Gene Expression Based Classification using Iterative Transductive Support Vector Machine

Hossein Tajari and Hamid Beigy

***Abstract*— Support Vector Machine (SVM) is a powerful and flexible learning machine. In recent years combination of SVM and Transductive learning has attracted more and more attention. In many applications such as gene expression, unlabeled data is abundant and available. However labeling data is much more difficult and expensive. Dealing with gene expression datasets, challenges such as curse of dimensionality and insufficient labeled data is inevitable. This paper introduces Iterative Transductive Support Vector Machine (ITSVM). This method which constructs a hyperplane using both training set and working set approximates the optimal solution. Applying proposed algorithm on gene expression datasets show that the proposed method can exploit unlabeled data distribution. In many cases our method improved the accuracy compared to related methods.**

***Index Terms*—Transductive learning; gene expression; support vector machine; cancer.**

I. Introduction

Reliable and successful classification is essential for diagnosing patients for further treatment. The DNA micro-array technology has brought to data-analysts extensive patterns of gene expression simultaneously recorded in a single experiment [1]. cDNA microarray and high density oligonucleotide chips are novel biotechnologies increasingly used in gene expression research [2]. Monitoring gene expression levels in cells may lead to better understanding of the molecular variations among tumor. Gene expression datasets has many challenges such as large number of features (usually thousands of features), relatively insufficient number of labeled data, noise and etc. the noise in datasets arises from irrelevant features and error in dataset preparation phase. In this situation we can easily find a linear classifier to separate the training data but it will perform poorly on working sets. In other words we deal with sparse points in the feature space.

It is crucial that the learner be able to generalize well using little training data [3]. So, we seek overall risk minimization to minimize empirical error on both training and working set. The transduction problem is to estimate the value of classification function at the given points in the working set. This contrasts with the standard inductive learning problem of estimating the classification method at all possible values and then using the fixed function to deduce the classes of the working dataset [4].

In this paper, we introduce a new Transductive learning algorithm based on SVM called Iterative Transductive Support Vector Machine (ITSVM). We use the ITSVM to solve transductive problem using overall risk minimization (ORM) posed by Vapnik. ITSVM tries to approximate optimal solution by iterative procedure. Many feature reduction techniques exist that can be useful for extracting proper features and discard others. Discarding features may cause loss of important information and needs careful observation. The proposed method, which constructs a hyperplane using both training set and working set, approximates the optimal solution. The experimental results show that the proposed method can exploit the distribution of unlabeled data on the used gene expression datasets. The experimental results show that in many cases the proposed method improves the accuracy compared to related methods.

The rest of the paper is organized as follows: In section II we review some related work on transductive learning. Then, we introduce the Iterative Transductive Support Vector Machine (ITSVM) in section III. Details about gene selection and experimental results discussed in section IV and the conclusion of the paper is given in section V.

Manuscript received December 6, 2011; revised January 11, 2012.

Hossein Tajari and are with Computer Engineering Department, Sharif University of Technology, Tehran, Iran (Email: h.tajari@gmail.com; beigy@ce.sharif.edu)

II. Related Works

The foundations of Support Vector Machine (SVM) have been developed by Vapnik based on the foundations of the statistical learning theory [5]. Various algorithms based on support vector machine try to use transductive approach instead of inductive approach to minimize error on empirical data in training and working set at the same time [6]-[8]. SVM uses Structural Risk Minimization (SRM) to minimize empirical misclassification rate and the capacity of the classification function using only training data. According to structural risk minimization, for a fixed empirical misclassification rate, larger margins prevent overfitting and lead to better generalization [4], [9].

There are several algorithms that are proposed for transductive learning based on SVM. In this section we briefly review two methods that are superior to other methods.

A. Transductive vs. Inductive Approach

The main goal of learning classifiers is to assign labels to the working set (unlabeled data) using the training set (labeled data). If the working set is empty the methods become supervised learning. If the training is empty methods become unsupervised learning that many clustering approaches exist in such situations. Transductive and semi-supervised learning occur in the problem that both working set and training set are non-empty and used for learning phase of classifiers.

The transductive approach is used when the only concern is labeling working set while inductive approach tries to classify whole problem space. Inductive approach is to learn a decision rule based on training set and then apply it to working set. Fig. 1 illustrates separation using only training data. This may not seem perfect solution when size of training data is relatively small. In other words when we only want to know the labels of some points in feature space, inductive learning tries to solve bigger problem with little information. Fig. 2 shows hyperplane separator when transductive approach is applied. The output of transductive methods is only labels of working sets, not the classifier.

B. Support Vector Machine

Consider the problem of separating the set of binary class training example in feature space. Given a set of training example,

$$(x_1, y_1), \dots, (x_l, y_l) \in R^n \times \{-1, +1\} \tag{1}$$

where vector x_i is a sample of training set in the feature space, SVM aims to build the following separating hyperplane based on the training set.

$$w \cdot x + b = 0 \tag{2}$$

So that

$$w \cdot x_i + b \geq 1 \quad if \;\; y_i = 1 \tag{3}$$

And

$$w \cdot x_i + b \leq -1 \;\; if \; y_i = -1, i = 1, \dots, l \tag{4}$$

Or equivalently

$$y_i(w.x_i + b) \geq 1 \; i = 1, \dots, l \tag{5}$$

According to structural risk minimization, for a fixed empirical misclassification rate, larger margin leads to better generalization and prevents overfitting [4]. The hyperplane that separates vectors x_i correctly and has the maximal margin is optimal hyperplane [5]. The distance of margin given by

$$D = \min_{x_i | y_i = 1} \frac{w.x_i + b}{|w|} - \max_{x_i | y_i = -1} \frac{w.x_i + b}{|w|} \tag{6}$$

Hence to maximize the margin, we seek vector w that minimizes

$$\theta(w) = \frac{1}{2} \left|\left|W\right|\right|^2 \tag{7}$$

subject to constraint given in equation (5). The optimization problem can also be expressed as

$$\min_{w,b} \max_{\alpha} \left\{ \frac{1}{2} ||w||^2 - \sum_{i=1}^{l} \alpha_i [y_i (w.x_i + b) - 1] \right\} \tag{8}$$

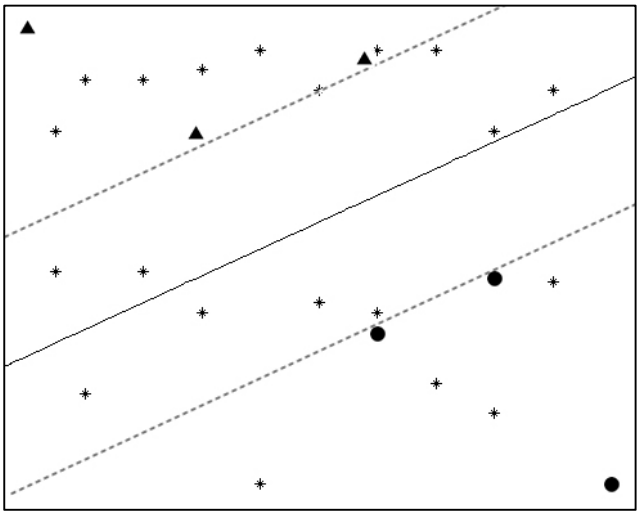

Fig. 1. Illustration of separator hyperplane constructed by SVM. triangle, circle an star are positive, negative and unlabeled points.

Above problem is known as hard margin SVM. For the non-linear separable case, soft margin SVM has been introduced by Vapnik. Soft margin allows SVM to have some mislabeled examples to maximize the margin. If there exist no hyperplane that split training example, the soft margin method chooses a hyperplane that splits data as cleanly as possible. This method uses slack variable, ξ_i which can be interpreted as error penalty. Constraints in this situation become:

$$y_i(w.x_i + b) \geq 1 - \xi_i, i = 1, \dots, l \tag{9}$$

And optimization problem becomes

$$\min_{w,\xi,b} \max_{\alpha} \left\{ \frac{1}{2} ||w||^2 + C \sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i [y_i (w.x_i + b) - 1] - \sum_{i=1}^{l} \beta_i \xi_i \right\} \tag{10}$$

Another property of SVM is kernel functions that can be used for non-linear classification. In this work we only focus on linear classification. For more information about kernel functions in SVM please refer to [5].

C. Transductive Inference in Support Vector Machine

The new type of inference, the transductive inference, has been introduced in order to improve performance on the given working set. For a class of linear indicator functions, Vapnik proved that bounds on test error rate are better than bounds on error rate for inductive inference. To formulate our discussion, we have

$$(x_1, y_1), \dots, (x_l, y_l) \in R^n \times \{-1, +1\} \tag{11}$$

$$(x_1^*, y_1^*), \dots, (x_k^*, y_k^*) \in R^n \times \{-1, +1\}$$

And objective function is to estimate $y_1^*, .., y_k^*$ that minimize the number of errors in the working set. Transductive inference suggests that solution with maximal margin separation leads to better performance. As a more

general setting (non-separable case), the optimization problem would be

$$\theta(w) = \frac{1}{2}\,||W||^2 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=1}^{k}\xi_i^* \quad (12)$$

$$s.t.\;\; y_i(w.x_i + b) \geq 1 - \xi_i, i = 1, \dots, l$$

$$y_i^*(w.x_i^* + b) \geq 1 - \xi_i^*, i = 1, \dots, k$$

$$\xi_i^* > 0, i = 1, \dots, l$$

$$\xi_i^* > 0, i = 1, \dots, k$$

Note that finding the optimal solution to this problem requires searching all possible values of $y_1^*, .., y_k^*$ and choose one that leads to maximal margin separation. This needs 2^k times of solving quadratic optimization that is infeasible for large numbers (more than 10 samples) of working set.
Some algorithms have been developed to approximate the optimal solution. We will discuss some of them in the rest of this section.

D. Related Algorithms

Currently, the most important work of transductive learning using SVMs is TSVM developed by Joachims [3]. TSVM tries to approximate the optimization problem (12). The most contribution of TSVM lies in the fact that it can solve the optimization algorithm that effectively handles large scale datasets. TSVM algorithm can be summarized in following steps:

Step 1: specify C and C^* where C is penalty factor for misclassification in training set and C^* is "effect factor" for unlabeled samples in working set. Assign a number N to the estimation of positive samples in working set.

Step 2: based on inductive inference classification using labeled samples, Label N samples with largest decision function value as positive samples and label others as negative. Assign a temporary factor C^*_+ and C^*_- for main loop.

Step 3: Train SVM with current assigned labels and current effect factor. Switch labels of two different-labeled samples that at least one of them is currently misclassified. Based on switching condition, it guarantees that objective function decreases. This procedure continues until no samples satisfy the switching condition.

Step 4: increase temporary factor C^*_+ and C^*_- slightly, if it exceeds the C^* factor, the algorithm is finished. Otherwise go to Step 3.

Optimization formula in TSVM is similar to the one that is presented by K. Bennett in the semi-supervised support vector machine (S3VM) [4]. But their approaches differ in solving optimization function.

Another algorithm in the transductive learning based on SVM is Progressive Transductive Support Vector Machine (PTSVM) developed by Chen, Wand and Dong [8]. PTSVM states that its method outperforms TSVM in some datasets and converges faster. The algorithm used in the PTSVM has no prior knowledge about positive and negative samples.

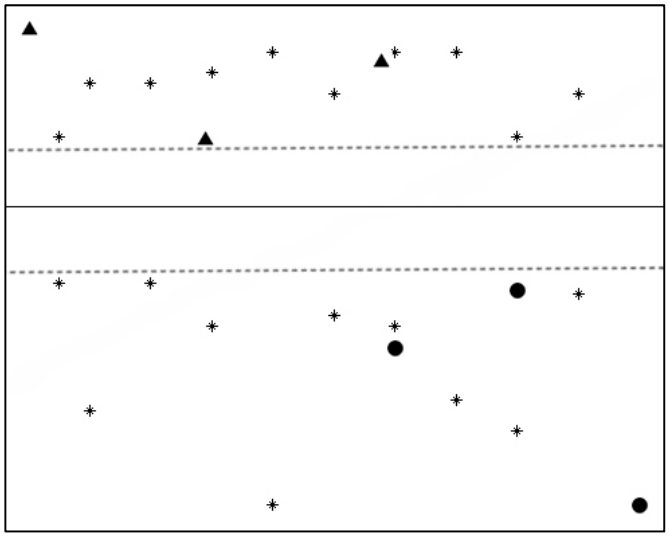

Fig. 2. Illustration of separator hyperplane constructed by Transductive approach. triangle, circle an star are positive, negative and unlabeled points.

Their progressive method selects the largest margin data in the margin area and labels it with current separation hyperplane. To prevent hyperplane to move just one side, at each step PTSVM labels positive and negative samples. We can summarize PTSVM in following steps:

Step 1: specify C and C^* where C is penalty factor for misclassification in training set and C^* is "effect factor" for unlabeled samples in working set.

Step 2: label one positive unlabeled sample and one negative unlabeled sample such that they satisfy the condition 9 or 10.

$$i_1 = arg \max_{i|0<f(x_i^*)<1} |f(x_i^*)| \quad (13)$$

$$i_2 = arg \max_{i|-1<f(x_i^*)<0} |f(x_i^*)| \quad (14)$$

Step 3: train SVM with training data and extra labeled data. Compute decision function value for every unlabeled example in the working set. Cancel all inconsistent labels.

Step 4: if no sample satisfy condition 9 or 10 anymore, label others based on decision function value and stop the algorithm, otherwise go to step 2.

When no samples satisfy these two conditions, it means that remaining samples lies out of margin area. In this situation PTSVM labels remaining samples based on classification function value and output labels.

The basic idea of PTSVM is moving the hyperplane based on most confidence unlabeled sample. PTSVM at each iteration cancels all the labels that are inconsistent with current hyperplane, so it doesn't guarantee that the algorithm will converge and it surely doesn't prove progress in each iteration.

III. Iterative Transductive Support Vector Machine

Experiments shows that TSVM can achieve better performance than inductive SVM in many applications because it successfully uses distribution information

implicitly in the working set.

However TSVM has its own drawbacks. Its performance is mainly sensitive to estimation of parameter N that has to be assigned in the beginning of the algorithm. Parameter N is the estimation of positive-negative sample ratio and positive-negative ratio of output will be around N. In practical applications this bias can be very destructive to method performance. For example in cancer classification datasets there is about half positive and half negative samples, but the ratio in real patients has a big deviation from datasets. Also, computation effort in TSVM can be high based on unlabeled data size. Based on computation needed for TSVM, PTSVM can be much faster than TSVM. PTSVM labeling is based on SVM decision hyperplane. A drawback of PTVM is that in this method, unlabeled data in margin area will be labeled before unlabeled data out of margin area. In other words, PTSVM tries to label most confident data in each step but it chooses data in margin area to label, that contradicts with its goal. In this situation, some data out of margin area will be switched in other steps.

A new transduction approach to SVM is proposed in this section to approximate the optimization problem presented before. In this method, positive-negative ratio in not specified with prior knowledge and in each Iteration some unlabeled data will be classified. We call this method Iterative Transductive Support Vector Machine (ITSVM).

The basic idea of ITSVM is choosing the most reliable sample and label it based on optimization formula. The method chooses the candidate sample as accurate as possible in each step and because of that there is no need to fix earlier decision. First of all we need to answer some questions:

1. *Could the distance from separator hyperplane be an accurate measurement?*
2. *Which sample is the most confident sample?*
3. *How do we label the most confident sample?*

Currently, the presented algorithms such as PTSVM chooses most confident samples based on the distance from separator hyperplane and label them based on SVM classification. Note that transductive approach will be used in area that inductive SVM doesn't achieve promising results. So finding samples and labeling them with current hyperplane may lead classifier to wrong direction. Also we have another issue, some samples have a same distance from hyperplane but choosing one or another can be very different for the learner. As you can see in Fig. 3, samples 1 and 2 have same distance from hyperplane. Based on iteration we can label sample 2 as negative and still have a separation hyperplane, but there is no way to label sample 1 as negative and be able to classify linear. This simple figure shows that the distance from the hyperplane may not be the best measurement. To answer the second and third question, we define the most confident sample as follows:

The most confident sample is the one that labeling it as positive (or negative) benefits the optimization formula the most than labeling it as negative (or positive).

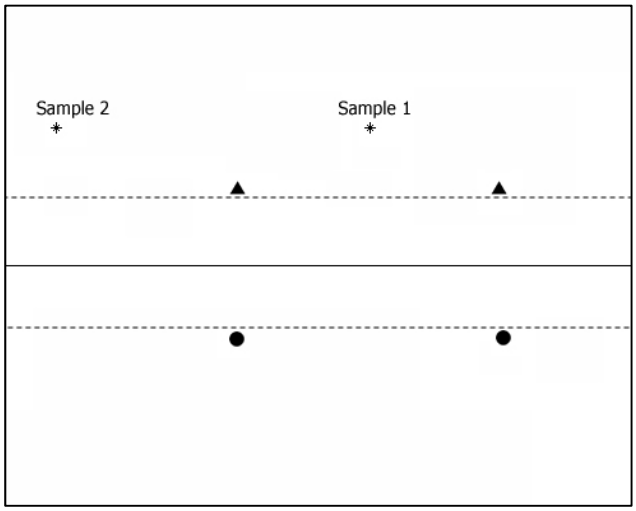

Fig. 3. Illustrating samples with same distance from hyperplane.

SVM hyperplane is linear combination of only Support Vectors and not entire training set, but the sample extracted from above definition is dependent to entire data training distributions.

Iterative methods suffer from earlier mistakes and fixing them is not always possible. Every labeling should be as accurate as possible. Adding every unlabeled sample to training set can shift and rotate hyperplane slightly, and we hope that this extra information leads the current hyperplane to the optimal solution.

The most confident sample can be computed based on formulation below. We can assign *optimization value* to any sample based on positive or negative labels.

$$\theta(w, y_{k+1}^*, x_{k+1}^*, k) = \frac{1}{2}||W||^2 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=1}^{k}\xi_i^* + C^*\xi_{k+1}^* \quad (15)$$

$$s.t.\ \ y_i(w.x_i + b) \geq 1 - \xi_i, i = 1, \dots, l$$

$$y_i^*(w.x_i^* + b) \geq 1 - \xi_i^*, i = 1, \dots, k+1$$

$$\xi_i^* > 0, i = 1, \dots, l$$

$$\xi_i^* > 0, i = 1, \dots, k{+}1$$

And we define *difference optimization value* of sample x_j^* (k is number of unlabeled data that is classified currently) as:

$$D(x_j^*, k) = \theta(w, -1, x_j^*, k) - \theta(w, +1, x_j^*, k) \quad (16)$$

In simple words, difference optimization value is the measurement of a sample labeling in optimization function. The positive (or negative) value means that positive (or negative) labeling of the sample leads to minimum value for the optimization method.

Now, we can summarize ITSVM in following steps:

Step 1, Specify parameters C^* and C.

Step 2, Set parameter k as unlabeled data count that are labeled so far. Compute optimization difference value for every sample in working set.
Step 3, If exists, find sample with the highest positive value. Label it as positive data and remove it from working set.

Step 4, If exists, find sample with the lowest negative value. Label it as negative data and remove it from working set.

Step 5, If exists, unlabeled data in working set, go to step 2, otherwise stop the algorithm.

The main contribution of this approach is that every decision is based on transductive objective function instead of sample distances of current hyperplane and decision function. This approach guarantees that all training samples contribute to choose the best sample to be labeled in each step.

IV. Experimental Results

Gene expression datasets usually have more than thousand features with less than hundred samples. In this problem space you can easily find a classifier that successfully separates training samples, but will perform poorly on test set. Generalization from this sparse space often leads to overfitting. Overfitting arises in areas that number of training patterns is relatively small due to the number of features. We use SVM Recursive Feature Elimination as a gene selection method. Using small subset of features (Genes) we can build a high accuracy classifier [10], [11].

The method SVM-RFE can be used for identification of discriminate genes which are of practical interest. Gene subset that we extract in this method can be very useful. Many irrelative genes are discarded in this procedure.

SVM-RFE is an iterative procedure based on ranking criterion. Ranking criterion can be computed from vector weights in SVM hyperplane. SVM-RFE builds a SVM classifier based on training samples in each step and computes ranking criterion for all features and discards feature with smallest ranking criterion.

In order to evaluate the proposed algorithm, computer experiments are conducted. In these experiments, we use the following gene expression data sets: Leukemia, Colon, and Prostate. The Leukemia dataset contains 72 Leukemia samples reported by Golub et al. [12]. It contains total of (training and test) 47 samples of acute lymphoblastic leukemia and 25 samples of acute myeloblastic leukemia. In this dataset, gene expression of 6,817 genes is collected. This dataset is available on web at http://www.genome wi.mit.edu/MPR

The colon dataset contains 62 samples from colon-cancer patients reported by Alon et al. [13]. It is a collection of 40 tumor biopsies samples are from tumors and 22 normal samples from healthy parts. 2,000 genes out of 6500 are selected based on confidence in experimental measurement. This dataset is available on web at
http://microaaray.princeton.edu/oncology

The prostate dataset contains 136 samples from non-tumor and tumor prostate samples reported by Dinesh Singh et al. [14]. It is a collection of 52 prostate tumor and 50 non-tumor samples with around 12,600 genes. This dataset is available on web at http://www-genome.wi.mit.edu/mpr/prostate

The cancer datasets Leukemia, Colon and Prostate is widely used for cancer classification. We used SVM-RFE to extract relevant genes and reduce feature space. In all experiments we selected top 100 genes of every dataset based on training set. This feature space is easily linear separable and we do not need any kernel function. Two experiments were designed for our method; classification accuracy and optimization values.

Table 1 shows average test error in 20 independent runs. We used different unlabeled sample size in each datasets and perform experiment. This could cover the cases were unlabeled samples are low or high. If one method could classify better in most cases, it would be more likely to do better in real world.

Test error values in Table 1 shows that ITSVM proposed in this paper performs better than inductive SVM in most experiments. It indicates our methods exploits from unlabeled data distribution for better classification. In nine out of twelve experiments the ITSVM achieved increase in generalization compared to other transductive methods. In most cases, our proposed method could separate gene expression samples more accurately. In each experiment the method which performs better is shown bold in Table 1 and Table 2.

Table 2 shows optimization values based on transductive optimization formula (12) for all methods. Based on theorem proposed by Vapnik, we try to minimize objective function and it will reduce average test error in general. As you could see in Table 2 ITSVM method could reduce the cost function compared to other methods and as a result larger margin could be obtained. In eleven out of twelve experiments ITSVM could optimize the equation better than related methods.

V. Conclusions

An *iterative transductive support vector machine* (ITSVM) is presented in this paper as an attempt to approximation the optimal solution. We define a new measurement for choosing and labeling sample in transductive inference in SVM. The experimental results show that ITSVM is not sensitive to datasets and informal decision in labeling samples can lead to better generalization. The proposed method calculates the quantitative value for unlabeled samples and chooses best action at each step. This feature could help us build a novel transductive multi-class classifier.

Classifiers used in cancer classification always suffer from relatively small number of training data; in this paper we study how samples with unlabeled data tackle this problem.

Still many questions are left, how do we apply ITSVM in multi-class problems? Kernel based methods can be used in ITSVM. Using kernel functions in ITSVM needs to be explored. How we use unknown patients' data to classify target patient? How well knowledge about relevant genes effects transductive methods accuracy? What is the effect of problem space VC-Dimension on accuracy? All this questions deserves further research.

TABLE I: AVERAGE TEST ERROR ACCURACY OF SVM,TSVM,PTSVM AND ITSVM ON DATASETS

Experiments			Averaging Test Error Rate			
Dataset	Unlabeled Samples	Labeled Samples	SVM	PTSVM	TSVM	ITSVM
Leukemia	12	60	4.0%	**2.8%**	18.4%	**2.8%**
	24	48	7.3%	5.8%	16.2%	**3.8%**
	36	36	7.6%	7.6%	14.7%	**5.8%**
	48	24	16.5%	12.7%	19.8%	**10.4%**
Colon	10	52	15.4%	15.4%	22.7%	**13.8%**
	21	41	19.8%	19.1%	22.8%	**18.0%**
	31	31	21.9%	19.3%	23.5%	**18.2%**
	41	21	24.4%	**23.3%**	24.9%	24.0%
Prostate	17	85	**6.7%**	7.4%	17.1%	7.4%
	34	68	9.1%	9.28%	15.3%	**7.6%**
	51	51	10.6%	**9.2%**	19.0%	9.5%
	68	34	17.0%	12.9%	18.8%	**10.6%**

TABLE II: OPTIMIZATION VALUE OF SVM,TSVM,PTSVM AND ITSVM ON DATASETS

Experiments			Averaging Test Error Rate			
Dataset	Unlabeled Samples	Labeled Samples	SVM	PTSVM	TSVM	ITSVM
Leukemia	12	60	0.281	0.279	0.454	**0.274**
	24	48	0.310	0.316	0.433	**0.306**
	36	36	0.390	0.390	0.534	**0.349**
	48	24	0.489	0.506	0.646	**0.479**
Colon	10	52	1.029	**0.943**	1.238	0.944
	21	41	1.026	1.020	1.255	**1.019**
	31	31	1.205	1.060	1.514	**1.032**
	41	21	1.164	1.060	1.626	**1.026**
Prostate	17	85	9.016	8.809	10.495	**8.658**
	34	68	9.407	10.394	11.570	**9.217**
	51	51	12.568	14.571	16.565	**10.082**
	68	34	14.657	13.795	18.011	**10.861**

REFERENCES

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

[2] S. Doduit, J. Fridlyand and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data", *Journal of American Statistical Association*, vol. 97, pp. 611-631, 2002.

[3] T. Joachims, "Transductive inference for text classification using support vector machines", *Proc. 16th International Conference on Machine Learning (ICML)*, 1999, pp. 200-209.

[4] K. Bennet and A. Demiriz, "Semi-Supervised support vector machines", *Advances in Neural Information Processing Systems*, vol. 11, pp. 368-374, 1998.

[5] V. Vapnik, *The nature of statistical learning theory*, New York: Springer-Verlag,, 1995,ch. 5.

[6] Z. Xu, R. Jin, J Zhu, I. King and M. Lyu, "Efficient convex relaxation for transductive support vector machine", *Proc. 16th ACM international conference on Multimedia*, 2008, pp. 41-50.

[7] Y. Li, J. Kwok and Z. Zhou, "Cost-Sensitive semi-supervised support vector machine", *Proc. 24th AAAI Conference on Artificial Intelligence*, 2010, pp. 500-5005.

[8] Y. Chen, G. Wang and S. Dong, "Learning with progressive transductive support vector machine", *Pattern Recognition Letters*, vol. 24, pp. 1845-1855, 2003.

[9] V. Vapnik, *Estimation of dependences based on empirical data*, New York, Inc., Secaucus, NJ: Springer-Verlag, 1982, ch. 6.

[10] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, "Gene selection for cancer classification using support vector machines", *Machine Learning*, vol. 46, pp. 389–422, 2002.

[11] K. Duan, J. Rajapakse, H. Wang, F Azuaje, "Multiple SVM-RFE for gene selection in cancer classification with expression data", *IEEE Transactions on NanoBioscience*, vol. 4, pp. 228-234, 2005.

[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 patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", *Proc. National Academy of Science,* vol. 96, 1999, pp. 6745-6750.

[14] Dinesh Singh, et al. "Gene expression correlates of clinical prostate cancer behavior". *Cancer Cell*, Vol. 1, 203-209, March, 2002.

Hossein Tajari, 1984, has received Bachelor's Degree in Software Engineering and Master Degree in Artificial Intelligence from Sharif University of Technology, Tehran, Iran in 2008 and 2011. His areas of interest are Machine Learning, Pattern Recognition and Bioinformatics.

Hamid Beigy, received the B.S. and M.S. degrees in Computer Engineering from the Shiraz University in Iran, in 1992 and 1995, respectively. He also received the Ph.D. degree in Computer Engineering from the Amirkabir University of Technology in Iran, in 2004. Currently, he is an Associate Professor in Department of Computer Engineering at the Sharif University of Technology, Tehran, Iran. His research interests include, learning systems, parallel algorithms, and computational intelligence.