008
6000×4	RLSSVR	58	17.6880	100	8.81e-007	100	9.72e-007
MG 6	LSSVR	6000	2916.65	100	8.06e-009	100	8.45e-009
6000×6	RLSSVR	155	166.6880	100	3.94e-007	100	4.03e-007
MG 8	LSSVR	6000	3105.77	100	3.30e-009	100	3.28e-009
6000×8	RLSSVR	51	22.3600	100	1.90e-006	100	1.94e-006
spiral	LSSVR	5000	1697.828	97.82	1.16e-004	97.60	1.13e-004
5000×1	RLSSVR	327	261.2500	95.96	4.63e-002	96.18	4.12e-002

Table 3. The ratio of support vector and speeding time for the proposed algorithm to that for LSSVR

data	sin	square	cube	sinc	MG 4	MG 6	MG 8	spiral
Ratio of support vector	4.10%	2.53%	2.40%	2.47%	0.97%	2.58%	0.85%	6.54%
Ratio of spending time	8.58%	2.69%	1.80%	2.80%	0.59%	5.72%	0.72%	15.4%

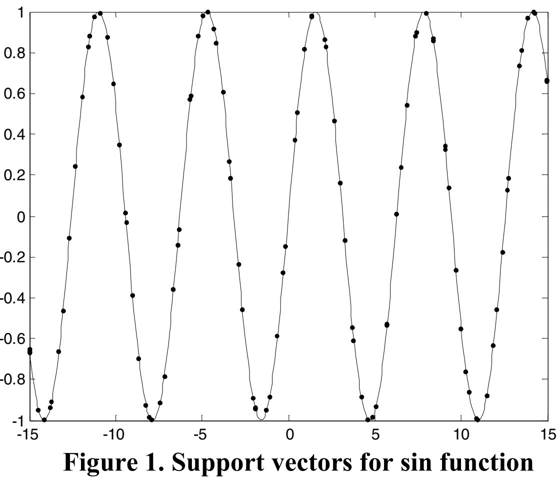

Figure 1. Support vectors for sin function

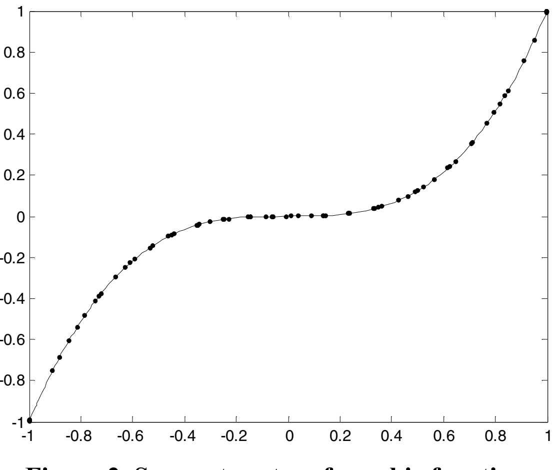

Figure 2. Support vectors for cubic function

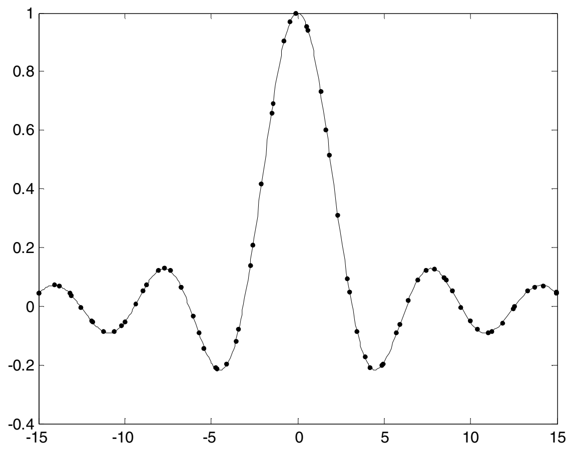

Figure 3. Support vectors for sinc function

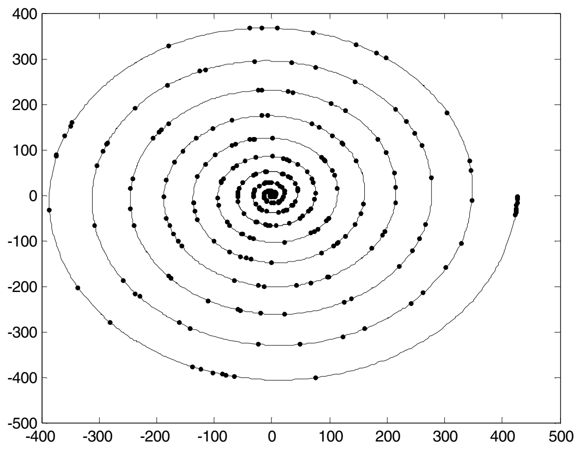

Figure 4. Support vectors for spiral function

7. Conclusion and discussion

An adaptive and iterative LS-SVR algorithm based on quadratic Renyi entropy is presented. All the training samples are support vectors in LS-SVM. So, LS-SVM loses the sparseness of support vector which is one of the important advantages of conventional SVM. The proposed algorithm overcomes this drawback. The quadratic Renyi entropy is the evaluating criterion for working set selection, and the size of working set is determined at the process of iteration adaptively. For the entropy is a measure of system randomization, the big entropy means that the system is randomized well. The working set with big entropy could reflect the law of the sample set. LS-SVM constructed by this working set has better generalization. The regression parameters are calculated by incremental learning and the calculation of inversing a large scale matrix is avoided. So the running speed is improved. We experimented on several datasets. The training time spent on the proposed algorithm is 0.59%--15.4% of standard LS-SVM, and the number of support vector is 0.85%--6.54% of training sample for the similar regression accuracy and small mean square error. This algorithm reserves well the sparseness of support vector and improves the learning speed.

References

[1] Suykens, J. A. K., Vandewalle, J. "Least squares support vector machine classifiers", *Neural Processing Letter*, 9(1999), pp. 293-300.

[2] Suykens, J.A.K., Lukas, L., Wandewalle, J. "Sparse approximation using least squares support vector machines", *In Proceeding of the IEEE International Symposium on Circuits and Systems (ISCAS 2000)*, (2000), pp. 757-760

[3] Wu, C. G. "Study on Generalized Chromosome Genetic Algorithm and Iterative Least Squares Support Vector Machine Regression", *doctoral dissertation*, Jilin university, (2006)

[4] Espinoza, M., Suykens, J.A.K., Moor, B.D. "Load forecasting using fixed-size least squares support vector machines". 8th International Workshop on Artificial Neural Networks, IWANN 2005, *Lecture Notes in Computer Science*, 3512(2005), pp. 1018-1026

[5] Espinoza, M., Suykens, J.A.K., Moor, B.D. "Fixed-size least squares support vector machines: a large scale application in electrical load forecasting". *Computational Management Science*, 3(2006), pp. 113–129

[6] Liu, J. H., Chen, J.P. et al. "Online SL-SVM for function and classification". *Journal of University of Science and Technology*, 10(5) (2003), pp,73-77

[7] Renyi, A. "On measures of entropy and information". *Proceedings of the Fourth Berkeley Symposium on Mathematics, Statistics and Probability*, University of California Press, Berkeley, CA, 1(1961), pp. 547-561

[8] Renyi, A. "Introduction a la theorie de linformation". *Calcul des probabilites*. Dunod, Paris, (1966)

[9] Girolami, M. "Orthogonal series density estimation and the kernel eigenvalue problem". *Neural Computation*, 14(3) (2002), pp. 669-688

[10]Flake G.W., Lawrence S. "Efficient SVM Regression Training with SMO". Machine Learning, 46 (2002), pp. 271–290.