Cancer Classification from Gene Expression Based Microarray Data Using SVM Ensemble

Shemim Begum
Government college of Engg and Textile Technology
Berhampore,Murshidabad
Email : shemim_begum@yahoo.com

Debasis Chakraborty
Electronics and Communication Engg. Dept.
Murshidabad College of Engg and Technology,Cossimbazar Raj
Email : debasisju67@gmail.com

Ram Sarkar
Assistant Professor
Computer science and Engg Department
Email : raamsarkar@gmail.com

***Abstract*—Ensemble classification, which is the combination of result of a set of base learner has achieved much priority in machine learning theory. It has explored enough prospective in improving the empirical performance. There are very little bit research in Support Vector Machines (SVMs) ensemble in contrast to Neural Network or Decision Tree ensemble. To bridge this gap we analyse and compare SVM ensemble (ADASVM) with K-Nearest Neighbour (KNN) and SVM classifiers. Leukemia dataset is used as benchmark to evaluate and compare the performances of ADASVM with KNN and SVM classifiers.**

keywords: Ensemble Classification; KNN; SVM; ADASVM;

I. Introduction

Ensemble method [1] is one of the significant achievement by the researchers in Machine Learning [2] [3], which produces high percentage of accuracy. A strong classifier can be obtained by integrating the many moderately perfect component classifiers. The ensemble classifier methods, which enlighten mostly the machine learning methods are bagging and boosting. AdaBoost is one of the most popular boosting methods [4], which assigns a set of weight over the training samples and each training sample is being updated at the end of each boosting iteration. Several techniques can be applied to update the weight of the training sample in AdaBoost. Lot of research have been done where Decision Tree or Neural Network has performed as a classifier. All these classifier produce a good generalisation performance, but they are suffering from some difficulties. Tree size, which plays an important role in Decision Tree learning algorithm impose a huge effect in achieving a good performance. In case of Neural Network as a component classifier, researchers needs to be concerned about how to get rid of overfitting. Optimum number of centres and optimum value for width can be considered as a measure to encounter the overfitting problem. All these factors needs to be deal very minutely for achieving a good generalised performance. In this paper we have shown AdaBoost incorporating linear SVM (SVM with linear kernel) as a component classifier [5] [6], which we call ADASVM perform better than SVM and KNN classifier. AdaBoost suffers from diversity. But very little bit measure has been taken to deal with diversity in existing AdaBoost algorithm.

SVM is developed from the theory of structural risk Minimization (SRM). The kernel trick in SVM help to map the training sample from an input space to high-dimensional feature space. SVM obtain an optimal separating hyperplane in the feature space and uses a regularization parameter C to control the model complexity and training errors. In this article we have taken SVM as a component classifier.

SVM [7] [8] with linear kernrel can be used as an effective component classifier in AdaBoost. AdaBoost can also be able to encounter the problem raised by accuracy/dilemma. SVM with linear kernel can be applied to solve a linear optimization problem. In addition with this if the number of features are really very large in comparison with training sample, it is always better to use linear kernel. Besides this if number of features are small, but training sample is large, linear kernel SVM would be good one to use. In ADASVM the boosting mechanism allows some linear SVM component classifier to focus on the misclassified sample from the minority class and this can prevent minority class from being considered as noise in the dominant class. ADASVM is exploring a convenient way to control the classification accuracy of each linear SVM as a component classifier.

II. Background

A. AdaBoost

The AdaBoost [9] algorithm is described as follows : A set of samples along with class label is given. The set is termed as the training set. The AdaBoost algorithm incorporates a weight distribution among the samples. Initially each sample is assigned with a weight of $1/M$, where M is the total number of samples. Then the algorithm calls the component classifier SVM. This will continue upto Tth iteration.

AdaBoost: Algorithm for two class Problem.

Input:$(x_1, y_1), ..., (x_m, y_m)$, where $x_i \in$X, $y_i = \{\pm 1\}$, the number of iteration $t = 1, ..., T$.

Step 1 :

Initialisation: The weights of training sample $w_i^1 = 1/M$, where $i = 1, 2, ..., M$.

Step 2 :

(a) The component classifier cl_t is trained on the weighted training samples.

(b) The weight error of the classifier cl_t is defined by : $\varepsilon_t = \sum_{i=1}^{M} w_i^t$, $y_i \neq cl_t(x_i)$.

(c) When error >0.5, then stop the process.

(d) The weight of the component classifier can be defined by : $\alpha_t = \frac{1}{2} \ln{(1 - \varepsilon_t)}/\varepsilon_t$.

(e) Now the training sample is being updated by : $w_i^{t+1} = w_i^t exp\{-\alpha_t y_i cl_t(x_i)\}/N_t$. where N_t is a normalisation factor and $\sum_{i=1}^{M} w_i^{t+1} = 1$.

Output: $f(x) = sign(\sum_{t=1}^{T} \alpha_t cl_t(x))$.

AdaBoost provides the training sample the weight distribution w_t, where t is the number of iteration. The weight w_t at any particular iteration t is updated at the end of each iteration following the prediction on the training sample. The correctly classified sample get reduction in weight and the samples that are misclassified by the classifier gets more weight. Thus AdaBoost highlights the sample achieving higher weight. The above procedure continues upto T number of iteration. The final or combined hypothesis $f(x)$ computes the sign of weighted combination of weak hypothesis. Thus $f(x)$ is computed as a weighted majority vote of the weak hypothesis cl_t, where each classifier is assigned weight α_t. The main criteria for AdaBoost is that if the weight error of the component more than 0.5 it stops the process.

B. K Nearest Neighbour:

An instance based learning method K-NN [10] [11] algorithm has been used in many application areas in data mining. We suppose that a set of n such vectors are given with their corresponding classes : $\{\mathrm{x}^{(i)}, y^{(i)}\}, where$i=1,2,$\cdots, n$. This set is referred to as the training set. Let x is a new sample that is to be classified.

It is simple to predict the class label of a sample in the training set when the number of neighbour is only one. The class label of the neighbour's sample is assigned to it. For *k*-NN we extend the idea of $1 - NN$ as follows : find the nearest *k* neighbours of the new sample x and then use a maximum voting rule [2] to classify the new sample. The advantage is that higher values of *k* provides the smoothing that reduces the risk of over-fitting due to noise in the training data. When we go for predicting the new sample, distance or dissimilarity index measure is being considered between the samples. Euclidian Distance: the most popular measure of distance is being considered here to classify the new sample.

C. Support Vector Machine

The SVM had been developed by Vapnik [12] [13] and became very popular due to it's immence interesting characteristics and fruitful empirical performance. SVM performs structural risk minimization, which override the traditional empirical risk minimization, commonly employed in conventional neural network. Structural risk minimization, which reduces an upper bound on the generalisation error contradicted with the empirical risk minimization, which minimizes error on the training data. SVM is ornamented with a greater ability to generalise, which is the major goal in statistical learning. SVM usually handles two-class problem. Let a data set $(\mathrm{x}_i, y_i), i = 1, 2, \cdots, M$, M is the total number of samples, $y_i = \{1, -1\}$. $x_i \in$R$, where$x$_i$ is a p dimensional real vector. In case of linear classification, the constrained optimization can be modelled as follows :

Minimize

$$1/2\|w\|^2 + C\sum_{i=1}^{m} \xi_i \tag{1}$$

such that

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \tag{2}$$

where $\xi_i \geq 0, i = 1, 2, ..., M$.

The parameter which compute the degree of misclassification of the sample x_i are denoted by ξ_i . The parameter *C* is termed as the error penalty, penalizing the non-zero ξ_i. The parameter *w* is the weight vector, *b* denotes the bias of the hyperplane. In case of data are linearly separable, the SVM search for the separating hyperplane with the maximum margin and then also $\xi_i = 0$. The above optimization problem can be solved by transfering the problem into the equivalent lagrangian problem:

Minimize

$$L(w, b, \alpha) = 1/2\|w\|^2 - \sum_{i=1}^{M} \alpha_i y_i (w_i x_i + b) + \sum_{i=1}^{M} \alpha_i \tag{3}$$

Eq.(3) can be resolved with the help of partial derivatives of L with respect to *B* and with respect to *W* and the following Eqs can be obtained:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{i=M} \alpha_i y_i x_i = 0 \tag{4}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{M} \alpha_i y_i = 0 \tag{5}$$

Now substituting Eq.(4) and Eq.(5) into Eq.(3), the quadratic optimization problem can be transformed into Eq.(6), that needs to be maximized with respect to α subject to Eq.(4) and Eq.(5):

Maximize

$$L(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=0}^{M} \alpha_i \alpha_j y_i y_j x_i . x_j \tag{6}$$

$$\sum_{i=1}^{i=M} \alpha_i y_i = 0, \tag{7}$$

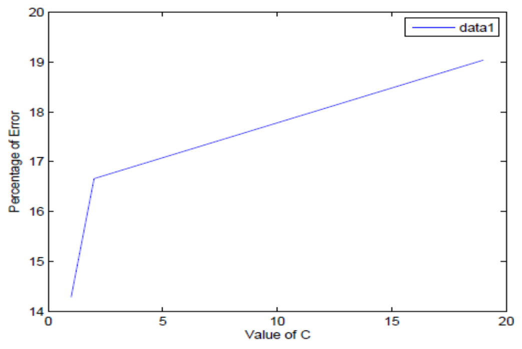

Fig. 1. Variation of error versus C

where $\qquad 0 \leq \alpha_i \leq$C.

Now following Karush Kuhn-Tucker (KTT) "complementarity" condition [12], the solution of the above optimization problem must follow the Eq.(8), which is as follows:

$$\alpha_i[y_i(wx_i+b)-1]=0 \tag{8}$$

for any given i, there will be either α_i^* =0 or y$_i(wx_i+b)$ =1. The training data vector x_i corresponding to α_i^* $\neq$0 are called the support vectors (SVs). In accordance with the SVs, the optimal separating hyperplane can be represented as

$$f(x,\alpha_i^*,b^*)=\sum_{i\in sv}^{M}\alpha_i^* y_i(x_i.x)+b^* \tag{9}$$

.

Depending on the SVs, in the future testing, the decision for testing data vector z is as follows:

$$h(z,\alpha_i^*,b^*)=sgn(\sum_{i}^{M}\alpha_i^* y_i(x_i.z)+b^*) \tag{10}$$

The model explained above only for linear classification with two-class labels. In this article we have considered linear kernel for the SVM classifier.

D. Influence of the parameter C in SVM :

The regularisation parameter C controls the trade-off between obtaining a low error on the training data and also reducing the norm of the weight. Tuning the value of C is really vital steps in SVM to achieve high accuracy. The higher value of C increases the complexity of the hypothesis classes. If the value of C can be increased slightly, we can still form all of the linear models that we could before and also some that we could not before. We increased the upper bound on the allowable norm of the weight as well as implementing Structural Risk Minimization (SRM) via maximum margin classification, it can also be implemented by the marginal complexity of the hypothesis class through maintaining C.

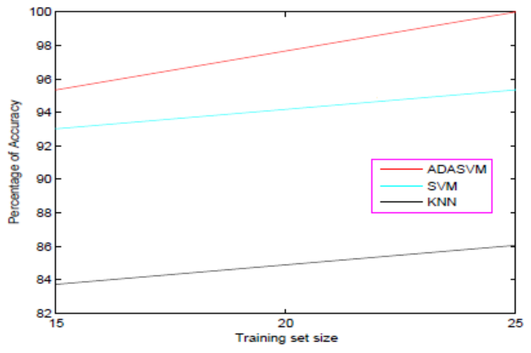

Fig. 2. A Binary classification can be viewed as the task of separating classes in feature space

How the percentage of error varies with the value of C is shown in FigI.

III. Proposed Technique

Initially the data set is divided into training set and test set respectively. Here S is the training set, which contains l elements. The test set V is contains j elements. Each sample x_i in the training set has its class label y_i. Initially each sample maintains a wt $W_i^t=1/l$. Then the algorithm ADASVM calls the component classifier h_t. At each iteration the number of misclassified sample set is denoted by M^t.

ADASVM : Algorithm for two class Problem
Input : Labeled Points $S=[(x_i,y_i)]$,i=1,......,l, unlabeled points V=[(x$_i$)]$,i=l+1,.....,n$,y$_i=\{\pm 1\}$, a component learn algorithm, the number of cycles $t=1,......,T$.
Step 1:
Initialisation : The weights of training sample is initialised as $W_i^1=1/l$, where $i=1,2,........,l$. S is the training set and V is the test set.
for t=1:T
Step 2:
(a) The component classifier h_t is trained on the weighted training set T_r^t.
(b) Obtain the label vector of the unlabeled set V.
(c) The set M^t denotes the set containing the misclassified samples at iteration t.
(d) The weight error of the classifier h_t is calculated as : $\varepsilon_t=\sum_{i=1}^{N}W_i^t$, where $y_i\neq h_t(x_i)$.
(e) If $\varepsilon>0.5$, then reassign the weight of each sample in test set by $W_i^t=1/j$, where j is the size of the test length and go to step 2. .
(f) The weight for the component classifier h_t is denoted by : $\alpha_t=\frac{1}{2}\ln{(1-\varepsilon_t)}/\varepsilon_t$.
(g) The weight of the test set samples is updated by :$W_i^{t+1}=W_i^t exp\{-\alpha_t y_i h_t(x_i)\}/C_t$, where $i=l+1,2,......,N$. where C_t is a normalisation factor and $\sum_{i=l+1}^{N}W_i^{t+1}=1$.
(h) The training set is updated by : $T_r^t=T_r^{t-1}\cup M^t$
Output: $f(x)=sign(\sum_{t=1}^{T}\alpha_t h_t(x))$.

The training set is updated at each iteration t. The new training set T_r^t at iteration t is formed by combining the previous training set T_r^{t-1} with the misclassified sample set M^t. The weight of each sample at each iteration t is updated following the prediction of the classifier h_t. The misclassified samples are readded to the training set maintaining the size of the training set as before. The weight error of the classifier h_t is calculated. If it crosses the value 0.5, the weight of the test set sample is reinitialised to $1/j$, where j is the total number of samples in the test set. The above procedure continues upto T iteration. The final or combined hypothesis f(x) is computed as the sign of weighted combination of weak hypothesis.

IV. Experimental Results:

In this section, our proposed ADASVM is compared with SVM and KNN classifiers. Here Leukemia dataset is used to compare their performances. We have taken training set size as 15 and 25. The percentage of accuracy obtained from ADASVM overcomes the other two clsassifiers. The accuracies obtained by applying the said three classifiers with the test set of size 43 is shown in the Table I.

A. Dataset:

Leukemia Dataset: Leukemia is the primary disorder of bone marrow. They are malignant neoplasms of hematopoietic stem cells. The leukemia dataset was taken from a collection of leukemia patients sample reported by Golub [14]. This dataset can often act as benchmark for microarray analysis methods. It contains two sub types of the disease such as acute lymphoblast leukemia(ALL) and acute myeloid leukemia (AML) from bone marrow and peripheral blood. The dataset contains of 72 samples: 25 samples of sub type AML, and 47 samples of sub type ALL. Each of the sample is judged over 7,169 genes. The details of which can be available in [15]. Leukemia dataset is used to evaluate the generalisation performance of the proposed ADASVM, SVM and KNN classifiers. The number of training sample range from 1 to 15 and 1 to 25 and the number of test sample ranges from 30 to 72. Each dataset is partitioned into training and test subsets respectively. Two partitions are generated randomly for the experiments. Each partition is trained and tested using the three algorithms respectively.

B. Selection Of Gene Markers

Since the gene microarray dataset consists of thousands of genes, hence it is very vital to find out the gene markers that affects mostly the identification of a particular subtype of cancer. In this paper we have used the Consistency Based Feature Selection (CBFS) algorithm for the selection of biomarkers. The selected biomarkers are: (Leukemia $M27891\_at$, $Y07604\_at$). The biomarkers are used to test the effectiveness of the classifier.

V. Conclusion

ADASVM, which is a new technique rather than tradional AdaBoost is proposed in this paper. Experimental results on benchmark dataset demonstrate that proposed ADASVM performs better than the component classifier SVM and KNN.

TABLE I
Experimental Results of ADASVM SVM and KNN classifiers on Leukemia Dataset

Dataset	*Testsetsize*	*Trainingsetsize*	*ADASVM*	*SVM*	*KNN*
Leukemia	43	15	95.34	93.02	83.72
		25	100	95.34	86.05

An improved version of ADASVM can also be developed to deal with accuracy/diversity dilemma in ADASVM algorithm giving rising to better generalisation performance.

References

[1] T. G. Dietterich, *Ensemble Methods in Machine Learning,* Multiple Classifier System, Lecture Notes in Computer Science, Volume 1857, 2000, pp 1-15.

[2] D. Optiz, R. Maclin, *Popular Ensemble Methods : An Empirical Study,* Journal Of Artificial Intelligence Research 11, 1999.

[3] H. Daume III, *A Course in Machine Learning,* version0.8, 2012.

[4] R. E. Schapire, *Explaining AdaBoost,* Springer, 2013.

[5] H. T. Lin, L. Li, *Support vector Machienary for Infinite Ensemble Learning,* Journal of Machine Learning Research, 2008.

[6] R. R. Guerra, J. Stork, *Case study Report: Building and analyzing SVM ensembles with Bagging and AdaBoost on BIg data sets,* 2013.

[7] S. wang, A. Mathew, Y. Chen, L.Feng, L. Ma, J. Lee, *Emperical Analysis of support Vector Machine* , Expert Systems with Applications, 36, pp. 6466-6476, 2008.

[8] Dietterich,T.G.., *An Experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting.and radomization,* Machine Learning 40(2), 139-157, 2000.

[9] J. Kodovsky, *Ensemble classification in stegenalysis cross-validation and AdaBoost,* 2011.

[10] S. B. Imandoust and M. Bolandraftar, *Application of K-Nearest Neighbour(KNN)Approach for predicting Economic Events:Theoretical Background,* S B Imandoust et al. Int. Journal of Engineering Research and Application, vol. 3, 2013.

[11] S. Thirumuraganathan, *A Detailed Introduction to K-Nearest Neighbour(KNN) Algorithm,* 2010.

[12] C. J. C. Burges, *A Tutorial On Support Vector Machines for Pattern Recognition,* Data Mining and Knowledge Discovery, 1998.

[13] J. P. Lewis, *A short SVM Tutorial,* , U. Southern California, version 0.zz, 2004.

[14] Golub, T R and Slonim,......, *Molecular classification of Cancer : Class Discovery and Class Predict,* Journal: Science, 1999.

[15] A. Be-Dor, L. Bruhn, N. Friedman, I. Nachman, N. Schummer and Z. Yakhini, *Tissue Classification With Gene Expression Profiles,* 2000.