An Evolutionary Gene Selection Method for Microarray Data Based on SVM Error Bound Theories

R. Debnath, and T. Kurita
Neuroscience Research Institute
AIST, Tsukuba, Ibaraki, 305-8568, Japan
Email: ramesward@gmail.com, takio-kurita@aist.go.jp

Abstract— *Microarrays have thousands to tens of thousands of gene features but patient samples are fewer or a few hundred. Identifying genes whose disruption causes congenital or acquired disease is the fundamental problem in microarray data analysis. In this paper, we propose an efficient evolutionary SVM-based classifier that can select smaller number of features with high accuracy. The proposed method uses SVM with a given subset of features to evaluate the fitness function, and new subset of features are selected based on several leave-one-out error bounds for the SVM classifier and the frequency of occurrence of the features in the evolutionary approach. We test our proposed method on different microarray data and find that the proposed method can obtain high classification accuracy with a smaller number of selected genes.*

Keywords: support vector machine, radius-margin error bound, Jaakkola-Haussler error bound, Opper-Winther error bound, evolutionary algorithm

1. Introduction

Numerous problems of bioinformatics have characteristics that patient samples are fairly small in number compared to the number of genes investigated, such as microarray datasets. The fundamental problem in microarray data analysis is to find a smaller size subset of genes which is responsible for a specific interest (such as cancer disease diagnosis). This problem can be treated as a machine learning with feature selection problem. Though the feature selection can be applied to both supervised or unsupervised learning, we focus here on the problem of supervised learning (classification). Several supervised learning techniques such as neural networks, k- nearest neighbor (kNN) and support vector machine (SVM), kernel based classifiers, etc., have been successfully applied to microarray data analysis in the recent years [1]-[4]. Moreover, the SVM-based classifier is the superior for its robustness to a small number of high dimensional samples.

In general, feature selection can improve the performance of the machine. The SVM does not offer the mechanism of automated internal relevance detection and hence the feature selection is often performed as a prepossessing step of the actual learning algorithm. The principle component analysis (PCA), kernel PCA (KPCA) can be used for dimension reduction where the methods transform the input space into another reduced space without loosing important information about the samples [5]-[7]. However, given the low number of samples relative to the very large number of genes, the PCA method is not suitable for gene-oriented analysis [7]. There exist several evolutionary approaches such as GA-kNN, GA-SVM, and other evolutionary SVM/kNN where kNN and SVM is used to evaluate the fitness function of the GA-based methods [8]-[10], and genetic programming [11], etc. These methods are capable of selecting a smaller subset of genes for sample classification. However, multiple sets of relevant features may have the same high training accuracy as the training dataset size is fairly small compared to the number of candidate features and resulting in model uncertainty. To cope the model uncertainty, some useful approaches have been proposed, such as boosting algorithm [12] and model averaging [13], etc.

In this paper, we propose an efficient evolutionary SVM-based classifier that can select smaller number of features with high accuracy. The proposed method uses SVM with a subset of features to evaluate the fitness function, and new subset of features are selected based on several leave-one-out error bounds for the SVM classifier and the frequency of occurrence of the features in the evolutionary approach. To select new features according to error bound theories, we consider radius-margin bound [21] for *L2-SVM*, and compare the classification accuracy on different microarray data. We use k-fold cross-validation as an estimator of the generalization ability. We also describe Jaakkola-Haussler bound [18] and Opper-Winther bound [20] that can be applied to our proposed algorithm, however experimental results are not presented.

The reminder of this paper is organized as follows. In Section 2, we briefly describe the support vector machine. We describe various error bounds for the SVM that can be used for gene selection in Section 3. In Section 4, we describe the proposed evolutionary algorithm. The computational results will be shown in Section 5. Section 6 concludes the paper.

2. Support Vector Machine

The support vector machine (SVM) is very popular algorithm for solving pattern recognition, regression and density

estimation problems, etc., and has already outperformed most of the machine learning algorithms. Theoretically, the support vector machine approximately implements the structural risk minimization principle, thus the support vector machine is situated on a strong theoretical foundation. This is a linear classifier that maximizes the margin between separating hyperplane and the data points. The SVM has no local minima, i.e., it solves a convex optimization problem. The algorithm can automatically determine the network architecture. For these why, it has attracted more in the application areas than the other neural networks. Basically SVM is designed for binary classification problems and many different forms of SVM algorithms have been introduced for different purposes. In this section, we describe only the binary SVM classifier. 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 $\mathbf{x}_i$. If these training examples are linearly separable in the input space, we may write the decision function that does the separation is as:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0, \tag{1}$$

where $\mathbf{w}$ is a weight vector and b is a bias. The SVM finds the separating hyperplane of the classes where the distance of either class from the hyperplane is maximum. Assume that the nearest points lie on $f(\mathbf{x}_i) = \pm 1$ for some i, the margin is then defined by

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

The SVM problem is expressed by the following optimization problem:

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

This problem is known as *hard margin* SVM. When the training data is not linearly separable in the input space, we introduce slack variables $\xi_i (> 0)$ into equations (3)-(4) as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{l} \xi_i && (5) \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, && (6) \\ & \xi_i \geq 0, \quad i = 1, \ldots, l. \end{aligned}$$

where C is a parameter that determines the tradeoff between the maximum margin and the minimum classification error. This form of SVM is known as the *1-norm soft margin* (L1-) SVM. Using the Lagrangian, this new optimization problem can be converted into a dual form which is a quadratic programming problem defined by

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle && (7) \\ \text{subject to} \quad & \sum_{i=1}^{l} \alpha_i y_i = 0, && (8) \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l, \end{aligned}$$

where α_i are Lagrange multipliers, $\langle \mathbf{x}_i, \mathbf{x} \rangle$ is inner-product. The $\mathbf{w}$ is then computed as:

$$\mathbf{w} \quad = \quad \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \tag{9}$$

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

$$b = y_j - \sum_{i=1}^{l} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{10}$$

It often happens that a sizable fraction of the l values of α_i is zero. Only the points lie closest to the hyperplane including those on the wrong side of the hyperplane are corresponding to non-zero α_i's. These points $\mathbf{x}_i$'s are called support vectors. When training data is not separable in the input space then it is transformed into a high dimensional non-linear feature space, and the inner-product is calculated using kernel function without considering the feature space itself, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. The requirement of the kernel function is to satisfy Mercer's theorem. Common types of kernels are Gaussian, polynomial, and sigmoidal kernels.

There is another choice $C \sum_{i=1}^{l} \xi_i^2$ against $C \sum_{i=1}^{l} \xi_i$ on the problem in equation (5). The problem is then known as the *2-norm soft margin* (L2-) SVM problem. The dual problem of the L2-SVM is as:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{C} \delta_{ij} \right) \\ \text{subject to} \quad & \sum_{i=1}^{l} \alpha_i y_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, \ldots, l, \end{aligned}$$

where $\delta_{ij} = 1$ if $i = j$, otherwise $\delta_{ij} = 0$. The computation of $\mathbf{w}$ and b is the same as that for the L1-SVM problem.

3. Bounds on Generalization

There have been several error bound theories developed for SVM and some bounds are useful to select hyperparameters of SVM for good performance. In this paper we will use error bound theories for new feature selection in the evolutionary approach. In this section, we describe several error bound theories. These bound theories are developed

for hard margin SVM. If the training data is non-separable then L2-SVM is considered. In this case $K(\mathbf{x}_i, \mathbf{x}_j) \leftarrow K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{C}\delta_{ij}$.

3.1 Radius-margin Bound

Vapnik has developed the radius-margin bound on the number of errors in the leave-one-out (loo) procedure without bias term b and with no training error given as:

$$loo \leq \frac{4}{l} R^2 ||\boldsymbol{w}||^2 \tag{11}$$

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

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

This bound is differentiable and it may use for hyperparameters selection for SVM [16]-[17]. In [15], Rakotomamonjy has applied this bound for feature selection. There exists a gradient-based feature selection method using this bound [16].

3.2 Jaakkola-Haussler Bound

Jaakkola-Haussler have developed the following bound on the number of errors in the loo procedure for SVM without bias term b given as:

$$loo \leq \frac{1}{l} \sum_{i=1}^{l} \Psi \left(\alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_i) - 1 \right) \tag{12}$$

Note that, in [19], Lin and Zhang have proposed an estimate of the number of errors made by the loo procedure for the hard margin SVM as:

$$loo \leq \frac{1}{l} \sum_{i=1}^{l} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_i), \tag{13}$$

which can be seen as an upper bound of the Jaakkola-Haussler method since $\Phi(x - 1) \leq x$ for $x \geq 0$. In this paper Lin and Zhan method is applied for feature selection instead of Jaakkola-Haussler method.

3.3 Opper-Winther Bound

Opper-Winther bound on the number of errors in the loo procedure for SVM without bias term b is given as:

$$loo \leq \frac{1}{l} \sum_{i=1}^{l} \Psi \left(\frac{\alpha_i}{\left(\boldsymbol{K}_{SV}^{-1}\right)_{ii}} - 1 \right) \tag{14}$$

where $\boldsymbol{K}_{SV}$ is the matrix of dot-products between support vectors. In this paper we use an upper bound of the Opper-Winther bound for gene selection given as:

$$loo^{upper} \leq \frac{1}{l} \sum_{i=1}^{l} \frac{\alpha_i}{\left(\boldsymbol{K}_{SV}^{-1}\right)_{ii}}. \tag{15}$$

4. Proposed Evolutionary SVM

Evolutionary algorithms have been applied to microarray classification in order to search for the optimal or near optimal set of predictive genes on complex and large spaces of possible gene sets. Evolutionary algorithms are stochastic search and optimization techniques that have been developed over the last 30 years. A general form of the evolutionary algorithm is shown below:

```
Generate initial population, evaluate fitness
While stop condition not satisfied do
    Produced next population by
        Selection
        Recombination
    Evaluate fitness
End while
```

The evolutionary algorithm, that we propose, 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. Instead of applying crossover and mutation operations, the proposed method selects and recombines new features based on leave-one-out error bounds on SVM such as radius-margin bound, Jaakkola-Haussler bound and Opper-Winther bound and frequency of occurrence of the features in the evolutionary approach. The number of features in a predictor is parameter that we shall explore experimentally in the following section. High performance of evolutionary SVM is obtained by choosing optimum parameters of SVMs. The k-fold cross validation is used as an estimator of the generalization ability where the evolutionary SVM is applied on a k-fold cross validation set and then the generalization ability of the selected feature is tested on several different k-fold cross validation sets. 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 predictor with the highest fitness will be one that contains the best subset of genes for the classification task.

4.1 Error Bound Effect

In every generation, the right hand side of any equation in equations (11), (13) and (15) is calculated to observe the effect on error bound of each gene in each predictor. Let us denote T_m is the bound value of m genes on a predictor and T_{m-1}^i is the bound value of all genes except gene i. 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 than removing gene k. Thus genes j with small T^j_{m-1} should be deleted in the next generation.

4.2 Gene Frequency

Let us denote z^j_i be the frequency of occurrence of selected gene i at generation j. Initially all z^0_i is set to 0. At any generation j, if gene i is selected then

$$z^j_i = z^{j-1}_i + 1$$

This frequency is calculated for each predictor separately.

4.3 Gene Deletion

In every generation, we calculate the error bound value and frequency of occurrence for each gene in a predictor. We remove those genes which can reduce the error bound much and which are selected a few in the previous generations. We calculate the scoring function as

$$T_i = \epsilon T^i_{m-1} + (1-\epsilon) z^j_i / l \tag{16}$$

where $\epsilon \in [0,1]$ is a tradeoff between bound value and the frequency of occurrence in the previous generations. Gene i with the minimum T_i will be deleted from the predictor.

4.4 Fitness Function

The fitness function is the only guide to evaluate the system. There are two objectives for designing evolutionary SVM. One is to maximize the classification accuracy C_a of the k-fold cross-validation and the other is to minimize the number N_f of selected genes. If S represents the set of parameters to be evolved in the whole system, the fitness function is defined as follows:

$$\max t(S) = (1-w_f) C_a(S) - w_f N_f(S)$$

where $w_f \in [0,1]$ is a control parameter between classification accuracy and the number of selected genes.

4.5 Proposed Algorithm

The proposed 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 features (genes) $\{g_1, g_2, ..., g_m\}$ initially created randomly. Evaluate the fitness values of all predictors.
2. UNTIL termination criteria NOT satisfied DO:
3. For each predictor $G_i \in E_k$, create a new predictor G'_i
   - 3.1. Delete p genes from G_i as described in Subsection 4.3
   - 3.2. Add p genes chosen randomly to keep the size of the feature set the same, i.e., $size(G_i) = size(G'_i)$. Compute the frequency of the selected genes as described in Subsection 4.2.
   - 3.3. Compute fitness function for the new predictor G'_i.
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 the selected best predictors from the previous generation and then randomly select features from the merge-feature set to create new G'_i like cross-fold validation technique.

This procedure will be performed for a set of SVM hyperparameters and the best hyperparameters for each predictor will be obtained. From this procedure we will get n feature sets. From the n sets we will choose N_{best} top-rank features in terms of occurrence frequency. The hyperparameters for the final learning machine (SVM) will be selected by averaging the best hyperparameters of the predictors.

5. Computational Experiments

In this paper we test our proposed method using two cancer-related gene expression datasets that are described in Table 1. The dataset *Brain Tumor* is collected from http://www-genome.wi.mit.edu/cancer/pub/glioma and the dataset *Prostate Tumor* is collected from http://www-genome.wi.mit.edu/MPR/prostate. These data files contain scaled average expression value of genes from different GeneChips where the expression value of each gene is obtained by Affymetrix's GENECHIP software [1],[4]. These microarray data can be preprocessed without loosing potential information about the genes. In the preprocessing stage, these expression values are first ranged by a lower threshold θ_l and an upper threshold θ_u. That is, if the expression value is less than θ_l, it is replaced with θ_l. Similarly if the expression value is greater than θ_u, it is replaced with θ_u. After this preprocessing, the expression values are subject to a variation filter that excludes genes that has minimal variation across the samples being analyzed. The variation filter tests fold variation and absolute variation for each gene by comparing (maximum/minimum) and (maximum-minimum) of genes over the samples, and excludes genes not obeying both conditions. In this paper we prepossess the *Brain Tumor* dataset by setting $\theta_l = 20$, $\theta_u = 16000$, maximum/minimum = 3, and maximum-minimum = 100. For the *Prostate Tumor* dataset we set $\theta_l = 10$, $\theta_u = 16000$, maximum/minimum = 5, and maximum-minimum = 50 as these parameters have been used in [1],[4]. After preprocessing, the *Brain Tumor* dataset has 4434 genes and *Prostate Tumor* dataset has 5966 genes. Then these datasets are linearly scaled into the range [-1,1] and then applied to the proposed algorithm. To evaluate the performance of the proposed method we use 5-fold cross-validation on each dataset. In this paper we experiment with only Linear SVM and tested with various SVM parameter as: $C = [2^{-2}, 2^{-1}, \ldots, 2^{11}, 2^{12}]$. We compare the results of our proposed method with the that of signal-to noise score (gene ranking) method. The

Table 1: Features of microarray datasets.

Dataset	Diagnostic Task	#Samples	#Genes	#Classes	Reference
Brain Tumor	Glioblastomas and anaplastic oligodendrogliomas Types	50	12625	2	Nutt et al. (2003) [1]
Prostate Tumor	Prostate Tumor and normal tissue	102	12600	2	Singh et al. (2002) [4]

Table 2: Mean accuracy rate (%) of the proposed method with radius-margin bound and signal-to-noise ratio approach using prespecified number of selected genes.

Dataset	Proposed Method			Signal-to-noise ratio		
	Training Ac.(%)	Test Ac.(%)	#Genes	Training Ac.(%)	Test Ac.(%)	#Genes
Brain Tumor	100	100	3	100	80.72	80
				100	74.51	50
Prostate Tumor	100	100	3	100	90.19	50
				100	88.23	30

signal-to-noise score method for a binary problem calculates the ranking function: $g(i) = (\mu_{class:1}(i) - \mu_{class:2}(i))/(\sigma_{class:1}(i)+\sigma_{class:2}(i))$ for each gene i and selects the top-ranked genes according to their sorted values in descending order where $\mu_{class:j}(i)$ and $\sigma_{class:j}(i)$ are the mean and variance of gene i in class j, respectively. The signal-to-noise score method selects features (genes) and then the selected features are applied to the SVM classifier. The computational results are shown in Table 2 for radius-margin bound. In this paper, we experiment with 50 predictors and 100 populations in our algorithm, and size of the feature subset, N, is prespecified. Thus we set $w_f = 0$ in the fitness function but check with different values of N, and show the best results. The parameter in the scoring function in equation (16) is also set experimentally. From the experimental result we see that the proposed method can select a small number of genes with high accuracy. This paper shows the result using Linear SVM; however, other non-linear kernels such as Gaussian and polynomial kernels may show better result.

6. Conclusions

In this paper we propose an efficient evolutionary gene selection method based on SVM error bound theories. The SVM is used to evaluate the fitness function as a classifier. Both feature selection and hyperparameters tuning for SVM are embedded in the proposed approach. Thus the proposed method can select few informative features with high predictive accuracy by considering model uncertainty.

In a future work, we will experiment the proposed method using other existing error bounds and with different non-linear kernels and for multi-class problems.

Acknowledgement

The authors would like to thank Japan Society for the Promotion of Science (JSPS), Japan for financial support of this research. The authors also thank to Dr. K. Nishida for helpful comments and discussions.

References

[1] Nutt, C.L., Mani, D.R., Betensky, R.A., Tamayo, P., Cairncross, J.G., Ladd, C., Pohl, U., Hartmann, C., McLaughlin, M.E., Batchelor, T.T., Black, P.M., Deimling, A.V., Pomeroy, S.L., Golub, T.R.& Louis, D.N. (2003). Gene expresion-based classification of malignant gliomas correlates better with survival than histological classification. Cancer Res. 63(7), 1602-1607.

[2] Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W., & Chen, L. (2001), Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. FEBS Letter 555(2), 358-362

[3] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M.L., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S.& Golub, T.R. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. Proc. Natl. Acad. Sci. U.S.A. 98(26), 15149-15154.

[4] Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T.& Sellers, W. (2002). Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1, 203-209.

[5] Möller-Levet, C.S., West, C.M. & Miller, C.J. (2007). Exploiting sample variability to enhance multivariate analysis of microarray data. Bioinformatics 23(20), 2733-2740.

[6] Haan, J.R.de, Wehrens, R., Bauerschmidt, S., Piek, E., Schaik R.C. van & Buydens M.C (2007). Interpretation of ANOVA models for microarray data using PCA. Bioinformatics 23(2), 184-190.

[7] Yeung, K.Y. & Ruzzo, W.C. (2001). Principle component analysis for clustering gene expression data. Bioinformatics 17, 763-774.

[8] Li, L., Weinberg, C.R., Darden T.A. & Pedersen L. G. (2001). Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. 17(12), 1131-1142.

[9] Huang, H.-L. & Chang, F.-L (2007) ESVM: Evolutionary support vector machine for automatic feature selection and classification of microarray data. Bio Systems, 90, 516-528.

[10] Paul, T.K.& Iba, H. (2005). Gene selection for classification of cancers using probabilistic model building genetic algorithm. Bio Systems, 82, 208-225.

[11] Rosskopf, M., Feldkamp, U. & Banzhaf, W. (2004). Genetic programming based DNA microarray analysis for classification of tumor tissues. Tech. Report #2007-03, Dept. of Computer Science, Memorial University of Newfoundland, NL, Canada.

[12] Detting, M., & Buhlmann, P. (2003). Bosting for tumor classification with gene expression data. Bioinformatics 19(10), 1061-1-69.

[13] Li, W., Yang, Y. (2002). How many genes are needed for a discriminant microarray data analysis. In: Lin, S.M., Johnson, K.F(Eds.), Methods of Microarray Data Analysis. Kluwer Academin, 137-150.

[14] Jirapech-Umpai, T. & Aitken, S. (2005). Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. BMC Bioinformatics, 6(148)
[15] Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. Journal of Machine Learning Research, 3, 1357-1370.
[16] Chapelle, O., Vapnik, V., Bousquet, O. & Mukherjee, S. (2002), Choosing multiple parameters for support vector machines. Machine Learning, 46, 131-159.
[17] Chung, K.-M, Kao,W.-C., Sun, T., Wang L. -L & Lin, C. -J (2003). Radius margin bounds for support vector machines with Gaussian kernel. Neural Computation 15(11), 2643-2681.
[18] Jaakkola, T.S. & Haussler, D. (1999). Probabilistic kernel regression models, Proc. 1999 Conference on AI and Statistics.
[19] Wahba, G., Lin, Y. & Zhang, H. (2000). Generalized approximate cross validation for support vector machines: Another way to look at margin like quantities. A. Smola, P. Bartlett, B. Schölkopf & D. Schuurmans(Eds.). Advances in large margin classifiers (pp. 297-309). Cambridge, MA:MIT Press.
[20] Opper, M. & Winther, O. (2000). Gaussian process and SVM: Mean field and leave-one-out. A. Smola, P. Bartlett, B. Schölkopf & D. Schuurmans(Eds.). Advances in large margin classifiers (pp. 297-309). Cambridge, MA:MIT Press.
[21] Vapnik, V. (1998). Statistical Learning Theory. New York:Wiley.