Statistics and Its Interface Volume 2 (2009) 153–159

Weighted random subspace method for high dimensional data classification

Xiaoye Li and Hongyu Zhao*

High dimensional data, especially those emerging from genomics and proteomics studies, pose significant challenges to traditional classification algorithms because the performance of these algorithms may substantially deteriorate due to high dimensionality and existence of many noisy features in these data. To address these problems, pre-classification feature selection and aggregating algorithms have been proposed. However, most feature selection procedures either fail to consider potential interactions among the features or tend to over fit the data. The aggregating algorithms, e.g. the bagging predictor, the boosting algorithm, the random subspace method, and the Random Forests algorithm, are promising in handling high dimensional data. However, there is a lack of attention to optimal weight assignments to individual classifiers and this has prevented these algorithms from achieving better classification accuracy. In this article, we formulate the weight assignment problem and propose a heuristic optimization solution.

We have applied the proposed weight assignment procedures to the random subspace method to develop a weighted random subspace method. Several public gene expression and mass spectrometry data sets at the Kent Ridge biomedical data repository have been analyzed by this novel method. We have found that significant improvement over the common equal weight assignment scheme may be achieved by our method.

Keywords and phrases: Classification, Aggregating algorithm, Voting weight, Random subspace projection.

1. INTRODUCTION

With rapid improvement of computing power and innovations in biological technology, huge amounts of biological data can be produced, recorded, stored and shared rather easily today. These data contain information that can be used to gain biological insight and to make further deductions or predictions based on the knowledge acquired. For the bioinformatics community, the microarray and mass spectrometry technologies have drawn great attention because of their abilities to measure the expression levels of thousands of genes or proteins simultaneously. Such information can be analyzed by statistical and data mining methods to identify disease related biomarkers and to understand the mechanism of the underlying biological processes, e.g. [5, 15]. However, the high dimensionality of genomics and proteomics data and the existence of many noisy features can substantially deteriorate the performance of many traditional classification algorithms. For example, the linear discriminant analysis (LDA) and the quadratic discriminant analysis (QDA) are not directly applicable because they require the number of features be less than the number of samples which is usually not the case for gene expression and mass spectrometry data. The performances of the support vector machine (SVM) and the k-nearest-neighbor classifier ($k$nn) also decay quickly with an increasing number of noisy features despite their ability to handle a large number of features. Therefore, various procedures have been proposed to tackle these problems posed by high dimensional data, including feature selection and individual classifier aggregation.

The pre-classification feature selection is commonly used to directly reduce the number of features and only those features likely to be differentially expressed among different classes are selected for classification analysis. The feature selection methods fall into two categories, namely, the filters and the wrappers, e.g. [8, 12]. The filter approach evaluates the discriminant power of each feature individually. It neglects potential interactions among the features. The wrapper approach selects the combination of features that has the best performance on the training set, which may cause over-fitting when the sample is small relative to the number of features. In practice, although the feature selection step helps to eliminate the noisy features, it also likely eliminates potentially informative features from the data. Therefore, the information in the original data may not be fully utilized due to the removal of informative features. In addition, feature selection increases the chance of over-fitting by constructing a classifier on a small fraction of features that may only have good performance on the training set. Therefore, the feature selection approach does not provide a satisfactory solution to the problems in high dimensional data classification.

Aggregating algorithms have been proposed in recent years for high dimensional data classification to overcome over-fitting and improve prediction accuracy. These methods

*Corresponding author.

have enjoyed great popularity because of their good performance. They include the bagging predictor [1], the boosting algorithm [4], and the Random Forests algorithm [3]. They share some common features such as the perturbation of the original training set to reduce over-fitting and the aggregations of a large number of individual classifiers hoping that the majority of the classifiers would make correct predictions. Random Forests introduces the random feature selection step into the traditional tree classifier to further reduce over-fitting and achieves better performance over the bagging predictor. The boosting algorithm adaptively changes the sampling probabilities in bootstrapping and the voting weights in aggregating according to the performance of the individual classifiers on the training set. Since these aggregating algorithms all use the tree classifier as the base learner which can handle high dimensional data, they can circumvent the feature selection step to make full use of the information contained in all the features. The implicit weight adjustment among individual classifiers through the random feature selection mechanism in Random Forests and the explicit weight assignment in the boosting algorithm help to reduce the effects from the noisy features. However, the bootstrap distorts the original distribution of the training samples and achieves improved performance only with instable algorithms like the tree classifier [2]. Furthermore, the weight adjustment procedures introduced in Random Forests and the boosting algorithm do not consider the correlations among individual classifiers, leaving room for improved weight assignment scheme.

The random subspace method introduced by Ho (1998) represents a distinct aggregating method that has a fundamentally different philosophy from the above aggregating methods. The random subspace method originated from the stochastic discriminant analysis [10, 11], which uses weak classifiers that are not necessarily very accurate but generalize well as base learner and achieves good prediction accuracy by aggregating many different such individual classifiers. The random subspace method generates multiple individual classifiers by projecting the original feature space into different subspaces. The projection from the high dimensional feature space into the low dimensional space can avoid the problems caused by high dimensionality. And all the information in the original data is maintained by aggregating many individual classifiers based on different subspaces. However, the original random subspace method did not emphasize the importance of selecting a base learner that should generalize well. Also, all individual classifiers have equal weights so the existence of uninformative subspaces would still deteriorate the performance of the aggregated classifier.

To address the limitations of the current methods, we consider alternative approaches for weight assignments in this article. More specifically, we incorporate the correlations among individual classifiers. As for individual classifiers, we focus on the random subspace idea to generate individual classifiers on the low dimensional subspaces. In contrast to perturbing the original samples, we believe that the random projection into subspaces can keep the completeness of the information in the original data. At the same time, it can substantially reduce the feature number so that many good traditional classifiers can be used as base learner. In this article, we focus on SVM as our base learner because it has good performance on low dimensional data and generalizes well, making the weights derived from the training set have good performance on the test data.

The layout of this article is as follows: In the Methods section, we discuss our proposed weight assignment schemes. In the Experiments section we evaluate the performance of our weight assignment procedures based on a number of gene expression and mass spectrometry data sets. We conclude this article by discussing the base learner selection and weight assignment issues.

2. METHOD

Individual classifiers are built by randomly projecting the original data into subspaces and training proper base learner on these subspaces to capture possible patterns that are informative on classification. Because SVM with appropriate kernels has the ability to classify sample with nonlinear boundary and generalizes well, we use SVM here as the base learner.

For high dimensional data, because most subspaces may only contain noisy feature and individual classifiers developed from these subspaces are not informative, an equal voting weight scheme may not be ideal. Intuitively, patterns with better classification accuracy should be assigned more weight. In the following, we discuss possible weight assignment schemes. We first consider a simplified scenario where all the individual classifiers are independent of each other, and derive an optimal weight under this assumption as shown in the following theorem.

2.1 Weight assignment for the independent case

Theorem 2.1 (Optimal weight assignment for the independent case)**.** *Suppose we have n independent individual classifiers to classify the samples with the classification error rates $e_1, e_2, \ldots, e_n$ respectively. If these classifiers have voting weights $q_1, q_2, \ldots, q_n$ respectively where $q_1+q_2+\cdots+q_n = 1$, the error rate for the aggregated classifier is*

$$\sum_{Q}\Big[\prod_{q_i\in Q} e_i \prod_{q_j\in Q^c}(1-e_j)\Big],$$

where Q is any subset of $\{q_1, q_2, \ldots, q_n\}$ that satisfies $\sum_{q\in Q} q > 1/2$, and Q^c is the complement of Q. The weight assignments $q_k = \frac{1}{Const}\log\frac{1-e_k}{e_k}$, where $Const = \sum_{k=1}^{n}\times \log\frac{1-e_k}{e_k}$ to make the q_k sum up to 1, achieve the minimal error rate.

Proof. Without loss of generality, we suppose $e_1 \leq e_2 \leq \cdots \leq e_n$. Note that for any subset of $\{q_1, q_2, \ldots, q_n\}$ whose sum is greater than 1/2, its complement has sum less than 1/2. Therefore, there are a total of 2^{n-1} subsets with sum greater than 1/2. So to minimize the error rate for the aggregated classifier, we only need to make sure that out of the possible 2^n subsets, we assign the weights such that the 2^{n-1} subsets Qs with the smallest values of $\prod_{q_i \in Q} e_i \prod_{q_j \in Q^c}(1-e_j)$ satisfy $\sum_{q \in Q} q > \frac{1}{2}$. This is equivalent to the following condition: given any two subsets P and Q, if

$$\prod_{q_i \in P} e_i \prod_{q_j \in P^c} (1-e_j) \geq \prod_{q_i \in Q} e_i \prod_{q_j \in Q^c} (1-e_j),$$

then $\sum_{q \in P} \leq \sum_{q \in Q} q$. Taking the logarithm of the last equation, we obtain:

$$\sum_{q_i \in P \setminus Q} \log \frac{1-e_i}{e_i} \leq \sum_{q_j \in Q \setminus P} \log \frac{1-e_j}{e_j}.$$

Thus, if we let $q_k = \log \frac{1-e_k}{e_k}$ for $k = 1, 2, \ldots, n$, we obtain $\sum_{q \in P \setminus Q} q \leq \sum_{q \in Q \setminus P} q$, which can be used to deduce that $\sum_{q \in P} q \leq \sum_{q \in Q} q$. Therefore the weight assignment achieves the minimal error rate.

The above theorem gives the exact weight for each individual classifier according to its performance to achieve the best aggregated accuracy. For individual weak classifiers with error rate 1/2, no weight will be assigned to them and for classifiers with error rate 0, all weight will be assigned to it, which is intuitively true.

We observe that the weights given in the above theorem are exactly the same as the heuristic weights used in the boosting algorithm. Thus, our results suggest that if we assume that all the individual classifiers are independent, the boosting algorithm gives the optimal aggregating accuracy. □

2.2 Weight assignment for the dependent case

Under our current random subspace sampling scheme, especially when the subspace dimension is large, it is very possible that there are overlaps of the features used in constructing individual classifiers on different subspaces. Furthermore, the original features may already have strong correlations among them. Thus, some of the individual classifiers based on the selected subspaces can show correlated classification results. Therefore, there are likely complicated correlations among the individual classifiers and the independence assumption discussed in the previous subsection usually does not hold. A weight assignment mechanism should incorporate information about the potential complicated correlations among individual classifiers in the ensemble of individual classifiers.

We first formally formulate the problem. Suppose there are k weak classifiers X_i, where $i = 1, \ldots, k$. Each X_i takes on two possible values, 1 with probability e_i, which represents misclassification, and 0 otherwise. Our objective is to find a weight assignment w_i for each classifier, satisfying $w_i \geq 0$ and $\sum_{i=1}^k w_i = 1$, such that the aggregated error rate $e = P\{\sum_{i=1}^k w_i X_i > 1/2\}$ is minimized. If the joint distribution of $(X_1, \ldots, X_k)$ is known, the distribution of $\sum_{i=1}^k w_i X_i$ in terms of the w_i can in principle be derived and the optimal weight assignment that minimizes the probability that a sample is misclassified may be identified. However, the joint distribution is often unknown and this optimization problem does not lead to easy numerical solutions.

We propose to formulate this problem from an alternative perspective that can be solved easily and has some statistical justifications. Because the aggregated classifier can be viewed as a random variable in the above paragraph and we can cast our goal as to minimize both its bias (from 0) and variance. The expectation of the square loss $\mathrm{E}(\sum_{i=1}^k w_i X_i)^2$ can be written as $(\mathrm{E}(\sum_{i=1}^k w_i X_i))^2 + \mathrm{Var}(\sum_{i=1}^k w_i X_i)$, which is the sum of the squared bias and variance. By minimizing this objective function, we simultaneously control the bias and variance and the aggregated classifier can be expected to achieve good performance. The objective function is a quadratic form of the w_is and can be easily solved. To summarize, our goal is to find the w_i:

$$\min\left\{ \sum_{i=1}^k (\mathrm{E}X_i^2) w_i^2 + 2 \sum_{1 \leq i < j \leq k} (\mathrm{E}X_i X_j) w_i w_j \right\}$$

subject to:

$$\sum_{i=1}^k w_i = 1,$$
$$w_i \geq 0 \text{ for } i = 1, \ldots, k.$$

In this simplified problem, only pair-wise correlations between individual classifiers are needed. The joint distribution of pair-wise weak classifiers can be estimated using the classification result on the training set. Actually, if we let a denote the number of samples both the classifiers X_i and X_j make correct classification, b denote the number of samples where both classifiers are wrong, c and d denote the number of samples where the two classifiers have opposite results, we have $\hat{P}\{X_i = 0, X_j = 0\} = \frac{a}{n}$, $\hat{P}\{X_i = 1, X_j = 1\} = \frac{b}{n}$, $\hat{P}\{X_i = 0, X_j = 1\} = \frac{c}{n}$, and $\hat{P}\{X_i = 1, X_j = 0\} = \frac{d}{n}$, where n is the total number of samples. Therefore $\mathrm{E}X_i X_j$ can be calculated as $\frac{b}{n}$. By solving the above quadratic programming problem, we may obtain better weight assignment than the equal weight assignment. In the following experiments, we used the "quadprog" package in the R software to solve the above quadratic programming problem. Its "solve.QP" function implements the dual method of Goldfarb and Idnani for solving quadratic programming problems of the form $min - b^T x + 1/2 x^T A x$ with the constraints

$C^T x \geq x_0$. We can easily reformat our problem in the matrix notation and use "quadprog" to obtain better weights for our aggregated classifier.

3. EXPERIMENTS AND RESULTS

To assess the performance of the proposed weighting schemes, we have conducted several experiments on the two-class classification problem using the data sets obtained from the Kent Ridge Bio-medical Data Set Repository. Four data sets are used here: the leukemia gene expression data, the breast cancer gene expression data, the lung cancer gene expression data, and the ovarian cancer protein mass spectrometry data. The number of the features in these four data sets ranges from several thousands to dozens of thousands. These four data sets are selected out of the repository because they are all two-class classification problems and they have a relatively large number of training samples and somewhat balanced distribution between the two classes. The details about these data sets are as follows.

The leukemia data set was originally used in [6] as an example of the generic approach to cancer classification based on gene expression profiles measured from DNA microarrays. The training set consists of 38 bone marrow samples (27 ALL and 11 AML) with over 7,129 probes from 6,817 human genes. The test set is also provided, with 20 ALL and 14 AML cases. The breast cancer data was from [14] that identifies a gene expression signature to accurately predict breast cancer status. In this data set, 78 (34/44)[1] training samples and 19 (12/7) test samples were collected and 24,481 genes were measured in the microarray. The lung cancer data used in [7] has 32 (16/16) training samples and 149 (15/134) test samples with each sample measured on 12,533 genes. The ovarian cancer data in [13] is the only protein expression data. There are 253 (162/91) samples without explicit separation into the training and the test sets. Each sample contains the relative amplitude of the intensities at 15,154 detected mass/charge (m/z) identity. The characteristics of these data sets are summarized in Table 1.

In the following experiments, SVM is selected as the base learner of the random subspace method. We used the "svm" function in R package "e1071" to train our classifier. The dimension of the randomly selected subspace is set to be 10. We generate 1500 individual learners in total and plot the error rate as we vary the number of individual learners used for aggregation. The choice of number of individual learners and subspace dimension are heuristic. There are several considerations in these choices. First we want to have enough base learners so that error rates converge to a stable level. Second, it is very time consuming to generate SVM base learners and solve the weight assignment quadratic programming problem. We tried several choices of the number of base learners, $1500 \sim 2000$ is a reasonable choice. As for the subspace dimension, the error rate is not very sensitive to it. We selected a small number 10 in hope that the selected subspaces don't overlap in features. In the discussion section, we will discuss further on this topic. The prediction error rate is based on the test set error rate if there exists one test set and the mean error rate of 10 iterations of randomly splitting the original data set into the training and the test sets if no independent test set exists.

[1]The number before the slash denotes the number of case samples and the number after the slash denotes the number of control samples.

Table 1. Characteristics of the high dimensional data sets

Data set	Training Set		Test set		Features
	Case	Control	Case	Control	
Leukemia	27	11	20	14	7129
Breast Cancer	34	44	12	7	24481
Lung Cancer	16	16	15	134	12533
Ovarian Cancer	253 (162/91)				15154

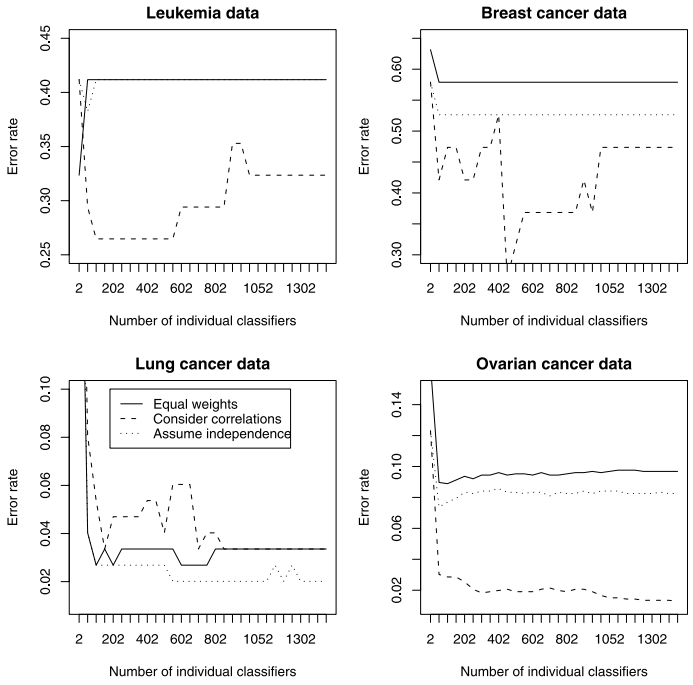

Figure 1. Comparison of different weight assignment procedures with SVM as the base learner. The solid line shows the result of equal weighting, the dotted line for assuming independence, and the dashed line for considering correlation.

In the experiments, we compare the performances of the three weight assignment procedures we discussed above, which are equal weight assignments, optimal assignments for the independent case and weight assignments considering correlations among the individual classifiers.

As observed from Figure 1, our proposed weight assignment procedure incorporating correlations has significant improvement over equal weight assignment except for the lung cancer data set when all classifiers have low error rate.

4. DISCUSSION

In this article, we have formulated the weight assignment problems in the aggregating algorithms and proposed weighting schemes both under the independent and dependent situations. Our experiments confirmed the improvement of our weight assignment procedures in terms of prediction accuracy.

To explore the outcome of applying the proposed weight assignment procedures to over-fitting base learners, we apply the weight assignment procedures to the Random Forests algorithm to check if we can improve its performance. We extracted the individual trees from the Random Forests algorithm and assigned weights to them according to our developed procedure to compare with equal weight assignment. Thus, we used exactly the same set of underlying trees but with different weights. The results are shown in Figure 2. We observe that the weighted Random Forests method outperform Random Forests with equal weight on the leukemia data and the ovarian cancer data. However, the results on the breast cancer data and lung cancer data show a similar error rate or even a little bit worse for the weighted method. We investigated the cause of the degradation and found that it was caused by the over-fitting of tree classifiers. The tree classifier is too over-fitting to have consistent performance on the training set and the test set. The trees that were assigned large weights because of good performances on the training set tended to have ordinary performances or sometimes poor performances on the test set. The effect of wrongly assigned weights to poorly performed classifier is substantial, which can result in the degraded performance of the weighted Random Forests method.

The problem exposed by inconsistent individual classifiers reminds us of the risk to use re-substitution error estimate on the training set to determine the weights of individual classifiers. We would suggest the introduction of an independent validation set if there are enough samples available. The performance of weak classifiers on the validation set should be used to determine the weight. Most of the time, due to the limited number of samples, a cross validation error estimate on the training set can be used.

We also compare the performance of the Random Forests algorithm and the weighted random subspace method. The subspace dimension in the random subspace method and the size of the randomly selected subset ("mtry") in the Random Forests algorithm are both set to be 10. Totally, 2,000 weak classifiers are generated. From the results shown in Figure 3, we observe that the random subspace space method is competitive to the Random Forests algorithm. The random subspace method has improved performance over Random Forests on the leukemia data and the ovarian data but is slightly worse on the breast cancer data and the lung cancer data. It appears that weighted random subspace method does not uniformly outperform the Random

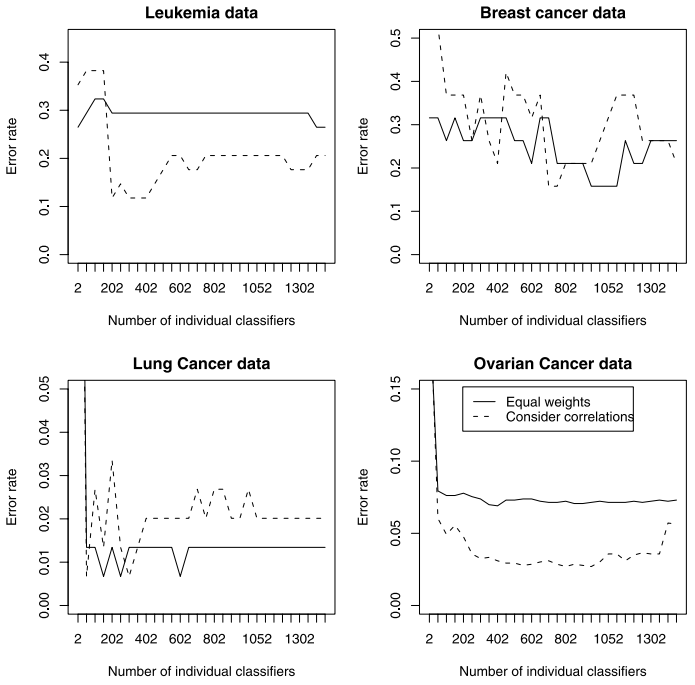

Figure 2. Comparison of different weight assignment procedures with tree classifier as the base learner. The solid line shows the result of equal weight assignment and the dashed line for considering correlations.

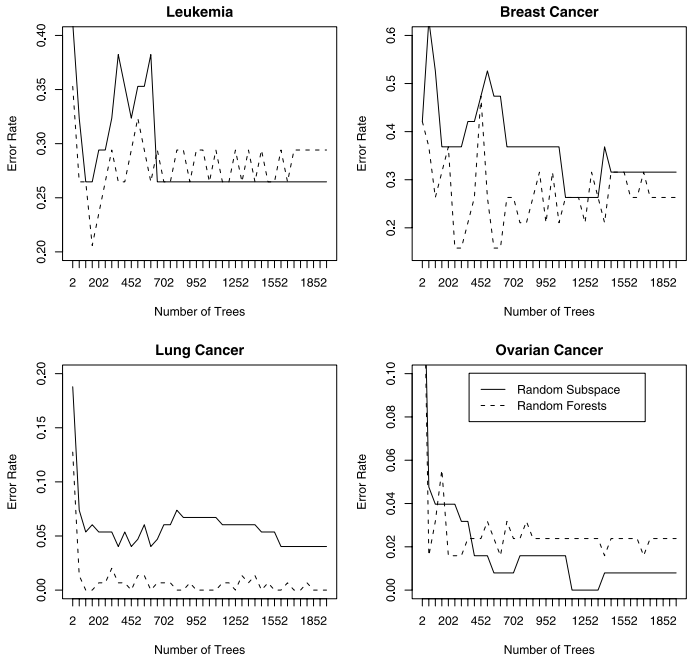

Figure 3. Comparison of the Random Forests algorithm and the weighted random subspace method. The solid line shows the results of the weighted random subspace method and the dashed line is for the Random Forests algorithm.

Forests algorithm. On one hand, data characteristics may be one factor that affect method performance. For example, the boundary shape of different classes. A good choice of kernel for SVM and careful tuning of the gamma parameter may adapt it to various data characters. We used the radial kernel and default value for gamma here. We did not fully explore all the possible choices because it is not our main focus here. There is a R package "caret" that developed a uniform front end to many classification and regression models. You can easily tune your parameters using this package to have improved performance. On the other hand, note that there are about $10,000$ features for each data set and about 10^{50} possible subspaces if using a subspace dimension 10. Among them, we only randomly selected 2,000 subspaces to construct our classifier. The Random Forests method possibly uses more features in constructing the final classifier with the same number of weak classifiers because it searches a new subspace for the best feature in each split and a fully developed tree can be very deep. From our past experience, the performance of the random subspace method would have significant improvement whenever a new pattern is identified in a new weak classifier. And the error rate usually does not change even when some poorly performed weak classifiers are incorporated because of the weight assignment procedure. It can be explained as follows: Assigning 0 weight to the newly generated weak classifier can at least retain the aggregated error rate of the original ensemble of weak classifiers. And the optimal weights determined by our weight assignment procedure should outperform the assignment giving 0 weight to the new weak classifier. Sometimes, the random subspace method may have a slightly increased error rate when certain new weak classifiers are incorporated. This is due to the inconsistent performance of the new weak classifier between the training set and the test set, resulting in more weight being incorrectly assigned to it. Since SVM generalizes well, the chance of this happening is small and the effect is limited. Therefore, the random subspace method has not reached its best performance with a small number of weak classifiers. But the Random Forests algorithm cannot guarantee its performance not being deteriorated by newly generated noisy trees.

As to the subspace dimension, our initial setting of a larger dimension (50) and fewer weak classifiers (500) did not result in a good performance of the weighted random subspace method. A large subspace dimension might have too many noisy features included that degrade the performances of the weak classifiers. Having too few weak classifiers would result in less chances to select any important pattern. The performance was improved only after we reduced the subspace dimension to 10 and increased the number of weak classifiers to 2,000. We expect that the performance of the random subspace method can be further improved if we have more weak classifiers included in the aggregated classifier.

In the above experiments, we used fixed subspace dimension size for the random subspace method. However, the size of the informative subspaces of the features is unknown in practice and needs to be tuned to have good performance. In addition, the underlying patterns may consist of different numbers of features. Ideally, we hope that all the weak classifiers are built on the selected subspaces that exactly capture those patterns. But in implementation, we may not be able to capture a pattern if the chosen subspace dimension is smaller than the true one or includes noisy features if the chosen subspace dimension is larger than the true one. In practice, we may want to vary the subspace dimensions in deriving the optimal classifier. For example, we can set the subspace dimension to be random to include subspaces of different dimensions. But the introduction of randomness may also induce a large number of uninformative weak classifiers.

All the above suggestions put a lot of requirement on computation speed for high dimensional data. High performance parallel computation may be a solution. It is especially suitable for aggregating algorithm. The R package "nws" can provide coordination and parallel execution facilities, which may be helpful for further improvement of the speed of weighted random subspace method.

5. CONCLUSION

In summary, we have proposed a weighted random subspace method for high dimensional data classification problems. We have also demonstrated the good performance of this method through the application to several widely-studied data sets.

ACKNOWLEDGEMENTS

This work was supported in part from NHLB/NIH contract N01-HV-28186, NIDA/NIH grant P30 DA 018343-01, and NIGMS grant R01 GM 59507.

Received 29 September 2008

REFERENCES

[1] Breiman, L. (1996a). Bagging Predictor. *Machine Learning* **24**, 123–140.

[2] Breiman, L. (1996b). Heuristics of instability and stablization in model selection. *Annals of Statistics* **24(6)**, 2350–2383. MR1425957

[3] Breiman, L. (2001). Random Forests. *Machine Learning* **45**, 5–32.

[4] Freund, Y. and R.E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science* **55**, 119–139. MR1473055

[5] Gerhold, D., M. Lu, J. Xu, C. Austin, and C. Caskey, et al. (2001). Monitoring expression of genes involved in drug metabolism and toxicology using DNA microarrays. *Physiological Genomics* **5(4)**, 161–170.

[6] Golub, T.R., D.K. Slonim, P. Tamayo, C. Huard, M. Gaaseenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science* **286**, 531–537.

[7] Gordon, G.J., R.V. Jensen, L.L. Hsiao, S.R. Gullans, J.E. Blumenstock, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, and R. Bueno (2002). Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Research* **62**, 4963–4967.

[8] Guyon, I. and A. Elisseeff (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**, 1157–1182.

[9] Ho, T.K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20(8)**, 832–844.

[10] Kleinberg, E.M. (1990). Stochastic discrimination. *Annals of Mathemathics and Artificial Intelligence* **1**, 207–239.

[11] Kleinberg, E.M. (1996). An overtraining-resistant stochastic modeling method for pattern recognition. *The Annals of Statistics* **24(6)**, 2319–2349. MR1425956

[12] Neumann, J., C. Schnörr, and G. Steidl (2005). Combined SVM-Based feature selection and classification. *Machine Learning* **61**, 129–150.

[13] Petricoin, E.F., A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, and L.A. Liotta (2001). Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet* **359**, 572–577.

[14] Veer, L.J., H. Dai, M.J. Van De Vijver, Y.D. He, A. Hart, M. MAO, H.L. Peterse, K. Van Der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, and S.H. Friend (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**, 530–536.

[15] Wu, B., T. Abbott, D. Fishman, W. McMurray, G. Mor, K. Stone, D. Ward, K. Williams, and H. Zhao (2003). Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics* **19(13)**, 1636–1643.

Xiaoye Li
Susquehanna International Group L.L.P.
401 City Avenue
Bala Cynwyd, PA 19004
E-mail address: xiaoye.li@sig.com

Hongyu Zhao
Department of Epidemiology and Public Health
Yale University
New Haven, CT 06520
E-mail address: hongyu.zhao@yale.edu