IEEE Int. Conf. Neural Networks & Signal Processing
Nanjing, China, December 14-17, 2003

TRANSDUCTIVE SUPPORT VECTOR MACHINES AND APPLICATIONS IN BIOINFOMATICS FOR PROMOTER RECOGNITION

Nikola Kasabov, Shaoning Pang

Knowledge Engineering & Discover Research Institute
Auckland University of Technology, Private Bag 92006, Auckland 1020, New Zealand

ABSTRACT

This paper introduces a novel Transductive Support Vector Machine (TSVM) model and compares it with the traditional inductive SVM on a key problem in Bioinformatics - promoter recognition. While inductive reasoning is concerned with the development of a model (a function) to approximate data from the whole problem space (induction), and consecutively using this model to predict output values for a new input vector (deduction), in the transductive inference systems a model is developed for every new input vector based on some closest to the new vector data from an existing database and this model is used to predict only the output for this vector. The TSVM outperforms by far the inductive SVM models applied on the same problems. Analysis is given on the advantages and disadvantages of the TSVM. Hybrid TSVM-evolving connections systems are discussed as directions for future research.

1. INDUCTIVE & TRANSDUCTIVE INFERENCES

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, [2] [3] [4-7], and particularly - in neuro-fuzzy reasoning systems [8, 9] [10], and in support vector machines (SVM) [11]and in their numerous applications (see for example [12]). 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 (see for example [13] [14]) that include the evolving connectionist systems (ECOS) [15] 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, when incremental, on-line learning is 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) - Fig.1. Most of the statistical, connectionist and fuzzy learning methods, such as SVM, MLP; RBF; ANFIS (see [10]; [16]) and ECOS [15] that include DENFIS [17], EFuNN [18, 19] and many more, have been developed and tested on inductive reasoning problems.

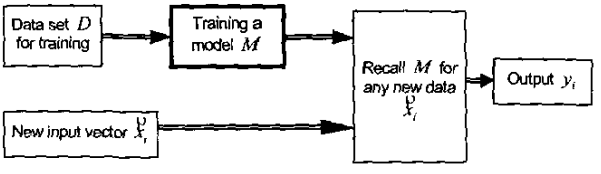

Fig. 1 A block diagram of an inductive reasoning system. A global model M is created based on data samples from D and then recalled for a new vector.

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 [11]. 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 [20]. Each individual data vector (a patient in the medical area [21, 22], a target day for predicting a stock index or control of a process [23],or a time moment in the future for predicting a time series [24, 25]) may need an individual, local model developed in an ad-hoc manner,

0-7803-7702-8/03/ $17.00 ©2003 IEEE

that best fits the new data, rather then - 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. For every new input vector *xi* 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 - Fig. 2. The system is then used to calculate the function value *yi* for this input vector. Fig. 2 Transductive inference methods are efficient when the size of the available data set D is relatively small (according to [11] 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).

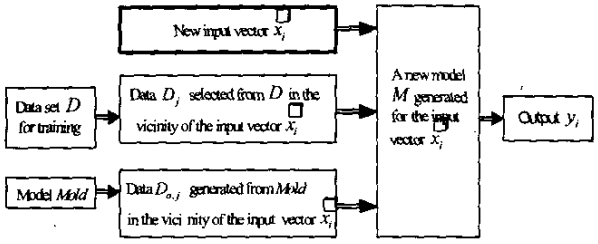

Fig.2 A block diagram of a transductive inference system. An individual model M_i is trained for a new input vector $\vec{x}_i$ with data samples D_i selected from a data set D, and data samples $D_{o,i}$ generated from an existing model.

A simple transductive inference method is the k-nearest neighbor method (k-NN), where a new data vector $\vec{x}_i$ 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 $\vec{x}_i$ 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 $\vec{x}_i$ is calculated based not on the majority in the set D_i of the k-nearest samples, but also on their distance to $\vec{x}_i$

$$y_i = \sum_{j=1,2,...,N_i} y_j \cdot w_j \qquad (1)$$

Many problems in Bioinformatics, and in Molecular Biology in particular, are characterized by a small data set sparsely distributed in a large dimensional space [12] 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 modelling, and many more. The problem of promoter recognition is taken in this paper as a case study problem. The traditional inductive SVM (section 2) are compared with the novel transductive SVM introduced in section 3. Section 4 presents some experimental results that demonstrate the superiority of the TSVM for the class of problems versus the inductive SVM, while section 5 discusses further development of the transductive SVM and hybrid systems for bioinformatics applications.

2. INDUCTIVE SVM

Support vector machine is first proposed by Vapnik and his group at AT\&T Bell laboratories [26],[27]. For a typical learning task $P(\vec{x}, y) = P(y \mid \vec{x})P(\vec{x})$, an inductive SVM learner aims to build a decision function $f_L : \vec{x} \rightarrow \{-1,+1\}$ based on a training set S_{train}, which is

$$f_L = L(S_{train})$$
$$Where: S_{train} = (\vec{x}_1, y_1), (\vec{x}_2, y_2), ..., (\vec{x}_n, y_n) \qquad (2)$$

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

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

where N and f represent the size of examples and the classification decision function, respectively. 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

$$f_{\vec{w},b} = sign(\vec{w} \cdot \vec{x} + b) \qquad (4)$$

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

$$Minimize: \frac{1}{2}\vec{w}^T\vec{w}$$
$$Subject\ to: y_i(\vec{w} \cdot \vec{x} + b) \geq 1. \qquad (5)$$

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.

$$\begin{aligned} &Minimize: \frac{1}{2}\vec{w}^T\vec{w} + C(\sum_{i=1}^{L}\xi_i)^k \\ &Subject\ to: y_i(\vec{w}\cdot\varphi(\vec{x}_i)+b) \geq 1-\xi_i. \end{aligned} \tag{6}$$

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

$$\begin{aligned} &Minimize: F(\Lambda) = \Lambda\cdot 1 - \frac{1}{2}\Lambda\cdot D\cdot\Lambda, \\ &Subject\ to: \Lambda\cdot y = 0; \Lambda \leq C; \Lambda > 0 \end{aligned} \tag{7}$$

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

$$f(x) = sign(\sum_{i=1}^{l} y_i\lambda_i^*(\vec{x}\cdot\varphi(\vec{x}_i)+b^*)) \tag{8}$$

3. INDUCTIVE SVM

In contrast to above introduced inductive SVM learning, transductive SVM learning specially includes the knowledge of test set S_{test} in training procedure [28], thus the above learning function Eq.(2) of inductive SVM can be reformulated as,

$$\begin{aligned} &f_L = L(S_{train}, S_{test}). \\ &Where: S_{train} = (\vec{x}_1, y_1^*),(\vec{x}_2, y_2^*),...,(\vec{x}_n, y_n^*) \end{aligned} \tag{9}$$

Therefore, in a linearly separable data case, to find a labeling $y_1^*, y_2^*,\Lambda\ , y_n^*$ of the test data, the hyperplane $<\vec{w}, b>$ should separate both training and test data with maximum margin.

$$\begin{aligned} &Minimize\ \text{Over}\ (y_1^*, y_2^*,\Lambda\ , y_3^*, \vec{w}, b): \\ &\qquad \frac{1}{2}\vec{w}^T\vec{w} \\ &Subject\ to: \quad y_i(\vec{w}\cdot\vec{x}_i+b) \geq 1 \\ &\qquad y_j^*(\vec{w}\cdot\vec{x}_j+b) \geq 1. \end{aligned} \tag{10}$$

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,

$$\begin{aligned} &Minimize\ \text{Over} \\ &(y_1^*, y_2^*,\Lambda\ , y_3^*, \vec{w}, b, \xi_1,\Lambda\ ,\xi_n,\xi_1^*,\Lambda\ \xi_k^*): \\ &\qquad \frac{1}{2}\vec{w}^T\vec{w} + C(\sum_{i=1}^{L}\xi_i)^k + C(\sum_{j=}^{K}\xi_j^*)^k \\ &Subject\ to: \quad y_i(\vec{w}\cdot\varphi(\vec{x}_i)+b) \geq 1-\xi_i \\ &\qquad y_j^*(\vec{w}\cdot\varphi(\vec{x}_j)+b) \geq 1-\xi_j^* \end{aligned} \tag{11}$$

Where C^* is the effect factor of the query examples, and $C^*\xi_i^*$ is the effect term of *ith* query example in above objective function. To solve this optimization equation, algorithms can be referenced from [28],[29].

4. CASE STUDY: PROMOTER RECOGNITION

4.1. Promoter Recognition

Only 2-5% of the human genome (the DNA) contains useful information what concerns the production of proteins. The number of genes contained in the human genome is about 40,000. Only the gene segments are transcribed into RNA sequences and then translated into proteins. The transcription is achieved through special proteins, enzymes called RNA polymerase, that bind to certain parts of the DNA (promoter regions) and start 'reading' and storing in a mRNA sequence each gene code.

Analysis of a DNA sequence and identifying promoter regions is a difficult task [30]. If it is achieved, it may make possible to predict, from a DNA information, how this organism will develop, or alternatively - what an organism looked like in retrospect [15]. The promoter recognition process is part of a complex process of gene regulatory network activity, where genes interact between each other over time, defining the destiny of the whole cell [31].

4.2. Data Sets

TSVM and SVM are trained in a supervised manner on a collection of promoter and non-promoter sequences. The promoter sequences are obtained from the Eukaryotic Promoter Database (EPD) http://www.epd.isb-sib.ch/. We have used 793 different vertebrate promoter sequences of length 250 bps contained in EPD Rel.65 and covering the region of 200 bps upstream of the transcription start site (TSS) and 49 bps downstream of the TSS. These 250 bps long sequences represent positive training data. We also collected a set of non-overlapping human exon and intron sequences of length 250 bps each, from the Genebank, http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html, Rel. 121. In total we used 800 exon and 4000 intron sequences. The training of the system is made on this set.

Since heuristic approaches have been proven to be detect promoters with a very high level of specificity [38], it is highly preferable to make as few heuristically decisions as possible, relying on an optimization process to find the best solution. In order to create a feature vector from promoter elements, we carried out an initial study on promoter vocabulary for feature encoding.

4.3. Promoter Feature Encoding

Our feature encoding is based on a promoter vocabulary in the meaning of promoter language. But our understanding of promoter vocabulary is very modest compared to our understanding of the vocabulary of human being language. This is mainly due to basic patterns of promoter encoding have not yet been identified; a standardized set of features for addressing the characteristics of promoter does not exit; nor are there rules defining how features are to be combined.

Previous investigations on promoter recognition concentrated mainly on promoter encoding by a set of Motifs [32],[33]. Particularly interesting is the work of Matthias Scherf etc. [32]. Their research focused on classifying sequence in terms of two disjunct sets of IUPAC groups: a set of promoter-related IUPAC groups define the class ``promoter", while a set of non-promoter-related IUPAC groups defines the class ``non-promoter". Similarly, we encode the promoter feature by judging the similarity between the query sequence and the basic promoter Motifs - promoter vocabulary, which is defined and optimized in two steps by K-NN classifier.

In the first step, using Motif search engine provided by Genome Net http://motif.genome.ad.jp/, we extract a set of promotor IUPAC group and a set of non-promotor IUPAC group by conducting DNA motif searching on promoter training set and non-promoter training set (including exon and intron), respectively. To select the motifs with the most important characteristic of promoter, we set the searching cut score as 98. Next, we use a set of promoter IUPAC that is not contained in the non-promoter IUPAC as an initial promoter vocabulary.

In the second step, a K-NN classifier is employed as a representative of Bayesian classifiers to judge how a promoter vocabulary response to bayesian classifiers on promoter recognition. Results are evaluated by three-folder cross-validation. Then, the set of promoter vocabulary is finally determined after a recursive selection procedure of one-by-one adding and removing examples according to the classification output of K-NN classifier.

Based on the selected promoter vocabulary $\{h_i\}_{i=1}^{Q}$, for one DNA sequence x of length L, its similarity reflexed on the ith word of vocabulary H can be computed as,

$$S_i = \sum_{j=1}^{L/(|h_i|+\delta)} |h_i| - LCS(h_i, x_j, \xi) \qquad (12)$$

Where LCS denotes the computation of longest common sequence [1], δ is matching interval of sequence alignment, and ξ is gap penalty.

In the third step, to reduce the classification difficulty, an ensemble of SVMs on promoter versus intron and SVMs on promoter versus exon is modeled by using the strategy of majority voting [34]. Fig. 3 is the structure of SVM ensemble. Due to the larger dataset of intron, the number of SVMs in ensemble on promoter versus intron is greater than the number of SVMs on promoter versus exon. TSVM and ISVM are tested in turn for comparison.

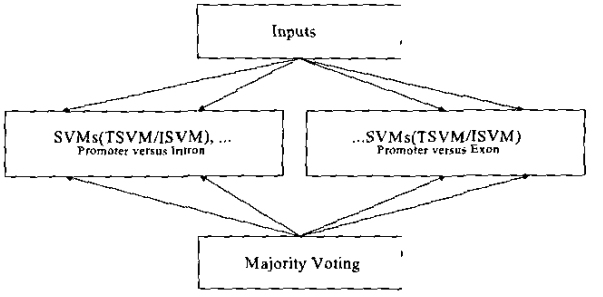

Fig. 3 The structure of SVM ensemble for promoter recognition

4.4. Comparison Results of TSVM versus ISVM

To evaluate the results using the approach of 3-fold cross-validation, we used 80\% of sequences in each class for training, while the left 20\% was kept for evaluation. We divided the training set into three disjunctive sets. From these sets, three different training sets are built by

Method	*TP*	*TP of total matches (%)*	*FP*	*TP/FP*	*Accuracy (%)*
ISVM: Ex1(que:1,reg:2,3)	81	64.8	298	0.27	51
Ex2(que:2,reg:1,3)	69	65.2	273	0.25	43
Ex3(que:3,reg:1,2)	92	63.3	317	0.29	58
Average	81	64.4	296	0.27	51
TSVM:Ex1(que:1,reg:2,3)	126	85.9	98	1.28	79
Ex2(que:2,reg:1,3)	132	83.9	126	1.04	83
Ex3(que:3,reg:1,2)	119	84.2	110	1.08	75
Average	126	84.6	111	1.13	79

Table. 1 Experimental results of promoter recognition using TSVM and ISVM

joining two of the sets in turn, while the third set was used as a test set.

For the convenience of results comparison, we follow the evaluation schema of Fickett [35] to consider the true positive (TP) and false positive (FP) of system. Table. 1 shows the comparison result of TSVM versus ISVM for promoter recognition. As we can see, the TSVM leads to an improved performance on promoter recognition, raising coverage from 50% for ISVM to 79%, and TP/FP from 0.27 to 1.13 as well. It indicates that transductive inference performs much better than inductive inference because it makes use of the information about the distribution of unlabelled data.

5. CONCLUSION

We compared transductive SVM and inductive SVM on promoter recognition. During this procedure, we first collected promoter motifs by performing motif searching on both promoter dataset and non-promoter dataset, and from which we select promoter motifs with strongest response to bayesian classifier as promoter words to make up of the promoter vocabulary. Next, we used this vocabulary as a codebook/dictionary, and extract promoter features for SVM classification by performing a LCS searching in this codebook. We demonstrated that TSVM performs better than ISVM on a specific promoter recognition task with a ready both training and test dataset, which indicates the special usage of trunsductive learning compared to inductive learning. However, datasets in practice are not always available in advance. They are usually provided as a data stream. It follows that we cant have the information of unlabelled data included in our model training. Thus it will be difficult to use Transductive models like TSVM as a general online classifier. To deal with this limitation, we think, evolving systems like Hybrid evolving TSVM can be a direction in our futures work.

6. 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] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent & Fuzzy System*, Vol. 2, 1994.

[4] G. Cybenko, "Approximation by super-positions of sigmoidal function," *Mathematics of Control, Signals and Systems*, Vol. 2, pp.303-314, 1989.

[5] T.M. Heskes and B. Kappen, "On-line learning processes in artificial neural networks," *Math. foundations of neural networks*, pp. 199-233, 1993.

[6] T. Kohonen, *Self-Organizing Maps*, Second ed: Springer, Verlag, 1997.

[7] G.A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," in Cambridge University Engineering Department, 1994.

[8] C. Bishop, *Neural networks for pattern recognition*, Oxford University Press, 1995.

[9] C.T. Lin and C.S.G. Lee, *Neuro Fuzzy Systems*, Prentice hall, 1996.

[10] N. Kasabov, *Foundations of neural networks, fuzzy systems and knowledge engineering*, MIT Press, 1996.

[11] V. Vapnik, *Statistical Learning Theory*, John Wiley&Sons, Inc, 1998.

[12]} P. Baldi and S. Brunak, Bioinformatics - A Machine Learning Approach, 2001.

[13] D. Saad, *On-line learning in neural networks*, Cambridge University Press, 1999.

[14] B. Fritzke, "A growing neural gas network learns topologies," *Advances in Neural Information Processing*

Systems, Vol. 7, pp. 625-632, 1995.

[15] N. Kasabov, *Evolving connectionist systems - methods and applications in bioinformatics, brain study and intelligent machines*, London-New York: Springer Verlag, 2002.

[16] R. Jang, "ANFIS: adaptive network-based fuzzy inference system," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 23, pp. 665-685, 1993.

[17] 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, pp. 144-154, 2002.

[18] N. Kasabov, "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning," *IEEE Trans. SMC - part B, Cybernetics*, Vol. 31, pp. 902-918, 2001.

[19] N. Kasabov, "Evolving Fuzzy Neural Networks for On-line, Adaptive, Knowledge-based Learning," *IEEE Transactions of Systems, Man, and Cybernetics B-Cybernetics*, Vol. 13, 2001.

[20] J. M. Jones, A. D. Redmond, and J. Templeton, "Uses and Abuses of Statistical Models for Evaluating Trauma Care," *The Journal of Trauma: Injury, Infection, and Critical Care*, Vol. 38, pp. 89-93, 1995.

[21] A. I. Akl, M. A. Sobh, Y. M. Enab, and J. Tattersall, "Artificial Intelligence: A New Approach for Prescription and Monitoring of Hemodialysis Therapy," *Americal Journal of Kidney Diseases*, Vol. 38, pp. 1277-1283, 2001.

[22] K. M. Anderson, P. M. Odell, P. W. F. Wilson, and W. B. Kannel, "Cardiovascular disease risk profiles," *Americal Heart Journal*, Vol. 121, pp. 293-298, 1991.

[23] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*, New Jersey: Prentice Hall Englewood Cliffs, 1984.

[24] J. D. Farmer and J. J. Sidorowitch, "Predicting chaotic time series," *Physical Review Letters*, Vol. 59, pp. 845-848, 1987.

[25] G. E. P. Box and G. M. Jenkins, *Time series analysis, forecasting and control*, San Francisco: Holden Day, 1970.

[26] V. Vapnik, Estimation of dependences based on empirical data, Springer-Verlag, 1982.

[27] C. Cortes and V. Vapnik., "Support vector network," *Machine learning*, Vol. 20, pp. 273-297, 1995.

[28] T. Joachims, "Transductive Inference for Text Classification using Support vecter Machines," *International conference on Machine Learning (ICML)*, pp. 200-209, 1999.

[29] Y. Chen, G. Wang, and S. Dong, "Learning with Progressive Transductive Support Vector Machine," *IEEE International Conference on Data Mining (ICDM'02*, Maebashi city, Japan, 2002.

[30] T. Werner, "Models for prediction and recognition of eukaryotic promoters," *Mammalian Genome*, Vol. 10, pp. 168-175.

[31] N. Kasabov and D. Dimitrov, "A method for gene regulatory network modelling with the use of evolving connectionist systems,", *International Conference on Neuro-Information Processing*, Singapore, 2002.

[32] M. Scherf, A. Klingenhoff, and T. Werner, "Highly Specific Localization of Promoter Regions in Large Genomic Sequences by PromoterInSpector: A Novel Context Analysis Approach," *J. Mol. Biol.* Vol. 297, pp. 599-606, 2000.

[33] S. Knudsen, "Promoter 2.0: for the Recognition of PolII Promoter Sequences," *Bioinformatics*, Vol. 15, pp. 356-361, 1999.

[34] 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.

[35] J. Fickett and A. G., Hatzigeorgiou, "Eukaryotic promoter recognition," *Genome Res.* Vol. 7, pp. 861-878, 1997.