Open Science Index, Bioengineering and Life Sciences Vol:4, No:10, 2010 publications.waset.org/10270.pdf

A Comparison of SVM-based Criteria in Evolutionary Method for Gene Selection and Classification of Microarray Data

Rameswar Debnath, *Member, IEEE,* and Haruhisa Takahashi

Abstract—An evolutionary method whose selection and recombination operations are based on generalization error-bounds of support vector machine (SVM) can select a subset of potentially informative genes for SVM classifier very efficiently [7]. In this paper, we will use the derivative of error-bound (first-order criteria) to select and recombine gene features in the evolutionary process, and compare the performance of the derivative of error-bound with the error-bound itself (zero-order) in the evolutionary process. We also investigate several error-bounds and their derivatives to compare the performance, and find the best criteria for gene selection and classification. We use 7 cancer-related human gene expression datasets to evaluate the performance of the zero-order and first-order criteria of error-bounds. Though both criteria have the same strategy in theoretically, experimental results demonstrate the best criterion for microarray gene expression data.

Keywords—support vector machine, generalization error-bound, feature selection, evolutionary algorithm, microarray data

I. Introduction

Patient samples for bioinformatic analyses are fairly small in number compared to the number of genes investigated such as microarray datasets. The vast amount of raw gene expression data leads to statistical and analytical challenges including the classification of datasets into correct classes. In machine-learning terminology, these data sets have high dimension and small sample size. Though the data management system allows researchers to gather number of genes of ever-increasing size, many of which are irrelevant to the distinction of samples. These irrelevant genes have negative effect on the accuracy of the classifier. The microarray data also contain technical and biological noise. Selection of relevant genes that will give higher accuracy for sample classification (for example, to distinguish cancerous from normal tissues) is a common task in most microarray data studies. There exist several ranking based and evolutionary computation methods for gene selection in the microarray data. Gene selection by evolutionary methods can outperform others, however; the success of these evolutionary methods depends on the appropriate choice of selection and recombination operations as well as choice of the appropriate classifier.

Recently, Debnath and Kurita have proposed an evolutionary SVM classifier that can select a subset of potentially informative genes based on SVM error-bound value for an SVM classifier very efficiently [7]. In the conventional gene selection and classification evolutionary methods, the selection and recombination operations are obtained based on GA-based algorithms, whereas fitness function is evaluated by classifiers such as SVM, kNN, and neural networks. The problem of these evolutionary methods is that the selection and recombination operations that select informative genes are independent from the algorithm used to construct the classifiers and thus selection and recombination operations do not directly reflect the performance of the classifier. The advantage of the error-bound based evolutionary method over conventional evolutionary methods is that the selection and recombination operators are chosen based on SVM error-bound values, whereas the SVM evaluates the fitness value. Thus, selected genes directly reflect to some extent the performance of SVM classifiers. Experimenting on various datasets, it is found that the error-bound based evolutionary method can select a subset of potentially informative genes for SVM classifier very efficiently [7].

In this paper, we will use derivative of error bound (first-order criteria) to select and recombine gene features in the evolutionary process, and compare the performance of the derivative of the error-bound with the error-bound itself (zero-order) in the evolutionary process. We also investigate several error-bounds such as radius-margin bound [12], Opper-Winther bound [8], Jaakkola-Haussler bound [9] and Zhou-Tuck bound [10], and their derivatives and the derivative of weight vector ($\nabla||\mathbf{w}||^2$) to compare the performance, and find the best criteria for gene selection and classification. We use 7 cancer-related human gene expression datasets to evaluate the performance of the zero-order and first-order criteria of error-bounds. Though both criteria have the same strategy in theoretically, from experimental results we see that zero-order criteria show better classification accuracy than first-order criteria. Among zero-order criteria, Opper-Winther bound and Zhou-Tuck bound perform better than others.

The paper is organized as follows: In Section II, we briefly describe the SVM classifier and its several error bounds and their derivatives. In Section III, we briefly describe the SVM

Rameswar Debnath and Haruhisa Takahashi are with the Department of Informatics, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan e-mail: {rdebnath,takahasi}@ice.uec.ac.jp

error bound based evolutionary algorithm that is presented in [7]. Computational results are presented in Section IV. Section V concludes the paper.

II. SVM CLASSIFIER

The SVM is a very popular supervised learning algorithm that often achieves superior generalization performance compared to other learning algorithms across most domains and tasks. The SVM classifier is a binary classifier that finds an optimal hyperplane as a decision function in a high dimensional space. Given l training examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)$, where $\mathbf{x}_i \in R^d, i = 1, \ldots, l$ and $y_i \in \{1, -1\}$ is the class label of $\mathbf{x}_i$. The method consists in first mapping $\mathbf{x}$ into a high dimensional space via a function Φ, then computes a decision function that does the separation with maximizing margin as:

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b, \tag{1}$$

where $\mathbf{w}$ is a weight vector and b is a bias. Assume that the nearest points lie on $f(\mathbf{x}_i) = \pm 1$ for some i, the margin is then defined by

$$\gamma = \frac{1}{\parallel \mathbf{w} \parallel^2}. \tag{2}$$

The SVM problem is expressed by the following optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 & (3) \\ \text{subject to} \quad & y_i(\mathbf{w}^T \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, \ldots, l. & (4) \end{aligned}$$

Using the Lagrangian, this new optimization problem can be converted into a dual form, which is a quadratic programming problem defined by

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) & (5) \\ \text{subject to} \quad & \sum_{i=1}^{l} \alpha_i y_i = 0, & (6) \\ & \alpha_i \geq 0, \quad i = 1, \ldots, l, & \end{aligned}$$

where α_i are Lagrange multipliers and $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ is the Gram matrix of the training data. The $\mathbf{w}$ is then computed as

$$\mathbf{w} \quad = \quad \sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{x}_i) \tag{7}$$

and b is computed by taking any $\mathbf{x}_j$ corresponding to $\alpha_j > 0$ as

$$b = y_j - \sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_j). \tag{8}$$

For misclassified examples, slak variables and trade-off parameter are introduced in Eqs. (3) and (4). The solutions of the new problems are obtained using the Lagrangian theory where $\mathbf{w}$ is the same as derived previously but b is computed with different conditions. For mathematical formulations and derivations of algorithms for misclassified examples, see [12]. The interesting point of SVMs is that they are provided with many statistics that allow to estimate their generalization performance from bounds on the leave-one-out error. The leave-one-out error is an unbiased estimate for the true error rate of a classifier. Several error bound theories for binary SVMs exist. In this paper, we apply the following bounds and their derivatives, and the derivative of weight vector for feature selection. We briefly describe several error bounds and their derivative in the following subsections.

A. Radius-margin Bound

Vapnik [12] has developed the radius-margin bound for hard-margin SVM on the number of errors in the leave-one-out procedure without bias term b given as

$$loo \leq \frac{4}{l} R^2 ||\mathbf{w}||^2, \tag{9}$$

where loo is the leave-one-out error rate, $||\mathbf{w}||^2$ is the weight vector, and R is the radius of the smallest spare containing all $\mathbf{x}_i$. R^2 is computed by solving the following optimization problem:

$$\begin{aligned} R^2 \quad = \quad \text{maximize} \quad & \sum_{i=1}^{l} \beta_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^{l} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{l} \beta_i = 1, \\ & \beta_i \geq 0, \quad i = 1, \ldots, l. \end{aligned}$$

To calculate the derivative of the bound value, a virtual scaling factor (whose value is 1) is introduced with each feature, and then the gradient of bound value is computed with respect to that scaling factor. Thus $K(\mathbf{x}_i, \mathbf{x}_j)$ becomes:

$$K(\boldsymbol{\nu} \cdot \mathbf{x}_i, \boldsymbol{\nu} \cdot \mathbf{x}_j)$$

where $\cdot$ denotes the componentwise vector product and $\boldsymbol{\nu}$ is a vector with all elements are 1. The radius-margin gradient with respect to a feature i:

$$\begin{aligned} \frac{1}{l} \Bigg| & \parallel \mathbf{w} \parallel^2 \sum_{k,j} (\beta_k \beta_j - \beta_k \delta_{k,j}) \frac{\delta K(\boldsymbol{\nu} \cdot \mathbf{x}_k, \boldsymbol{\nu} \cdot \mathbf{x}_j)}{\delta \nu_i} \\ & + R^2 \sum_{k,j} \alpha_k \alpha_j y_k y_j \frac{\delta K(\boldsymbol{\nu} \cdot \mathbf{x}_k, \boldsymbol{\nu} \cdot \mathbf{x}_j)}{\delta \nu_i} \Bigg| \end{aligned} \tag{10}$$

where $\delta_{jk} = 1$ if $j = k$, otherwise $\delta_{jk} = 0$.

B. Opper-Winther Bound

Opper and Winther [8] have developed the following bound on the number of errors in the leave-one-out procedure for the hard-margin SVM without bias term b is given as

$$loo \leq \frac{1}{l}\sum_{i=1}^{l}\Psi\left(\frac{\alpha_i}{\left(\boldsymbol{K}_{SV}^{-1}\right)_{ii}}-1\right), \tag{11}$$

where $\boldsymbol{K}_{SV}$ is the matrix of dot-products between support vectors. The step function is not a good choice for feature selection due to the small number of samples in microarray datasets. Avoiding step function, an upper bound of Opper-Winther bound is used for feature selection as

$$loo^{upper} \leq \frac{1}{l}\sum_{p=1}^{l}\frac{\alpha_p}{\left(\boldsymbol{K}_{SV}^{-1}\right)_{pp}}. \tag{12}$$

Opper-Winther bound gradient with respect to a feature i:

$$\frac{1}{l}\left|\sum_{p=1}^{l}\alpha_p S_p^2\left(\boldsymbol{K}_{SV}^{-1}\frac{\delta\boldsymbol{K}_{SV}}{\delta\nu_i}\boldsymbol{K}_{SV}^{-1}\right)_{pp}\right| \tag{13}$$

where $S_p = 1/\left(\boldsymbol{K}_{SV}^{-1}\right)_{pp}$.

C. Jaakkola-Haussler Bound

Jaakkola and Haussler [9] have developed the following bound on the number of errors in the leave-one-out procedure for SVM without bias term b given as

$$loo \leq \frac{1}{l}\sum_{i=1}^{l}\Psi\left(\alpha_i K(\mathbf{x}_i,\mathbf{x}_i)-1\right), \tag{14}$$

where Ψ is a step function: $\Psi(x)=1$ when $x>0$ and $\Psi(x)=0$ otherwise. Avoiding step function, it can be written as:

$$loo^{upper} \leq \frac{1}{l}\sum_{i=1}^{l}\alpha_i K(\mathbf{x}_i,\mathbf{x}_i) \tag{15}$$

Jaakkola-Haussler bound gradient:

$$\frac{1}{l}\sum_{k=1}^{l}\alpha_k\frac{\delta K(\boldsymbol{\nu}\cdot\mathbf{x}_k,\boldsymbol{\nu}\cdot\mathbf{x}_k)}{\delta\nu_i} \tag{16}$$

D. Zhou-Tuck Bound

Recently, Zhou and Tuck [10] have proposed an error bound for the SVM, which can be applicable in either separable or non-separable cases. The bound is given as

$$loo \leq \sqrt{\frac{1}{l^2}\boldsymbol{W}(\alpha)\left[\sum_p\left(D_p^2+{D'_p}^2\right)+\frac{4n_{SV}}{C}\right]}, \tag{17}$$

where $\boldsymbol{W}(\alpha)$ is the objective value of the dual problem generated from SVM classification, $\sum_p(\cdot)$ indicates that the sum is taken only over support vectors $\mathbf{x}_p$, D_p is the Euclidean distance between support vector $\mathbf{x}_p$ and its nearest support vector in the same class, while D'_p is the Euclidean distance between support vector $\mathbf{x}_p$ and its farthest sample (not support vector) in the other class, and n_{SV} is the number of support vectors. The Euclidean distance in the feature space, $\phi(\cdot)$, is calculated using the kernel function as

$$D_p = \min||\Phi(\mathbf{x}_p)-\Phi(\mathbf{x}_k)||_2$$

$$= \min\sqrt{\langle\Phi(\mathbf{x}_p),\Phi(\mathbf{x}_p)\rangle-2\langle\Phi(\mathbf{x}_p),\Phi(\mathbf{x}_k)\rangle+\langle\Phi(\mathbf{x}_k),\Phi(\mathbf{x}_k)\rangle}$$

$$= \min\sqrt{K(\mathbf{x}_p,\mathbf{x}_p)-2K(\mathbf{x}_p,\mathbf{x}_k)+K(\mathbf{x}_k,\mathbf{x}_k)},$$

where $\mathbf{x}_k$ are the support vectors in the same class of $\mathbf{x}_p$. Gradient of $W(\alpha)$:

$$\left|\sum_{k,j}\alpha_k\alpha_j y_k y_j\frac{\delta K(\boldsymbol{\nu}\cdot\mathbf{x}_k,\boldsymbol{\nu}\cdot\mathbf{x}_j)}{\delta\nu_i}\right| \tag{18}$$

Gradient of distance:

$$\min 1/2\left(\sqrt{\frac{\delta K(\boldsymbol{\nu}\cdot\mathbf{x}_p,\boldsymbol{\nu}\cdot\mathbf{x}_p)}{\delta\nu_i}-2\frac{\delta K(\boldsymbol{\nu}\cdot\mathbf{x}_p,\boldsymbol{\nu}\cdot\mathbf{x}_k)}{\delta\nu_i}+\frac{\delta K(\boldsymbol{\nu}\cdot\mathbf{x}_k,\boldsymbol{\nu}\cdot\mathbf{x}_k)}{\delta\nu_i}}\right) \tag{19}$$

III. Evolutionary SVM

The evolutionary algorithm that we use maintains a population of predictors whose effectiveness can be determined by using them as features in an SVM classifier. The initial predictors in the population are randomly constructed from the gene features set. Instead of applying crossover and mutation operations, the method selects and recombines new features based on leave-one-out error bound values of SVMs and the frequency of occurrence of the features in the evolutionary approach. If T_m is the bound value of m gene features on a predictor and T_{m-1}^i is the bound value of all genes except gene i of that predictor. Then, T_{m-1}^i for all i are calculated. The $T_{m-1}^j < T_{m-1}^k$ means removing gene j from the predictor can reduce error bound much more than removing gene k. Thus genes j with small T_{m-1}^j should be deleted in the next generation. Again, if T_{m+1}^i is the bound value of m genes on a predictor plus a new gene i. The $T_{m+1}^j < T_{m+1}^k$ means adding gene j to the predictor can reduce error bound much more than adding gene k. The k-fold cross validation is used as an estimator of the generalization performance that also measures the fitness value. The termination criteria is defined using both the maximum number of generations and the criteria of no improvement of maximum fitness value of the population. The algorithm is described below:

1. A population E_0 of n predictors $\{G_1, G_2, ..., G_n\}$ is created. A predictor G_i is a subset of m gene features $\{g_1, g_2, ..., g_m\}$ initially created randomly. Evaluate the fitness values of all predictors. Fitness values are evaluated by SVMs.
2. Until termination criteria not satisfied, do the following:

3. For each predictor $G_i \in E_k$, create a new predictor G'_i
   - 3.1. Delete p genes from G_i, whose error bound values are minimum and selected in a few previous generations as briefly described above. For details, see [7].
   - 3.2. Add the same number of p genes from a random subset of data except those are in G_i in population E_k whose error bound values are minimum with the rest of the genes in G_k after deletion and frequently selected in the previous generations.
   - 3.3. Compute the fitness function for the new predictor G'_i using SVMs.
4. Create a new population E_{k+1} by replacing all new G'_i.
5. Replace some worse predictors of the new population E_{k+1} based on classification accuracy by some best predictors from the previous generation. To do this, merge the features of some best predictors from the previous generation and then randomly split features of the merged features set into the same number of predictors. Then select some predictors for new G'_i.

This procedure will be performed for a set of SVM hyperparameters and the best hyperparameters for each predictor will be obtained. Different combinations of genes with the same high accuracy rate can be evaluated in evolutionary computations through generation of individuals of a population. From this procedure we will get n' feature sets according to the best high classification accuracies where $n' \leq n$. From the n' sets we will choose N_{best} features according to occurrence frequency and classification accuracy rate. The hyperparameters for the final learning machine (SVM) will be selected by averaging the best hyperparameters of that n' predictors. For details about the algorithm and principles behind these, see [7].

IV. Computational Experiments

In our experiments, we use 7 cancer-related human gene expression datasets that are described in Table I. The dataset are available on http://www-gems-system.org for non-commercial use. The studied datasets were produced primarily by oligonucleotide-based technology. Specifically, all datasets except for SRBCT, RNA were hybridized to high-density oligonucleotide Affymetrix arrays HG-U95 or Hu6800, and expression values (average difference units) were computed using Affymetrix GENECHIP analysis software. The SRBCT dataset was obtained by using two-color cDNA platform with consecutive image analysis performed by DeArray Software and filtering for a minimal level of expression. The datasets have 2-5 distinct diagnostic categories, 50-102 patient samples, and 2308-11225 variables (gene features) after preprocessing (details in [6]). We rescale gene expression values of these datasets linearly into the range [-1,1].

The number of predictors is set to 50. The size of each predictor and the numbers of deletions and additions of genes are set experimentally (usually half of the predictor is deleted and added in our experiments). In each generation, at best 10 worst predictors in the new population is replaced with the 10 best predictors of the previous population according to step 5 of the algorithm. To evaluate the performance of the proposed method we use 5-fold cross-validation on each dataset. The stopping condition of the algorithm is to use 100 generations. Only linear kernel is used for experiments because linear kernel shows better performance than RBF kernel. The SVM trade-off parameter is set to $[2^{-2}, 2^{-1}, \ldots, 2^9, 2^{10}]$.

The comparison of performances of zero-order and first-order SVM error-bound criteria in the evolutionary process is shown in Table II. We performed experiments on all error-bounds and their derivatives but reported the results in the table of the bounds and their derivatives that show the best performance. From the experimental results we see that zero-order criteria show better performance than first-order criteria. Among the zero-order criteria, Opper-Winther bound and Zhou-Tuck bound perform better than others. Among the first-order criteria, derivative of weight vector shows the best results and then the derivative of Zhou-Tuck and Opper-Winther bounds. Other derivatives are not as good as derivatives of weight vector and Zhou-Tuck bound. From the experiments and theoretical analyses, we see that computational cost using weight derivative is minimum, followed by the cost of Jaakkola-Haussler bound and Zhou-Tuck bound, then Opper-Winther upper bound, and then the cost of radius-margin bound. The derivative of each error-bound is computationally more expensive than the cost of itself. Regarding the accuracy rate and computational cost, Opper-Winther bound and Zhou-Tuck bound perform better than others. Thus, we suggest that Opper-Winther bound and Zhou-Tuck bound are more suitable for practical use.

V. Conclusion

In this paper, we compare the performance of zero-order and first-order SVM error-bound criteria in the evolutionary process. From the experimental results we see that zero-order criteria show better performance than first-order criteria. Moreover, zero-order criteria are computationally less expensive than first-order criteria. Only derivative of weight vector shows better results among the derivative criteria, however it's results are not as good as zero-order criteria. Regarding computational complexity among all, the cost of the derivative of weight vector is the least. However, considering both accuracy rate and computational cost, Opper-Winther bound and Zhou-Tuck bound perform better than all zero-order and first-order criteria.

Acknowledgment

This work is supported by Grant-in-Aid for JSPS Fellows from the Ministry of Education, Culture, Sports, Science and

Table I: Features of microarray datasets.

Dataset	Diagnostic Task	#Samples	#Genes	#Classes
Brain_Tumor1	Five human brain tumor types	90	5920	5
Brain_Tumor2	Four malignant glioma types	50	10367	4
SRBCT	Small, round blue cell tumors of childhood	83	2308	4
Leukemia1	Accute myelogenous leukemia (AML), acute lympboblastic leukemia (ALL) B-cell, and ALL T-cell	72	5327	3
Leukemia2	AML, ALL, and mixed-lineage leukemia (MLL)	72	11225	3
Prostate_Tumor	Prostate tumor and normal tissue	102	10509	2
DLBCL	Diffuse large B-cell lymphomas and follicular lymphomas	77	5469	2

Table II: Mean accuracy (Ac.) rate of the zeor-order and first-order criteria in the evolutionary method. Here ‘OW’, ‘ZT’, ‘RM’, and ‘JH’ represent the Opper-Winther bound, Zhou-Tuck bound, radius-margin bound, and Jaakkola-Haussler bound, respectively.

Dataset	#Genes	Zeor-order Criteria		First-order Criteria	
	(Selected)	Ac. Rate (%)	Bounds	Ac. Rate (%)	Bound Derivatives
Brain_Tumor1	6	**97.84**	OW	93.39	$\nabla\|\|\mathbf{w}\|\|^2$
Brain_Tumor2	5	**100.0**	OW/ZT/RM/JH	**100.0**	$\nabla\|\|\mathbf{w}\|\|^2/\nabla$ZT
SRBCT	4	**100.0**	OW/ZT/RM/JH	**100.0**	∇OW
Leukemia1	3	**100.0**	OW/ZT/RM/JH	**100.0**	$\nabla\|\|\mathbf{w}\|\|^2$
Leukemia2	3	**100.0**	OW/ZT/RM/JH	**100.0**	$\nabla\|\|\mathbf{w}\|\|^2/\nabla$OW/$\nabla$ZT
Prostate_Tumor	3	**100.0**	OW/ZT	97.10	$\nabla\|\|\mathbf{w}\|\|^2$
DLBCL	3	**100.0**	OW/ZT/RM/JH	**100.0**	$\nabla\|\|\mathbf{w}\|\|^2/\nabla$OW/$\nabla$ZT

Technology of Japan.

REFERENCES

[1] T. Jirapech-Umpai and S. Aitken, “Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes”, *BMC Bioinformatics*, vol. 6, no. 148, 2005.

[2] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, “Gene selection for cancer classification using support vector machines”, *Machine Learning*, vol. 46, pp. 389-422, 2002.

[3] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, “Feature selection for svms”, *Advanced in Neural Information Processing Systems 13*, 2001.

[4] H.-L. Huang and F. -L. Chang, “ESVM: Evolutionary support vector machine for automatic feature selection and classification of microarray data”, *Bio Systems*, vol. 90, pp. 516-528, 2007.

[5] A. Rakotomamonjy, “Variable selection using SVM-based criteria”, *Journal of Machine Learning Research*, vol. 3, pp. 1357-1370, 2003.

[6] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin and S. Levy, “A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis”, *Bioinformatics*, vol. 21, no. 5, pp. 631-643, 2005.

[7] R. Debnath and T. Kurita, “An evolutionary approach for gene selection and classification of microarray data based on SVM error-bound theories”, *BioSyatems*, vol. 100, issue 1, pp. 39-46, 2010.

[8] M. Opper and O. Winther, “Gaussian process and SVM: Mean field and leave-one-out”, Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D. (Eds.), *Advances in large margin classifiers*, Cambridge, MA:MIT Press, pp. 311-326, 2000.

[9] T.S. Jaakkola and D. Haussler, “Probabilistic kernel regression models”, in *Proc. 1999 Conference on AI and Statistics*, Floria, USA, 1999.

[10] X. Zhou, and D. P. Tuck, “Gene selection using a new error bound for support vector machines”, in *Proc. Eleventh Annual International Conference on Research in Computational Molecular Biology*, San Francisco, USA, 2007.

[11] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, “Choosing multiple parameters for support vector machines”, *Machine Learning*, vol. 46, pp. 131-159, 2002.

[12] V. Vapnik, *Statistical Learning Theory*, New York:Wiley, 1998.