

# Inductive vs Transductive Inference, Global vs Local Models: SVM, TSVM, and SVMT for Gene Expression Classification Problems

Shaoning Pang and Nikola Kasabov

Knowledge Engineering & Discover Research Institute  
Auckland University of Technology, Private Bag 92006, Auckland 1020, New Zealand  
Emails: spang@aut.ac.nz, nkasabov@aut.ac.nz

**Abstract**—This paper compares inductive-, versus transductive modeling, and also global-, versus local models with the use of SVM for gene expression classification problems. SVM are used in their three variants – inductive SVM, transductive SVM (TSVM), and SVM tree (SVMT) – the last two techniques being recently introduced by the authors. The problem of gene expression classification is used for illustration and four benchmark data sets are used to compare the different SVM methods. The TSVM outperforms the inductive SVM models applied on a small to medium variable (gene) set and a small to medium sample set, while SVMT is superior when the problem is defined with a large data set, or - a large set of variables (e.g. 7,000 genes, with little or no variable pre-selection).

**Keywords:** SVM; transductive SVM; SVM trees; gene expression classification; cancer.

## I. INDUCTIVE VERSUS TRANSDUCTIVE INFERENCES, GLOBAL VERSUS LOCAL MODELS

Most of the learning models and systems in artificial intelligence apply inductive inference where a model (a function) is derived from data and this model is further applied on new data [1]. This is the case in the area of soft computing, and particularly - in neuro-fuzzy reasoning systems [2,3,4], and in support vector machines (SVM) [5], as well as in their numerous applications. The model is created without taking into account any information about a particular new data vector. The new data would fit into the model to certain degree (an error is estimated). The model is in most cases a global model, covering the whole problem space. Creating a global model (function) that would be valid for the whole problem space is a difficult task and in most cases - it is not necessary. In some local learning systems, that include the evolving connectionist systems (ECOS) [6-8] and the SVM ensembles [10], the global model consists of many local models (rules) that collectively cover the whole space and are adjusted individually on new data. The output for a new vector is calculated based on the activation of one or several neighboring local models (rules). The inductive learning and inference approach is useful when a global model

("the big picture") of the problem is needed even in its very approximate form. Incremental, on-line learning may be applied to adjust this model on new data and trace its evolution.

Generally speaking, inductive inference is concerned with the estimation of a function (a model) based on data from the whole problem space and using this model to predict output values for a new input vector, which can be any point in this space (deduction). Most of the statistical, connectionist and fuzzy learning methods, such as: SVM [5]; MLP, RBF, and other neural network (NN) models [2,4]; ANFIS [3] and other neuro-fuzzy inference systems [4], as well as ECOS [6], that include DENFIS [8], EFuNN [7] and many more, have been developed and tested on inductive reasoning problems.

In contrast to the inductive inference, transductive inference methods estimate the value of a potential model (function) only for a single point of the space (the new data vector) utilizing additional information related to this point [5,9]. This approach seems to be more appropriate for clinical and medical applications of learning systems, where the focus is not on the model, but on the individual patient data. And it is not so important what the global error of a global model over the whole problem space is, but rather - the accuracy of prediction for any individual patient. Each individual data vector (a patient in the medical area, a target day for predicting a stock index or control of a process, or a time moment in the future for predicting a time series) may need an individual, local model developed in an ad-hoc manner, that best fits the new data, rather than - a global model used and new data tried to be matched into it without taking into account any specific information on where this new data point is located in the space.

Transductive inference is concerned with the estimation of a function in single point of the space only, regardless of its dimensionality [5,9]. For every new input vector  $x_i$  that needs to be processed for a prognostic task, the closest  $N_i$  examples that form a set  $D_i$  are derived from an existing data set  $D$  or/and generated from an existing model  $M$  (if necessary) and a new model  $M_i$  is dynamically created from these samples to approximate

the function in the locality of point  $x_i$  only. The system is then used to calculate the function value  $y_i$  for this input vector. Transductive inference methods are efficient when the size of the available data set  $D$  is relatively small (according to [5] a sample size is considered small if the ratio  $N/M < 20$ , where  $N$  is the size of the data set  $D$  and  $M$  is the VC dimension - an estimate of the possible number of functions in the space for the defined problem and for the available data set).

A simple transductive inference method is the  $k$ -nearest neighbor method ( $k$ -NN), where a new data vector  $\bar{x}_i$  is classified into one of the existing classes in the data samples from  $D$  based on the majority of classes among  $k$  nearest to the new vector samples, that form the set  $D_i$ . The distance is measured as Euclidean distance or as another type of distance. In terms of prediction systems, the output value  $y_i$  for the new vector  $\bar{x}_i$  is calculated as the average value of the output values of the  $k$ -nearest samples from the data set  $D_i$ . In a weighted  $k$ -NN method (WKNN) the output for a new vector  $\bar{x}_i$  is calculated based not only on the majority in the set  $D_i$  of the  $k$ -nearest samples, but also on their distance to  $\bar{x}_i$

$$y_i = \sum_{j=1,2,\dots,K_i} y_j \cdot w_j, \quad (1)$$

Where  $K$  is the number of nearest neighbors. Many problems in Bioinformatics, and in Molecular Biology in particular, are characterized by a small data set sparsely distributed in a large dimensional space [11]-[13] where data samples are being added continuously. This type of problems would be suitable to solve with the use of transductive inference techniques. Such problems are: promoter recognition, microarray gene expression data classification, gene expression time course data modeling, and many more.

The problem of gene expression classification is taken in this paper as a case study problem to illustrate and compare the applicability of SVM [5], transductive SVM (TSVM) [9] and SVM tree [10]. The traditional inductive SVM, and the recently introduced TSVM [9] and SVMT[10] are introduced in section II. Section III presents the gene expression classification problem and the data sets used. Experimental results are presented in section IV. Section V discusses the advantages and disadvantages of different modeling techniques for the chosen and for other problems in Bioinformatics.

## II. INDUCTIVE SVM, TSVM, AND SVMT

### A. Inductive SVM

Support vector machine was first proposed by Vapnik and his group at AT&T Bell laboratories [5]. For a typical

learning task  $P(\bar{x}, y) = P(y | \bar{x})P(\bar{x})$ , an inductive SVM learner aims to build a decision function  $f_L : \bar{x} \rightarrow \{-1, +1\}$  based on a training set  $S_{train}$ , which is

$$f_L = L(S_{train}) \quad (2)$$

Where :  $S_{train} = (\bar{x}_1, y_1), (\bar{x}_2, y_2), \dots, (\bar{x}_n, y_n)$

In SVM theory, the computation of  $f_L$  can be traced back to the classical structural risk minimization (SRM) approach, which determines the classification decision function by minimizing the empirical risk, as:

$$R = \frac{1}{l} \sum_{i=1}^N |f(\bar{x}_i) - y_i| \quad (3)$$

where  $N$  and  $f$  represent the size of examples and the classification decision function respectively,  $l$  is a constant for normalization. For SVM, the primary concern is determining an optimal separating hyper-plane that gives a low generalization error. Usually, the classification decision function in the linearly separable problem is represented by

$$f_{\bar{w},b} = \text{sign}(\bar{w} \cdot \bar{x} + b) \quad (4)$$

In SVM, this optimal separating hyperplane is determined by giving the largest margin of separation between different classes. It bisects the shortest line between the convex hulls of the two classes, which is required to satisfy the following constrained minimization, as:

$$\text{Minimize} : \frac{1}{2} \bar{w}^T \bar{w} \quad (5)$$

$$\text{Subject to} : y_i (\bar{w} \cdot \bar{x} + b) \geq 1.$$

For the linearly non-separable case, the minimization problem needs to be modified to allow misclassified data points. This modification results in a soft margin classifier that allows but penalizes errors by introducing a new set of variables  $\xi_{i=1}^l$  as the measurement of violation of the constraints.

$$\text{Minimize} : \frac{1}{2} \bar{w}^T \bar{w} + C \left( \sum_{i=1}^L \xi_i \right)^k \quad (6)$$

$$\text{Subject to} : y_i (\bar{w} \cdot \varphi(\bar{x}_i) + b) \geq 1 - \xi_i.$$

where  $C$  and  $k$  are used to weight the penalizing variables  $\xi_{i=1}^l$ , and  $\varphi(\cdot)$  is a nonlinear function which maps the input space into a higher dimensional space. Minimizing the first term in Eq.(6) corresponds to minimizing the VC-dimension of the learning machine and minimizing the second term in Eq.(6) controls the empirical risk. Therefore, in order to solve problem Eq.(6), we must construct a set of functions, and implement the classical risk minimization on the set of functions. Here, a Lagrangian method is used to solve the above problem. Then, Eq.(6) can be written as:

$$\text{Minimize} : F(\Lambda) = \Lambda \cdot 1 - \frac{1}{2} \Lambda \cdot D \cdot \Lambda, \quad (7)$$

$$\text{Subject to} : \Lambda \cdot y = 0; \Lambda \leq C; \Lambda > 0$$

where  $\Lambda = (\lambda_1, \dots, \lambda_l)$ ,  $D = y_i y_j \bar{x}_i \cdot \bar{x}_j$  for binary classification and the decision function Eq. (3) can be re-written as

$$f(x) = \text{sign}\left(\sum_{i=1}^l y_i \lambda_i^* (\bar{x} \cdot \varphi(\bar{x}_i) + b^*)\right) \quad (8)$$

### B. Transductive SVM (TSVM)

In contrast to inductive SVM learning, transductive SVM (TSVM) learning includes the information of test set  $S_{test}$  in the training procedure [9], thus the above learning function Eq.(2) of inductive SVM can be reformulated as,

$$f_L = L(S_{train}, S_{test}). \quad (9)$$

$$\text{Where} : S_{train} = (\bar{x}_1^*, y_1^*), (\bar{x}_2^*, y_2^*), \dots, (\bar{x}_n^*, y_n^*)$$

Therefore, in a linearly separable data case, to find a labeling  $y_1^*, y_2^*, \dots, y_n^*$  of the test data, the hyperplane  $\langle \bar{w}, b \rangle$  should separate both training and test data with maximum margin:

$$\begin{aligned} \text{Minimize Over } (y_1^*, y_2^*, \dots, y_n^*, \bar{w}, b) : \\ \frac{1}{2} \bar{w}^T \bar{w} \\ \text{Subject to} : \quad y_i (\bar{w} \cdot \bar{x}_i + b) \geq 1 \\ y_j^* (\bar{w} \cdot \bar{x}_j^* + b) \geq 1. \end{aligned} \quad (10)$$

To be able to handle non-separable data, similar to the way in above inductive SVM, the learning process of transductive SVM can be formulated as the following optimization problem:

*Minimize Over*

$$(y_1^*, y_2^*, \dots, y_n^*, \bar{w}, b, \xi_1, \dots, \xi_n, \xi_1^*, \dots, \xi_n^*) : \\ \frac{1}{2} \bar{w}^T \bar{w} + C \left( \sum_{i=1}^L \xi_i \right)^k + C \left( \sum_{j=1}^K \xi_j^* \right)^k \quad (11)$$

$$\begin{aligned} \text{Subject to} : \quad y_i (\bar{w} \cdot \varphi(\bar{x}_i) + b) \geq 1 - \xi_i \\ y_j^* (\bar{w} \cdot \varphi(\bar{x}_j^*) + b) \geq 1 - \xi_j^* \end{aligned}$$

Where  $C^*$  is the effect factor of the test examples, and  $C^* \xi_i^*$  is the effect term of the  $i$ th test example in the above objective function. Different methods can be used to solve Eq. (11), on of the them

### C. SVM Tree (SVMT)

SVM tree is constructed by a divide-and-conquer approach using a binary class specific clustering and SVM classification technique [18].

Basically, we perform two procedures at each node in the above tree generation. First, the class specific clustering performs a rough separation of membership because it splits the data into two disjoint subsets based on the global features such as the membership and the non-membership eigenvectors features. Next, the SVM classifier performs a 'fine' classification of membership based on training supported by the previous separation result. Algorithm 1 shows the procedure of constructing the SVM tree.

Algorithm 1: SVM Tree Training

```
Function SVMTree_Train(A training set X){
  If (X contains the same class membership) {
    Mark the end of a branch in SVM Tree T ;
    return;
  }
  Class_specific_Clustering (X; X1, X2)
  /* Partition X into subsets X1 and X2 by Eq. 12 */
  T=Append_node(T);
  /*Train SVM to follow above data partitioning */
  Train_SVM_Classifier(X; X1, X2)
  /* Recursive partitioning and training*/
  SVMTree_Train(X1);
  SVMTree_Train(X2);
}
```

After constructing the SVM tree, we can predict its membership of a test input data  $x$  as follows. First, we decide in which cluster the test input data belongs by executing the  $SVM\_Test\_T_{current}(x)$  at the root node in SVM tree. Depending on the result of the decision made by the root node, we will go down to one of the children nodes. This procedure is repeated until a terminal node is reached. We assign the membership or non-membership label to the test input face  $x$  depending on the label of terminal node's membership. Algorithm 2 illustrates the testing procedure of the SVM tree.

Algorithm 2: SVM Tree Testing

```
Function SVMTree_Test(SVM tree T, Input data x){
  current = 1; /* Set current node to the root node of T */
  /* Search SVM tree until the terminal node is reached*/
  while (Tcurrent is an internal node) {
    SVM_Test_Tcurrent(x);
    next = Find_next_node(T, current);
    current = next;
  }
  /* Return the label of the terminal node*/
  return Label(Tcurrent);
}
```

Consider a binary classification problem, the training set can be divided into the positive and the negative subsets as  $D_{train} = M_+ \cup M_-$ . Applying the above PCA technique to  $M_+$  and  $M_-$  respectively, we obtain two representative eigenvector sets such as the positive-class eigenvectors  $\mathbf{U}_+ = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K]$  and the negative-class eigenvectors  $\mathbf{U}_- = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L]$ . They characterize the negative class and positive class respectively.

For the binary membership-based clustering, we need to define a  $N$  dimensional binary partition vector  $\mathbf{V}$  for two partitioned subgroups, where each element  $v_i$  ( $1 \leq i \leq N$ ) is 1 when the  $i$ th data point  $\mathbf{x}_i$  belongs to cluster 1, or 0 when the  $i$ th data point  $\mathbf{x}_i$  belongs to cluster 2. Under the given two eigenvectors  $\mathbf{U}_+$  and  $\mathbf{U}_-$ , the element  $v_i$  of the partition vector can be determined as follows.

$$v_i = \begin{cases} 1 & \text{if } \sum_{k=1}^K x_i \cdot \mathbf{u}_k \leq \sum_{k=1}^K x_i \cdot \bar{\mathbf{u}}_k \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where  $\sum_{k=1}^K x_i \cdot \mathbf{u}_k$  is the distance of the data point  $\mathbf{x}_i$  projected onto the positive class eigenvectors ( $\mathbf{U}_+$ ), and  $\sum_{k=1}^K x_i \cdot \bar{\mathbf{u}}_k$  is the distance of the data point  $\mathbf{x}_i$  projected onto the negative eigenvectors ( $\mathbf{U}_-$ ).

### III. CASE STUDY: GENE EXPRESSION CLASSIFICATION PROBLEMS

#### A. General Introduction to the Problem

In the area of bioinformatics, the identification of gene subsets responsible for classifying available samples to two or more classes (such as ‘malignant’ or ‘benign’) is an important task. Most of current classifiers are sensitive to disease-marker genes selection. Here we use SVM, TSVM and SVM on different tasks of the same problem. While the SVM creates a global model and SVMT creates a local model for each sample, the SVMT creates a global model and performs classification in many local subspaces instead in the whole data space as typical classifiers do.

#### B. Data Sets

We use four different cancer data sets: Lymphoma [11], Leukaemia [12], Colon [13], leukemia cell line time series data [14-16]. The flexibility in using different modifications of SVM is demonstrated in section 4.

The lymphoma data set is a collection of gene expression measurements from 77 malignant lymphocyte samples reported by Shipp et al. [11] (Shipp, Ross et al. 2002). It contains 58 samples of diffused large B-Cell lymphoma (DLBCL) and 19 samples of Follicular lymphoma (FL), where DLBCL samples are divided into two groups: those with cured disease ( $n = 32$ ) and those with fatal or refractory disease ( $n = 26$ ). The lymphoma

data containing 6,817 genes is available at <http://www.genome.wi.mit.edu/MPR/Lymphoma>.

The Leukaemia data is a collection of gene expression measurements from 72 Leukaemia (composed of 62 bone marrow and 10 peripheral blood) samples reported by Golub et al. [12]. It contains an initial training set composed of 27 samples of acute lymphoblastic Leukaemia (ALL) and 11 samples of acute myeloblastic Leukaemia (AML), and an independent test set composed of 20 ALL and 14 AML samples. The gene expression measurements were taken from high-density oligonucleotide micro-arrays containing 7,129 probes for 6,817 human genes. This data is available at <http://www.genome.wi.mit.edu/MPR>.

The second Leukaemia data is a collection of gene expression observation of two cell lines U937 (MINUS – a cancer cell line that is positively affected by retinoic acid and becomes normal cell after a time interval of 48 hours, and PLUS cell line – that is cancerous and not affected by the drug [14,15]). Each of the two time series contains the expression value of 12,000 genes at four time points: CTRL, 6hours, 24hours and 48hours. We can view this problem also as a classification problem where we have 4 variables (the time points) and 24,000 examples (the gene expression of a gene over the 4 time points) classified in two classes – MINUS and PLUS. The data was collected at NCI- Frederick, USA [14,15].

The Colon data set is a collection of 62 expression measurements from Colon biopsy samples reported by Alon et al. [13]. It contains 22 normal and 40 Colon cancer samples. The colon data having 2,000 genes is available at <http://microarray.princeton.edu/oncology>

#### C. Experimental Steps

On the above gene expression cancer data sets, we applied the following methodology:

Step 1. Define target classes.

Step 2. Identify a gene subset (variable selection): We employed the multi-objective GA (NSGA-II) [17], where three objective functions are used. The first objective is to minimize the size of gene subset in the classifier. The second objective is to minimize the number of mismatches in the training data samples calculated using leave-one-out cross-validation procedure. The third objective is to minimize the number of mismatches in the test samples.

Step 3 Filter and normalize data: We eliminate genes with not much variation in the expression values for the two classes to ensure a differentiation of the classes. We normalize data by evaluating the difference of the maximum and minimum gene expression values for every gene, and by measuring its stand deviation.



TABLE 1. RESULTS OF APPLYING SVM, TSVM, AND SVMT ON FOUR GENE EXPRESSION CLASSIFICATION PROBLEMS

	With/Without marker gene Selection	K-NN	SVM	TSVM	SVM Tree
Lymphoma [11]	DLBCL vs FL; 6432genes, 77samples	53.2%	55.8%	57.1%	<b>77.9%</b>
	DLBCL vs. FL; 30genes, 77samples	90.9%	92.2%	<b>93.5%</b>	92.2%
	cured vs fatal 6432genes,58 samples	51.7%	53.3%	55.1%	<b>72.4%</b>
	cured vs fatal; 13genes, 58 samples	70.7%	72.4%	<b>79.3%</b>	70.7%
Leukaemia [12]	ALL versus AML; 7219genes, 72samples	52.8%	55.6%	52.8%	<b>78.3%</b>
	ALL versus AML; 3859genes, 72samples	93.1%	94.4%	95.8%	<b>100%</b>
	ALL versus AML; 27genes, 72samples	91.6%	98.6%	<b>98.6%</b>	90.3%
L. cell line [14-16]	Min vs Plus;4 variables; 24,000 samples	52.5%	53.3%	50.0%	<b>81.3%</b>
Colon Cancer [13]	Normal Vs. Cancer; 2000genes 62samples	75.8%	79.0%	72.6%	<b>80.7%</b>
	Normal Vs. Cancer; 12genes, 62samples	98.4%	<b>100%</b>	<b>100%</b>	98.4%

Step 4. Build a classifier: For each variable set and defined classes we build and test classifiers in a cross-validation mode (leave-one-out) by removing one sample and then using the rest as a training set. Several models are built using different numbers of marker genes and the final chosen model is the one that minimizes the total cross validation error.

Step 5. Evaluate results: We evaluate prediction results and compute confusion matrices. For the purpose of comparison with the past studies, we compare the proposed classifier algorithm with the K-NN model and an inductive, global SVM.

#### IV. EXPERIMENTAL RESULTS OF APPLYING INDUCTIVE SVM, TSVM AND SVMT ON THE CASE STUDY PROBLEMS

Experimental results are presented in Table 1. Fig. 1,2 and 3 show the created SVMTs for the data sets used. Each internal node of the tree identifies an SVM classifier, which is represented as an ellipse with a number as its identity.

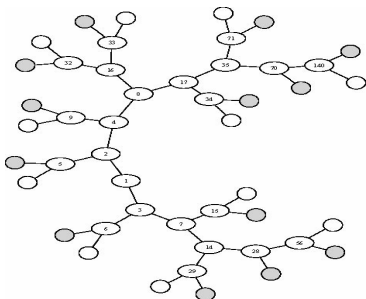


Fig. 1. SVMT For The Classification Of DLBCL Vs. FL

When the parent node is labeled as  $i$ , its two children

nodes are identified as  $2i$  and  $2i+1$ , respectively. We also represent the terminal node as a circle or a filled circle, which denotes positive or negative class, respectively.

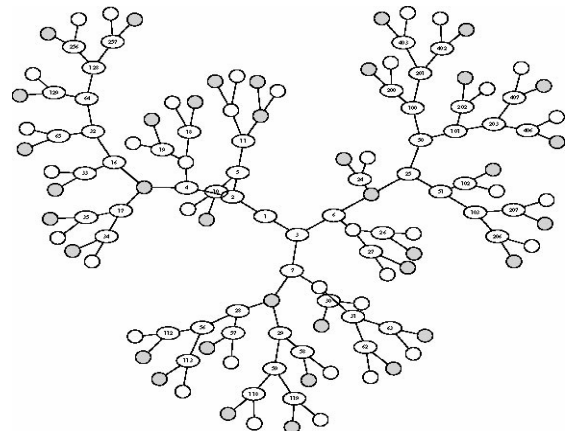


Fig. 2. SVMT for The Classification Of MINUS Vs. PLUS Leukemia.

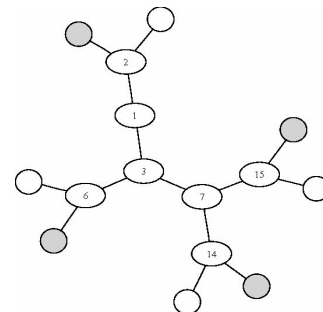


Fig. 3. SVMT For The Classification Of Cured (Positive Class) vs. Fatal (Negative Class) Lymphoma Cases

#### V. DISCUSSIONS AND CONCLUSION

From the results in Table 1 we can compare inductive SVM, transductive SVM (TSVM), and the SVM tree (SVMT) on the case study data sets above. The TSVM performs at least as good as the inductive SVM on a small or a medium variable set (several genes or several hundred genes). A TSVM model can be generated on a smaller number of variables (genes) evaluated on the selected small data set from a local problem space for a particular new sample (e.g. a new patient's record). The TSVM allows for an individual model generation and therefore is promising as a technique for personal medicine.

The SVMT performs best on a large variable space (e.g. thousand of genes, sometimes with a little or no pre-processing and no pre-gene selection). This feature of the SVMT allows for a microarray data collection from a tissue sample and an immediate analysis without the analysis being biased by a gene pre-selection.

As a future direction, we are working on on-line adaptive SVM classification systems where we assume that datasets are not always available in advance. They are usually provided as a data stream, so that we cannot have the information of unlabelled data and new types of SVM techniques will be needed.

#### ACKNOWLEDGEMENT

The research is fully supported by the Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, New Zealand. The time series gene expression Leukaemia data has been collected at the National Cancer Institute NCI at Frederick, USA.

#### REFERENCES

- [1] M.T. Mitchell, *Machine Learning*, MacGraw-Hill, 1997.
- [2] G. Carpenter and S. Grossberg, *Pattern recognition by self-organizing neural networks*, Massachusetts: MIT Press, Cambridge, 1991.
- [3] R. Jang, "ANFIS: adaptive network-based fuzzy inference system," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 23, pp. 665-685, 1993.
- [4] N. Kasabov, *Foundations of neural networks, fuzzy systems and knowledge engineering*, MIT Press, 1996.
- [5] V. Vapnik, *Statistical Learning Theory*, John Wiley&Sons, Inc, 1998.
- [6] N. Kasabov, *Evolving connectionist systems: Methods and Applications in Bioinformatics, Brain study and intelligent machines*, Springer Verlag, London, New York, Heidelberg, 2002.
- [7] N. Kasabov, "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning," *IEEE Trans. SMC – part B, Cybernetics*, vol.31, No.6, pp. 902-918, December 2001.
- [8] N. Kasabov and Q. Song, "DENFIS: Dynamic, evolving neural-fuzzy inference systems and its application for time-series prediction," *IEEE Trans. on Fuzzy Systems*, vol.10, No.2, pp. 144-154, April 2002.
- [9] N. Kasabov and S. N. Pang, "Transductive Support Vector Machines and Applications in Bioinformatics for Promoter Recognition," *Proc. of International Conference on Neural Network & Signal Processing*, Nangjing, China, 2003, IEEE Press.
- [10] S. N. Pang, Dajin Kim, and S. Y. Bang, "Membership authentication in the dynamic group by face classification using SVM ensemble," *Pattern Recognition Letters*, Vol. 24, pp. 215-225, 2003
- [11] M. A. Shipp, K. N. Ross, et al. (2002). "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning." *Nature Medicine*, Vol.8, No.1, pp. 68-74, 2002.
- [12] T. R. Golub, et. al. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, Vol. 286, pp. 531-7, 1999.
- [13] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D Mack, and A. J. Levine, "Broad pattern of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *In Proceeding of National Academy of Science, Cell Biology*, Vol. 96, pp. 6745-6750, 1999.
- [14] D. S. Dimitrov,, "Identification and characterization of rapidly dividing U937 clones with differential telomerase activity and gene expression profiles: role of c-Myc/Mad1 and Id/Ets proteins," *Leukemia*, Vol. 16, pp. 1877-1880, 2002.
- [15] X. Xiao, S. Phogat, I. A. Sidorov, & J. Yang, " Identification and Characterization of Rapidly Dividing U937 Clones with Differential Telomerase Activity and Gene Expression Profiles: Role of c-MYC/MAD1 and ID/ETS Proteins," *Leukemia*, Vol. 16, No. 9, pp. 1877-1880, 2002.
- [16] N. Kasabov., & Dimitrov, D. S. "A Method for Modelling Genetic Regulatory Networks by using Evolving connectionist Systems and Microarray Gene Expression Data," *Proc. of 9th International Conference on Neural Information Processing*, pp. 596-601, IEEE Press, 2002.
- [17] K. Deb, S. Agrawal, A. Pratap, and T. Meyarvian, "A fast and elitist multi-objective genetic algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation*, Vol.6, No. 2 pp. 182-197, 2002.
- [18] S. N. Pang, "Constructing SVM Multiple Tree for Face Membership Authentication", *Proc. Of International Conference on Biometric Authentication, Hong Kong, LNCS, Springer, 2004.*