University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

8-2011

An adaptive ensemble learner function via bagging and rank aggregation with applications to high dimensional data.

Jasmit SureshKumar Shah
University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

Recommended Citation

Shah, Jasmit SureshKumar, "An adaptive ensemble learner function via bagging and rank aggregation with applications to high dimensional data." (2011). *Electronic Theses and Dissertations.* Paper 1306.
https://doi.org/10.18297/etd/1306

This Master's Thesis is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

AN ADAPTIVE ENSEMBLE LEARNER FUNCTION VIA BAGGING AND RANK AGGREGATION WITH APPLICATIONS TO HIGH DIMENSIONAL DATA

By

Jasmit SureshKumar Shah

B.S. in Mathematics and Statistics,

University of South Alabama, 2009

A Thesis
Submitted to the Faculty of the
School of Public Health and Information Sciences
University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Masters of Science in Biostatistics

Department of Bioinformatics and Biostatistics
University of Louisville
Louisville, Kentucky

August 2011

Copyright 2011 by Jasmit SureshKumar Shah

All rights reserved

AN ADAPTIVE ENSEMBLE LEARNER FUNCTION VIA BAGGING AND RANK AGGREGATION WITH APPLICATIONS TO HIGH DIMENSIONAL DATA

By

Jasmit SureshKumar Shah

B.S. in Mathematics and Statistics,

University of South Alabama, 2009

A Thesis Approved on

8/5/11

Date

By the following Thesis Committee

Thesis Director (Susmita Datta)

Somnath Datta

Ryan Gill

DEDICATION

This thesis is dedicated to my parents

Mr. SureshKumar Lakhamshi Shah

and

Mrs. Taruna SureshKumar Shah

and all my family members who have given me invaluable opportunities to fulfill my career.

ACKNOWLEDGMENTS

I would like to personally thank my thesis supervisor Dr. Susmita Datta for her continuous guidance and support throughout the testing and completion of my thesis. I would also like to thank the members of my graduate committee, Dr. Somnath Datta and Dr. Ryan Gill for their guidance and suggestions. I would also like to thank Late Dr. Satya Mishra, who was in continuous support and guidance in my Undergraduate career and made me be a confident person in going ahead and fulfilling my graduate career. My special thanks to Dr. Somnath Datta for allowing me to use his flowchart on the ensemble regression model for my thesis.

Furthermore, I want to thank my parents, brothers and sisters for their constant encouragement and support as I completed my Masters education in USA. Finally I would like to thank God for providing me with the direction and guidance throughout my life in USA.

ABSTRACT

AN ADAPTIVE ENSEMBLE LEARNER FUNCTION VIA BAGGING AND RANK AGGREGATION WITH APPLICATIONS TO HIGH DIMENSIONAL DATA

Jasmit SureshKumar Shah

August, 8th 2011

An ensemble consists of a set of individual predictors whose predictions are combined. Generally, different classification and regression models tend to work well for different types of data and also, it is usually not know which algorithm will be optimal in any given application. In this thesis an ensemble regression function is presented which is adapted from Datta et al. 2010. The ensemble function is constructed by combining bagging and rank aggregation that is capable of changing its performance depending on the type of data that is being used. In the classification approach, the results can be optimized with respect to performance measures such as accuracy, sensitivity, specificity and area under the curve (AUC) whereas in the regression approach, it can be optimized with respect to measures such as mean square error and mean absolute error. The ensemble classifier and ensemble regressor performs at the level of the best individual classifier or regression model. For complex high-dimensional datasets, it may be advisable to combine a number of classification algorithms or regression algorithms rather than using one specific algorithm.

TABLE OF CONTENTS

PAGE

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

ENSEMBLE PREDICTION MODELS FOR HIGH DIMENSIONAL DATA

INTRODUCTION

Ensemble is a method of combining a finite number of different types of predictors that are trained for the same purpose. Ensemble learning is one of the techniques that have been increasingly used to combine multiple algorithms to give better accuracy in making predictions. In the context of statistical problems, prediction methods fall into two categories: classification and regression (Indurkhyn and Sholom, 2001). For classification, the predicted output is a discrete number, a class, and performance is typically measured in terms of error rates. Whereas for regression, the predicted output is a continuous variable, and performance is typically measured in terms of distance, for example mean squared error or absolute distance (Indurkhyn and Sholom, 2001).

Ensemble of classifiers represents one of main research aspect in applied statistics and machine learning. The most popular ensemble methods are bagging, boosting and random forests. Mostly the classification of the ensemble is obtained by means of majority voting, where an unlabeled observation is assigned to the class with the highest votes among the individual classifiers' predictions. To explain the success of ensemble methods two main theories are considered (Valentini and Dietterich, 2004). The first theory considers the ensembles in the framework of large main classifiers showing that ensembles enlarge the margins, improving the generalization capabilities of learning algorithms (Valentini and Dietterich, 2004; Mason el al, 2000; Schapire et al, 1998). The second theory is based on the classical bias-variance decomposition of the error and it shows that the ensembles can decrease variance and bias (Valentini and Dietterich, 2004; Geman et al, 1992; Breiman et al, 1996; Kong et al, 1995). An ensemble method works most of time as the desired target function may not be implementable with individual models, but may be approximated by averaging. Here, the literature in general with the context of ensemble methods is reviewed. The main aim is that the ensemble as a whole will outperform any of the individual models for the given learning task. In this thesis the overall ensemble predictive model is created using the idea of bagging and rank aggregation. Out-of-bag samples play a very important role in the computation of the performance measures which are then aggregated over through the rank aggregation method to obtain the locally best regression model or the classification model.

LITERATURE REVIEW

The concept of boosting was introduced by Schapire (1990). This is a widely used ensemble method which was originally designed for classification problems but can also be extended to regression problems. Hansen and Salamon (1990) showed the advantages of bringing ensembles of similar neural networks. Perrone and Cooper (1993) presented a general framework for ensemble methods of better regression estimates. Breiman (1996) introduced the concept of bagging. Bagging is a name that was derived from bootstrap aggregation. This is a randomized algorithm based on the concept of bootstrapping. Bootstrapping is a sampling procedure that generates the random samples from the study sample with replacement. Bagging works mostly because as it takes the average of the multiple models, the variance is reduced. Freund and Schapire (1996) introduced AdaBoost. Boosting is where the final prediction is a combination of the predictions of multiple models. Larkey and Croft (1997) analyzed based on three different classifiers, K-nearest-neighbor, relevance feedback, and Bayesian independence classifiers. They concluded that the combination of the different classifiers produced better results than any single classifier. Ho (1998) showed the random subspace method for constructing decision forests. The method was shown to perform really well with larger data sets with huge feature variables. Opitz and Maclin (1999) compared bagging, AdaBoost and arching. They concluded that bagging is almost always more accurate than any single classifier and it is much less accurate than boosting. They also mentioned that the performance of boosting methods depends on the characteristics of the data set in use and further of their results suggested that boosting ensembles may over fit noisy data sets and thus decrease the performance. Dietterich (2000) compared three methods for constructing ensemble classifiers using randomization, bagging and boosting. The results show that boosting gives the best results in most cases. Randomization and bagging give similar results and also they suggest that randomizing is slightly better than bagging in low noise settings. Skuruchina and Duin (2002) compared bagging, boosting and the random subspace method to linear discriminant analysis (LDA). They concluded that all three methods may be useful in LDA but suggested that the efficiency is affected by the training sample size and the choice of classifiers. They finally concluded that bagging was useful in LDA for weak classifiers, boosting was useful in LDA only for the classifiers that perform bad on the large training samples and the random subspace method was useful in LDA for weak and linear classifiers obtained on small training samples. Valentini and Masulli (2002) presented an overview of ensemble learning, showing the main areas of research and why ensemble methods are able to outperform the single classifiers used within the ensemble. Topchy, Jain, and Punch (2004) considered combining weak clustering algorithms. They analyzed combination accuracy as a function of parameters, which control the power and resolution of component partitions. Chandra and Yao (2006) used a co-evolutionary framework to evolve new evolutionary ensemble learning algorithms. The framework treats diversity and accuracy as evolutionary pressures which are exerted at multiple levels of abstraction. Reyzin and Schapire (2006) stated that it is useful to consider boosting algorithms that maximize the average margin rather than the minimum one. Zhang and Zhang (2008) proposed a local boosting algorithm. The algorithm is based on the boosting by the resampling method. Their experimental results show that the local boosting algorithm performs better in most of the cases.

CHAPTER 2

ENSEMBLE CLASSIFIERS FOR LUNG CANCER CELL LINES

INTRODUCTION

Lung cancer is one of the most frequent causes of cancer deaths in North America. There are two main types of lung cancer, which are referred to as primary lung cancer which are non small cell lung cancer (NSCLC) and small cell lung cancer (SCLC). The classification of these two types of cancer is reproducible in approximately 90% of cases but the distinction between the two groups can be problematic when limited diagnostic material is available (e.g., from a fine needle aspirate) (Marchevsky et al, 2004). Molecular markers specific for the cancer types are more helpful and those based on DNA are more beneficial as they allow signal amplification by polymerase chain reaction (PCR) (Sozzi, 2001). A very promising alteration of DNA that is frequently found in cancer is DNA methylation (Marchevsky et al, 2004; Virmani et al, 2002). DNA methylation is an epigenetic event that affects cell function by altering gene expression and refers to the addition of a methyl group, to the 5-carbon of cytosine in a CpG dinucleotide. The CpG island is a short stretch of DNA in which the frequency of the CG sequence is higher than other regions and that "C" and "G" are connected by a phosphodiester bond. CpG dinucleotides are distributed unevenly across the human genome. CpG islands rarely exceed 5,000 base pairs and are often associated with functional elements. In particular, CpG islands overlap with the promoter regions of 50%

to 60% of human genes, including most housekeeping genes. CpG islands are usually found in the promoter regions of the genes and are usually not methylated in normal cells. The non-methylated state of promoter CpG islands is associated with transcriptional activity. The hypermethylated promoters lack the transcriptional activity that may account for gene inactivation both in normal physiological and disease states, remarkably the inactivation of the tumor suppressor genes in cancers (Yu et al, 2002).

The main aim of classification problems is to assign individuals to one of the identified classes based on their measurements. Usually in classification problems the datasets are divided into training and testing sets where the training sets are used to build the classifier which is then validated by the test sets. Another important aspect that characterizes classification of high-dimensional data is the need to obtain important variables. Variable importance involves in the identification of a subset of variables that are used to express the classification model. The main reason why variable selection is important is that removing the variables with less variability across observations gives better predictive accuracy. Classification algorithms can be used to process high dimensional data such as the cancer data to distinguish their disease subtypes. Class prediction is a method where the model learns from a set of individuals whose class subtypes are known in a training set which then creates a prediction rule to classify new individuals whose class types are not known in a test set. The class prediction method usually consists of three steps: selection of predictors; fitting the prediction model to create the classification rule; and performance assessment. The last step mainly assesses the performance of the prediction models. Accuracy, sensitivity, specificity, area under the ROC curve, positive predictive value, and negative predictive value are some of the primary criteria used in the assessment of the performance of a classification algorithm. Accuracy is the total number of correct classifications out of the total number of observations. The sensitivity is the proportion of the number of correct positive classifications out of the number of positives. The specificity is the proportion of the number of correct negative classifications out of the number of negatives. The area under the curve is one of the main characteristic of a receiver operating characteristic or simply and ROC Curve. ROC Curve is a graphical plot of the sensitivity vs. 1 – specificity.

However, due to complex and high dimensional data, it is difficult to use any single classification model that is reasonably flexible to keep the important variables, and yet feasible to fit. Since no single algorithm performs optimally for all the types of data, an ensemble classifier consisting of commonly used good individual classification algorithms is used which would adaptively change its performance depending on the type of data to that of the best performing individual classifier. Here we see how different classification methods might be applied to lung cancer diagnosis based on DNA methylation profiles, using the obtained methylation data from 87 lung cancer cell lines as a model system. We compare the utility of support vector machines (SVM), random forests (RF), linear discriminant analysis (LDA), Lasso Penalized Logistic Regression (PLR), Recursive partitioning (Rpart) and ensemble classifier (Datta et al, 2010) as classification tools of DNA methylation profiles, in an effort to develop models that can classify SCLC and NSCLC. For high-dimensional data, variable importance becomes a challenge as most classical methodologies fail to cope with high dimensionality, and so we then look at the variables important in classifying the data from the best classifier.

This chapter is organized as follows. In the Materials and Methods section, we describe the dataset used for the analysis and introduce some common classification algorithms. The Results section presents the data example (lung cancer) and the performance measures for the dataset. It is then followed by discussion and conclusion.

MATERIALS AND METHODS

DNA methylation is an epigenetic event that affects cell function by altering gene expression and refers to the addition of a methyl group, to the 5-carbon of cytosine in a CpG dinucleotide. The CpG island is a short stretch of DNA in which the frequency of the CG sequence is higher than other regions and that "C" and "G" are connected by a phosphodiester bond. CpG dinucleotides are distributed unevenly across the human genome. CpG islands rarely exceed 5,000 base pairs and are often associated with functional elements. In particular, CpG islands overlap with the promoter regions of 50% to 60% of human genes, including most housekeeping genes. MethyLight, a quantitative real-time PCR Technique is used for the measurement of DNA Methylation (Eads et al, 2000). The technology measures the frequency of molecules in which a series of CpG sites (usually ~8 sites) in a given CpG region are methylated. A data set consists of DNA measurements on a sample of N subjects at F CpG regions (features). The outcome is displayed in a $N \times F$ matrix where each row denotes a subject and each column a feature. DNA methylation profiles are collected from 87 lung cancer cell lines. The primary analysis of the data is described in a paper by Virmani et al. (2002). Cell lines were initiated by Gazdar et al. (1996) at the National Cancer Institute and Hamon Cancer Center. Three sets of primers and probes, designed for bisulfate- converted DNA, were used concurrently: a methylation- specific set for the gene of interest and two reference sets to normalize for input DNA (Virmani et al, 2002). We want to demonstrate whether DNA methylation profiles could distinguish between two subtypes of lung cancer, non-small cell (NSCLC) and small cell (SCLC). The analysis is limited to a subset of seven CpG regions that showed the best discrimination between SCLC and NSCLC (Virmani et.al, 2002). It was established that each of the seven CpG regions was predictive of lung cancer subtypes. We want to study the classification performance of several classification algorithms including the ensemble classifier (Datta et al, 2010) in this experimental data. We also want to find the relative importance of the features (CpG Islands) in order to best classify the samples into two cancer subtypes.

SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) were introduced as a machine learning method by Cortes and Vapnik (1995) and has since attracted a high degree of interest in machine learning community. It applies the simple linear method to the data but in a high dimensional feature space non-linearly related to the input space. If given a two-class training set, SVMs assigns its data in a higher dimensional space and attempts to specify a maximum-margin separating hyperplane between the data of two classes. This hyperplane is ideal in the sense that it generalizes well to unobserved data. The training input of SVMs involves of data that are vectors of real numbers. Given a set of training samples $\{(x_i, y_i)\}$ with the data x_i contained in the d dimensional Euclidean space of a set of real numbers and the corresponding class type y_i in {-1, +1}. In SVMs, the hyperplane classifies all the training samples correctly. In a two dimensional space this hyperplane is a line whereas in a three dimensional space this hyperplane could be a plane. The hyperplane is constructed with the largest possible margin:

$$f(x) = wx + b$$

To separate the two classes $wx + b = 0$ is needed to be found, where w is the weight vector in the feature space while b is the bias. Figure 2.1 shows a hyperplane that seperates the two classes and it shows the distance between the hyperplane and its nearest vectors.

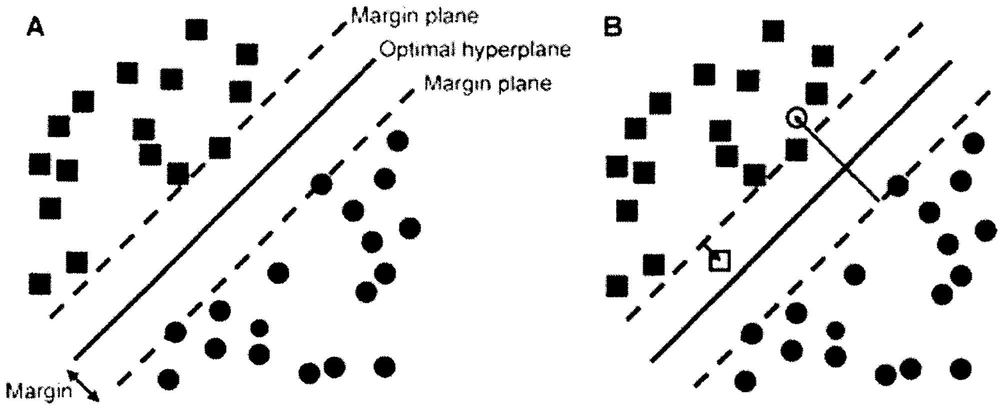


Figure 2.1: Separating hyperplane of the Support Vector Machine that maximizes the margin between two sets of perfectly separable objects, represented as circles and squares. (A) Optimal hyperplane that perfectly separates the two classes of objects. (B) Optimal soft margin hyperplane which tolerates some points (unfilled square and circle) on the "wrong" side of the appropriate margin plane. Reproduced by kind permission of the authors., Jorissen, R. N.; Gilson, M. K. Virtual Screening of Molecular Databases Using a Support Vector Machine. *J. Chem. Inf. Model.* 2005, 45, 549-561

To ensure that all the training samples are classified correctly, the following equation must hold for the nearest samples and thus the hyperplane can be computed.

$$wx + b = \begin{matrix} \geq +1 \; if \; y_i = +1 \\ \leq -1 \; if \; y_i = -1 \end{matrix}$$

SVMs are excellent examples of supervised learning that tries to maximize the generalization by maximizing the margin and also supports nonlinear separation using advanced kernels, by which SVMs try to avoid overfitting and underfitting (Yu et al, 2003; Burges, 1998; Vapnik, 1998).

RANDOM FORESTS

The Random Forest algorithm was proposed by Leo Breiman in 1999 (Breiman, 2001). It is an extension of the CART (1983) to a group of trees. CART, known as Classification and Regression Trees was developed by Breiman and his colleagues (Breiman et al, 1983). In CART the root node contains all observations and every node is divided into two further nodes depending on a true-false answer to a question, until the same node is homogeneous with the cases. CART is easy to use and interpret and the classification accuracy is low. Random Forest as compared to CART gives higher classification accuracy. Random Forests uses 63% of the samples to construct each tree and the remainder 37% samples comprise out-of-bag samples (O-O-B) which are used to evaluate the performance of each tree. A large number of trees are produced by the random forest and together these trees vote for the most popular class. When each tree is grown, the model randomizes the search for the best split in each leaf. The model tries to find good trees through a randomized search, and then averages the predictions across the good trees. Suppose M trees are constructed, and the prediction of tree i at the feature vector x is $g_i(x)$, then the random forest prediction is: $(g_1(x) + g_2(x) + \ldots + g_m(x))/m$. Usually with decision trees, the predicted class labels can be obtained by choosing the class with the largest prediction score. Two types of randomization methods are used for growing a random forest. One is the bootstrap method, where sampling with replacement is repeated over and over again to produce many trees. This is known as bagging which was proposed by Breiman (1996). The second method is randomized tree growing. In randomized tree growing, each leaf of each tree is grown using a subset of all the variables chosen at random, and the best split among these variables is chosen. Each tree generates predictions; the average is taken overall these trees.

The two randomized methods give two tuning parameters to decide an appropriate model: One is the number of bootstrap samples to be drawn and the other is the number of variables to be used in each tree. The number of variables to be used in each tree can be optimized by out-of-bag (O-O-B) performance. Usually 37% of the data is used as the O-O-B observation and are not used for training the model and so the O-O-B observation can be used to validate the training model by getting an estimate of error rate for each of the tree.

LINEAR DISCRIMINANT ANALYSIS

Discriminant analysis was first developed by R. A. Fisher in 1936 which is a multivariate method of classification. It is similar to regression analysis except that the dependent variable is categorical. In discriminant analysis, the objective is to predict class of the individual observations based on the predictor variables. LDA usually tries to find linear combinations of predictor variables that separate the groups of observations, and these combinations are known as discriminant functions. Suppose K different groups are given, each is assumed to have a multivariate normal distribution with mean vectors

$\mu_k$and a common covariance matrix $\sum$. The idea in LDA is to classify observations x_i to the group k.

$$k = argmin\ (x_i - \mu_k)^T \textstyle\sum^{-1}(x_i - \mu_k)$$

Generally the prior probability can be estimated using the proportion of the number of observations in each group to the total number of observations. Instead of maximizing the likelihood, the posterior probability is maximized. In the case, when the assumption of the covariance matrix common in all groups is not satisfied, an individual covariance matrix for each group is used, thus leading to Quadratic Discriminant Analysis. Discriminant functions are found based on the assumptions of homogeneity of covariance between groups and multi-normality in each group. The discriminant functions in a binary case are built linearly as follows:

$$d_1(x) = x^T \textstyle\sum^{-1}\mu_1 - \frac{1}{2}\mu_1^T \sum^{-1}\mu_1 + log\pi_1$$

$$d_2(x) = x^T \textstyle\sum^{-1}\mu_2 - \frac{1}{2}\mu_2^T \sum^{-1}\mu_2 + log\pi_2$$

If $d_1(x) > d_2(x)$, the observation is assigned to group one, otherwise the observation is assigned to group two.

LOGISTIC REGRESSION

Logistic regression provides a good method for classification by modeling the probability of membership of a class with transforms of linear combinations of explanatory variables. It is a well known method used for determining the relation between the feature and the response variables. When the response variable is binary, logistic regression models are similar to multiple linear regression methods. The simple logistic model has the form

$$logit(Y) = ln\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta X\pi = P(Y = \text{Outcome}|X = x) = \frac{e^{\alpha+\beta X}}{1+ e^{\alpha+\beta X}}$$

Given two classes in a dataset, we are interested in modeling the probabilities of the two classes using a linear function of variable x.

$$\log\left[\frac{P\ (Y = 1|x)}{(1 - P(Y = 1|x))}\right] = \beta_0 + \sum_{i=1}^{p} \beta_j x_i$$

When the number of features is larger than the number of samples, feature selection is performed to reduce the dimensionality of the dataset. Another way is to use a penalized logistic regression (PLR) where a penalty is imposed on the log likelihood function corresponding to the general logistic regression. The penalized log likelihood function may be written as follows

$$l_p(\beta_0; \beta; \lambda) = -l(\beta_0; \beta) + \lambda J(\beta)$$

Where λ is the regularization parameter controlling the amount of shrinkage and $J(.)$ is a penalty function on the parameter β.

ENSEMBLE CLASSIFIER VIA BAGGING AND RANK AGGREGATION

This classification method was originally proposed by Datta et al, 2010, which is a combination of bagging and rank aggregation in a single procedure. Bagging reduces the variance and improves the accuracy of weak classifiers. Weak classifiers are defined as classifiers whose final predictions change drastically with little changes to training data (Datta et al, 2010). For every bootstrap sample (sampling with replacement) several classifiers are trained and a classifier with the best performance on out-of-bag samples are kept for predicting the testing data. The weighted rank aggregation is used for multi-objective optimization, where more than one performance measure is required. Each performance measure ranks the algorithm according to the performance and the ordered lists of algorithms are then aggregated to produce a single list which ranks algorithms according to the performance. The algorithm below is a step- by- step procedure on how the ensemble classifier is built. Assuming we have a training data consisting of n samples with the vector form $\{X_{(n\times p)}, y_{(n\times 1)}\}$.

1. Initialization. Set the number of bootstrap samples to draw. Let $j\ =1$, select M classification methods along with K different performance measures to be optimized.

2. Sampling. Draw a bootstrap sample of the same size from the training samples using simple random sampling with replacement so that we can obtain $\{X_j^*, y_j^*\}$. Sampling is repeated until the samples from all the classes are present in the training set. Some samples will be repeated more than once while others will be left out of the bootstrap sample, and such samples are called out-of-bag (O-O-B) samples.

3. Classification. Using the bootstrap samples, M classifiers are trained.

4. Performance assessment. The M models fitted in the classification step are used to predict the classes of the O-O-B samples. Since we know the true classes of the samples, K different performance measures can be computed. Each measure will rank the classification algorithm according to the performance of the algorithm under the particular measure, producing K ordered list of size M, $L_1, \dots, L_K$.

5. Rank Aggregation. Once we obtain the ordered lists from the performance measures, they are rank aggregated using the weighted rank aggregation procedure which determines the best classification algorithm.

Steps 2-5 are repeated many times say N times.

6. Predictions. Using the N best individual models, built on the training data for each bootstrap sample, N class predictions for each sample is made.

Given a new sample $x_{(p\times 1)}$, let $\widehat{y_1}, \dots, \widehat{y_N}$ denote N class predictions from the N classifiers. The final classification is based on the most frequent class in the N predicted class labels. Figure 2.2 shows the flowchart of both building the ensemble classifier as well as using it to predict new samples.

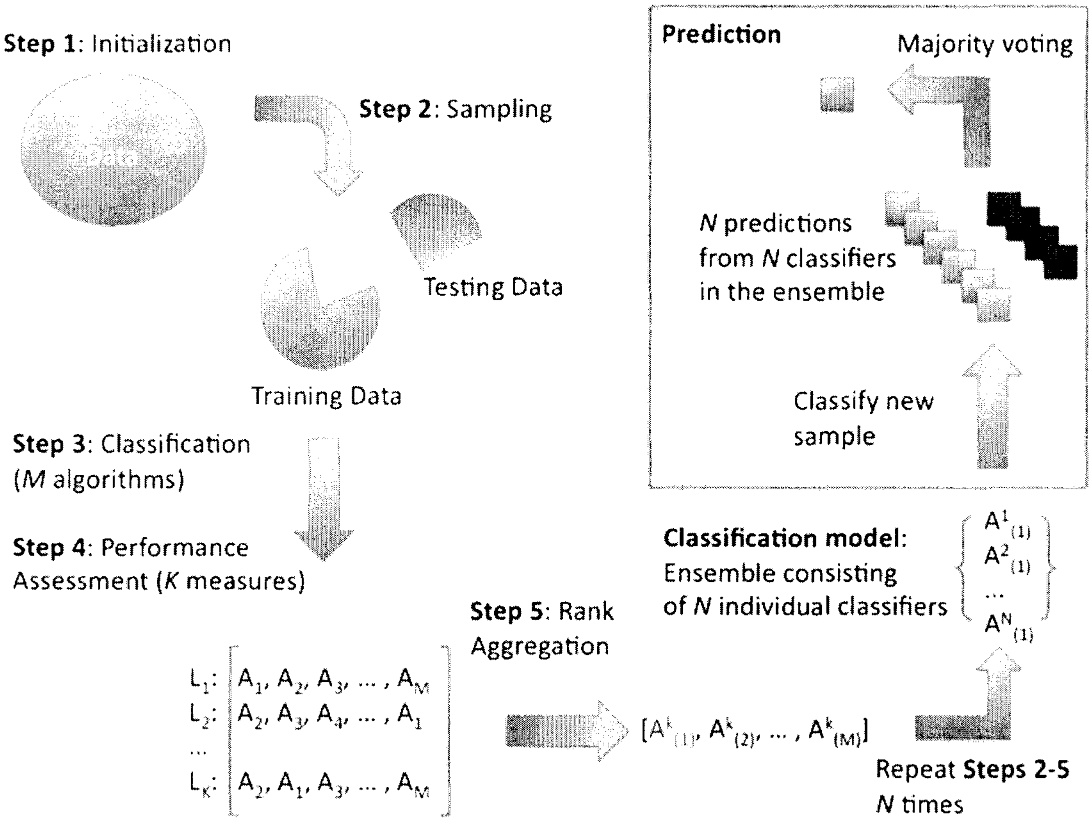

Step 1: Initialization
Data
Step 2: Sampling
Testing Data
Training Data
Step 3: Classification (M algorithms)
Step 4: Performance Assessment (K measures)
L_1: $A_1, A_2, A_3, \ldots, A_M$
L_2: $A_2, A_3, A_4, \ldots, A_1$
...
L_K: $A_2, A_1, A_3, \ldots, A_M$
Step 5: Rank Aggregation
$[A^k_{(1)}, A^k_{(2)}, \ldots, A^k_{(M)}]$
Repeat Steps 2-5 N times
Classification model: Ensemble consisting of N individual classifiers
$A^1_{(1)}$, $A^2_{(1)}$, ..., $A^N_{(1)}$
Prediction
Majority voting
N predictions from N classifiers in the ensemble
Classify new sample

Figure 2.2. A schematic flowchart for the classification problem. Reproduced by kind permission of the author, Datta et al. An adaptive optimal ensemble classifier via bagging and rank aggregation with applications to high dimensional data.2010: BMC Bioinformatics; 11:427.

VARIABLE IMPORTANCE

Classifying high-dimensional data is a difficult problem due to the large number of variables involved. Variable importance therefore becomes a challenge due to high dimensionality. With reducing the dimensions of the data allows the performing measures to give better classification of the data. In Random Forests, Breiman proposed a permutation-based variable importance measure (Breiman, 2001). To access the importance of a certain variable, Breiman proposes to permute the variable values in the out of bag samples randomly, and then to classify the out-of-bag samples with one permuted variable. We are using rank aggregation for feature selection. The mathematical problem of rank aggregation was first proposed by Dwork et al, 2001. For variable selection using rank aggregation please refer to Datta et al, 2010.

RESULTS AND DISCUSSION

The data consists of 87 samples, 41 SCLC and 46 NSCLC cell lines samples. Four individual classification algorithms were selected with the number of bootstrap samples equal to 101. An external cross validation was implemented and the scores are listed in the table below. The samples were divided into training and testing data sets each consisting of 46 and 41 samples respectively. 100 different training and testing data sets were created from the 87 samples randomly. SVM and the ensemble classifier give the best accuracy measure whereas SVM and random forest gives the best AUC measure.

	Accuracy	Sensitivity	Specificity	AUC
Random Forest	0.6538	0.6682	0.7647	0.8233
SVM	0.8063	0.7778	0.8325	0.8399
LDA	0.7307	0.7670	0.7447	0.7624
PLR	0.6538	0.7256	0.7216	0.7196
Ensemble	0.8125	0.7977	0.8824	0.8089

Table 2.1: Averages of cross validation for the cancer data. The number of bootstrap samples was $N = 101$.

From the above table, it shows Random Forest, SVM and the Ensemble Classifier; perform better in classifying the data. The Ensemble classifies in the best for accuracy, sensitivity, specificity. In AUC it is close to the SVM which is the best performing individual classifier. The list below shows the order of the variables from most important to least important: APC, ESR1, CALCA, MTHFR, MYOD1, PTGS2, MGMTM1.

CONCLUSION

Classifying high-dimensional data is a difficult problem due to the large number of variables involved. Variable importance therefore becomes a challenge due to high dimensionality. With reducing the dimensions of the data allows the performing measures to give better classification of the data. For the data considered here, the performance measures considered (accuracy, sensitivity, specificity), the ensemble and the random forest classifier perform the best compared to the other methods. For generating the bootstrap samples simple random sampling was used. Some bootstrap samples do not include all the classes. We have used some common classification methods and also dimension reduction techniques. Also, the performance measures used are the common ones and there are other measures such as the Brier score (Brier, 1950) or the Kappa statistic (Galton, 1892) that are available.

CHAPTER 3

ENSEMBLE REGRESSION FOR HIGH DIMENSIONAL DATA

INTRODUCTION

Cancer is a main public health problem in most parts of the world. Lung cancer represents the main cause of cancer-related deaths in Western countries. The overall 5-year survival rate is 16% and has been the same rate over many decades. The main reason for the cancer to be a leading cause is due to the discovery at the advanced stages. Most patients at the early stage discovery are treated primarily by surgery but 30-60% will develop and die of metastasis recurrence.

Lung cancer is further classified according to the histological criteria. The four main subtypes of lung cancer are: small cell lung cancer (SCLC), squamous cell carcinoma (SC), adenocarcinoma (AC), large cell carcinoma (LC). The last three subtypes are categorized as non-small cell lung cancer (NSCLC), and accounts for about 85% of all the lung cancers. Accurate classification and diagnosis of the cancer is very crucial for the selection of the appropriate medical therapies.

Proteomics is likely to play a key role in cancer biomarker discovery. Despite, a lot of attempts in searching for biomarkers using various methods no biomarker with 100% diagnostic accuracy have been established for any cancer type (Karpova et al, 2010). Due to the heterogeneous nature of the cancer, existence of such biomarkers is not easy. Most efforts are therefore concentrated on searching for panels of differentially expressed proteins/peptides instead of individual biomarkers and building of diagnostic methods based on numerous features (Karpova et al, 2010; Skates et al 2004). Although it has become feasible to rapidly analyze proteins from crude cell extracts using mass spectrometry, sample complexity complicates these studies (Gamez-Pazo et al, 2009). Thus, for effective proteome analysis it is important to enrich samples for the analytes of interest (Hanash, 2003). Despite the fact that one-third of the proteins in eukaryotic cells are assumed to be phosphorylated at some point in their life cycle, only a low percentage of the intracellular proteins is phosphorylated at any given time (Cohen, 2002; Makrantoni et al, 2005). Thus, a purification or enrichment step that isolates phosphorylated species would reduce complexity and increase sensitivity (Oda et al).

Mass spectrometry for proteomics consists of many different platforms and is used to profile the serum peptidome. Magnetic bead-assisted serum peptide capture coupled to matrix assisted laser desorption/ ionization time-of-flight MS (MALDI-TOF-MS) is a serum peptide profiling strategy gaining in popularity compared to surface enhanced laser desorption/ ionization (SELDI)-based platforms due to superior resolution of MALDI instruments, the possibility to obtain structural (MS/MS) information of signature peptides and superior binding capacity of the magnetic beads compared to a flat SELDI chip surface (Voortman et al, 2009; Jimenez et al, 2007). MALDI-TOF has been widely used in cancer investigation. A typical dataset from a Mass Spectrometer consists of hundreds of spectra; each spectrum contains of thousands of intensity measurements representing an unknown number of protein peaks which are the key features of interest.

In either SELDI- TOF or MALDI-TOF, we obtain from a biological sample a calibrated output which is a mass spectrum characterized by several peaks, which relate to individual proteins or protein fragments (polypeptides) present in the sample. The heights of the peaks represent the intensities of ions in the sample for a specific mass to charge ratio (m/z) value. The heights along with the m/z values characterize the fingerprint of the sample. Therefore, detecting location and amplitude of common peaks from a set of spectra is a way to recognize specific biomarkers that can be used to characterize patients and to compare the groups of the patients. A huge amount of data is produced to be analyzed and create a need for a rapid and efficient method for comparing multiple MS spectra. Raw spectra acquired by TOF mass-spectrometers are usually a mixture of a real signal, noise of different characteristics and a varying baseline. Statistically, a likely model for a given mass spectrometry (MS) spectrum is to denote it schematically by the equation:

$$Y\left(\frac{m}{z}\right) = B\left(\frac{m}{z}\right) + NS\left(\frac{m}{z}\right) + \epsilon\,(\frac{m}{z})$$

Where $Y\left(\frac{m}{z}\right)$ is the observed intensity of the spectrum at mass to charge ratio m/z, $B\left(\frac{m}{z}\right)$ is the baseline representing a relatively smooth artifact commonly seen in mass spectrometry data, $S\left(\frac{m}{z}\right)$ is the true signal of interest consisting of a sum of possible overlapping peaks, N is a normalization factor to adjust for possibly differing amounts of protein in each sample, and $\epsilon\left(\frac{m}{z}\right)$ is an additive white noise with variance σ_{ϵ}^{2} arising from the measurement process (Antoniadis et al, 2010). Pre-processing of the mixed data is therefore important to extract $S\left(\frac{m}{z}\right)$ which is the signal of interest. Incorrect preprocessing methods can result in data sets that show substantial biases and make it difficult to reach significant conclusions. The main preprocessing steps used are baseline correction and denoising of the data. The software used was proposed by Ndukum et al (2011). Baseline subtraction uses an algorithm to eliminate the baseline slope and offset from a spectrum by interactively calculating the best-fit straight line through a set of estimated baseline points (Ndukum et al, 2011).

However, due to complex and high dimensional data, it is difficult to use any single regression model that is reasonably flexible to keep the important features, and yet feasible to fit. Since no single algorithm performs optimally for all the types of data, an ensemble regressor consisting of commonly used good individual regression algorithms is used which would adaptively change its performance depending on the type of data to that of the best performing individual regressor. Here we want to predict the survival times of patients from proteomic profiles using MALDI- TOF Mass Spectrometry data. The outcome of interest is progression free survival at the end of treatment. The formula is given as:

$$\log Y_i = X_i\beta + \varepsilon_i$$

where Y_i are the survival times of the patients and X_i are the intensities of the proteomic features. We see how different regression methods might be applied to lung cancer diagnosis based on proteomic features. We compare the utility of least absolute shrinkage and selection operator (LASSO), partial least squares (PLS), sparse partial least squares (SPLS), principal component regression (PCR), and the ensemble regressor adapted from the ensemble classifier (Datta et al, 2010). The ensemble regressor model is created in a highly adaptive manner, which is a nonlinear predictive model that is multi-objective in nature which optimizes the prediction power for a number of features. In a prediction analysis, we are interested in fitting different models to capture the relationship between independent variables X and a dependent variable Y, and then using the models to make predictions on an independent dataset. Prediction analysis mainly focuses on prediction errors and these are error measures in the estimated period. Examples of such error measures can be mean squared error, mean absolute error, mean

absolute percentage error, and mean percentage error. The Mean Squared Error (MSE) is a measure of how close a fitted line is to the data points. For every data point, you take the distance vertically from the point to the corresponding y value on the curve fit (the error), and square the value. Then you add up all those values for all data points, and divide by the number of points. The squaring is done so negative values do not cancel positive values. The smaller the Mean Squared Error, the closer the fit is to the data. The Mean Absolute Error is a quantity used to measure how close predictions are to the eventual outcomes. It is an average of the absolute errors $e_i = |f_i - y_i|$, where f_i is the prediction and y_i the true value.

MATERIAL AND METHODS

We utilize the data set reported in Voortman et al (2009). The MALDI-TOF-MS dataset of serum samples of 27 patients with advanced non-small cell lung cancer (NSCLC) were treated with first line chemotherapy, consisting of ciplastin and gemcitabine, as well as bortezomib. The efficacy of ciplastin-gemcitabine alone is limited, a partial tumor response being attained in about one third of NSCLC patients with a median progression free survival of four to five months (Voortman et al, 2009; Smit et al, 2003). Serum spectra of these 27 patients are available at three different time points: pre-treatment (preTX), after two cycles of treatment (post-2), and at the end of treatment (EOT). For each patient, there is an associated progression free survival time recorded in days (PFS). There is no censoring available in the data and the range of observed survival time is from 27 days to 601 days. We want to study the regression performance of several multivariate regression models including the ensemble regression function in this experimental data.

We compare the utility of Least Absolute Shrinkage and Selection Operator (LASSO), Partial Least Squares (PLS), Sparse Partial Least Squares (SPLS) and Principal Component Regression (PCR). With the above models we develop an ensemble model that comprises of all the models taken individual and compare the prediction results of the ensemble model to the individual models.

LASSO

Shrinkage methods are attractive in modeling and predictive learning because they allow continuous shrinkage with small generalization error, and they are usually easy to solve in practice. LASSO, proposed by Tibshirani (1996), is a variable selection technique which allows shrinkage of the coefficients while setting some of the coefficients to zero. The LASSO tries to find a model which minimizes the residual sum of squares subject to a constraint that the sum of the absolute values of the coefficient for each variable is less than a constant, say c. Suppose that a linear regression model is given in the form below:

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \varepsilon ,$$

where X_j is the jth variable, Y is the response vector, β_0 is the intercept, β_j is the coefficient of the jth variable, p is the total number of variables taken, and ε is the random errors vector that are assumed to be independently identically distributed (i.i.d.) with a normal distribution with mean zero and variance σ^2. The LASSO estimate of $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$ is given by the following formula:

$$\hat{\beta} = \arg min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \right\}, \text{subject to} \sum_{j=1}^{p} |\beta_j| \leq c,$$

where $c \geq 0$ is a tuning parameter that controls the shrinkage applied to the estimates. The constraint also allows the removal of the variables from the model by setting their coefficients to zero. The value of c ranges from zero to

$$t_{max} = \sum_{j=1}^{p} |\hat{\beta}_j{}^0| \, ,$$

Where $\hat{\beta}_j{}^0$ is the ordinary least squares estimate of β_j. The optimal choice of c is solved by the normalized parameter $s = t/t_{max}$, which ranges from zero to one.

PARTIAL LEAST SQUARE

Partial Least Squares (PLS) regression was introduced by Wold (1966), and has been used as an alternative approach to the Ordinary Least Squares (OLS) regression. As PLS utilizes the dimension reduction principle, it can handle a large number of variables (p) with a small sample size (n). PLS has a modeling aspect that relates the modeling to two data sets, X , matrix of the variables/ covariates and Y, vector of responses. At the basics of PLS regression is a dimension reduction technique that functions under the assumption of a basic latent decomposition of the response matrix ($Y \in R^{n\times q}$)and the predictor matrix($X \in R^{n\times p}$):

$$Y = TQ^T + F,$$

$$X = TP^T + E,$$

where $T \in R^{n\times K}$ is a matrix that produces K linear combinations; $P \in R^{p\times K}$ and $Q \in R^{q\times K}$ are the matrices of the coefficients and $E \in R^{n\times p}$ and $F \in R^{n\times q}$ are the matrices of the random errors. For specification of the latent component matrix T such that $T = WX$, PLS requires finding the columns of $W = (w_1, w_2, \dots, w_K)$ from successive optimazion problems.

Several iterative procedures have been proposed to solve nonlinear optimization problems such as PLS Mode A, PLS-SB, NIPALS and SIMPLS algorithms that vary by the deflation theme required for the orthogonally of the derived components. PLS Mode A algorithm (Kong et al, 1995) targets to model existing relationships between variables rather than to model for prediction. PLS-SB calculates all eigenvectors at once, and the score vectors obtained by this method are not necessarily orthogonal. The commonly used methods, NIPALS and SIMPLS, involve two steps may be called graduation (deriving components) and prediction.

The NIPALS algorithm (Jemal et al, 2007) was established as an alternative to principal component algorithms. NIPALS employs sequential simple linear regressions instead of singular value decomposition to calculate principal components. PLS algorithm can be considered as carrying out two simultaneous NIPALS principal component analyses, one for X and one for Y, while interchanging the results from X for analysis of Y and to solve the following maximization problem

$$\max_{||r||=||s||=1} \mathrm{cov}(X_r Y_s)^2$$

under the orthogonality constraint of derived components, where $s = 1$ and $Y = y$ for univariate model. Since both X and Y are used in the calculation of the components,

PLS is presented as a member of the bilinear class of methods and the bilinear model can be written as:

$$Y = TQ^T + F,$$

$$X = TP^T + E,$$

It is assumed that the score matrix T is a good predictor for Y and a linear, inner relationship between the score matrices T and U exists, i.e. $U = T\,B + H$ where B is a $k \times k$ diagonal matrix and H is a matrix of errors. The mixed relation then becomes:
$Y = UQ' + F = (TB + H)Q' + F = TA' + F^*$,
where $A' = BQ'$is a matrix of regression coefficients and $F^* = HQ' + F$ is matrix of errors.

SIMPLS algorithm (Schiller et al, 2002) is an alternative to NIPALS algorithm that targets to derive PLS components directly in terms of the original data which results in faster computation with less memory requirements. SIMPLS reduces the cross-covariance matrix, $S_{xy} \propto X'Y$, whereas NIPALS reduces the original data matrix X to obtain orthogonal components.

SPARSE PARTIAL LEAST SQUARES

With recent advancement in biotechnology such as high throughput sequencing, regression based modeling of high dimensional data has never been that important. Two most important problems that arise within the regression problems is the selection of the important variables and covariates being highly correlated with the data sample size much smaller than the variables. Sparse Partial Least Squares is based upon the PLS. It is a new technique that combines and generalizes the strength of principal component analysis and multiple regression.

Suppose there exist a latent component $T_{n\times k}$ such that

$X = TP^T + E,$

$Y = TQ^T + F$

where $X_{n\times p}$is a predictor variable and $Y_{n\times q}$ is the response variable, $P_{p\times k}$ and $Q_{q\times k}$ are the coefficient matrix, $E_{n\times p}$ and $F_{n\times q}$ are the errors. From the $X\ and\ Y$ equations, we suppose there exists a director matrix W such that $T = WX$, the usual way for finding the latent components T is by finding the direction columns of a director matrix $W = (w_1, w_2, \dots, w_k)$ by solving many optimization problems. If the response variable Y is univariate, then the kth direction vector w_k can be obtained by solving the constrained optimization problem

$$w_k = \arg max_w \ \{\rho_{Y}^2, X_w var(Xw)\} \text{ with } w^T w = 1, w^T \sum XX w_j = 0$$

for $j = 1, \dots, k-1$, where $\sum XX$ represents the covariance of X.

When the response Y is multivariate, SIMPLS or NIPALS can be used to find the direction vectors. SIMPLS was proposed by de Jong (de Jong, 1993) which directly uses the univariate PLS formula. The SIMPLS formula is as below:

$$w_k = \arg max_w \ \{w^T \sigma_{XY} \sigma_{XY}^T w\} \text{ with } w^T w = 1, w^T \sum XX w_j = 0$$

for $j = 1, \dots, k-1$, where σ_{XY} represents the covariance of X and Y. The other formula of NIPALS was proposed by Wold (1966) but the specific formula of the direction vector was not given and later on Tar Braak and de Jong (ter Braak and De Jong, 1998) gave the following formula and proved that the direction vector obtained by the formula are exactly what solved by using the NIPALS algorithm.

$$w_k = \arg max_w \ \{w^T \sigma_{XY} \sigma_{XY}^T w\} \text{ with } w^T (I_p - W_{k-1} W_{k-1}^{-1}) w = 1, w^T \sum XX w_j = 0$$

for $j = 1, \dots, k-1$, where I_p is a $p \times p$ identity matrix and W_{k-1}^{-1} is a unique Moore-Penrose inverse of W_{k-1}.

For different response Y, the corresponding latent components T can be obtained, and the coefficient matrix Q can be estimated by solving $min_Q ||Y - TQ^T||_2$. Once the latent components and the coefficient estimators $\hat{Q}^T$ are obtained, the final model's parameters can be estimated via $\hat{\beta} = \widehat{W}_K \hat{Q}^T$ and the final model is $Y = \hat{\beta} X$. A threshold for $\hat{\beta}$ was proposed by Huang et al (2004) via adding sparse constraint to the procedure of finding $\widehat{Q}$. Chun and Keles proposed sparse partial least square by imposing the sparsity constraint in the process of dimension reduction. In SPLS dimension reduction and variable selection is performed at the same time and is equivalent to solving the following constrained problem;

$$min_{w,c}\{kw^TMw + (1-k)(c-w)^TM(c-w) + \lambda_1|c|_1 + \lambda_2|c|_2^2\}, \text{with } w^Tw = 1,$$

where $M = X^TYY^TX$. In the equation above c is a surrogate of the original direction vector w.

PRINCIPAL COMPONENT REGRESSION

Principal component analysis (PCA) is a multivariate technique in which a number of correlated variables are handled through a linear transformation into a set of uncorrelated variables. This method is primarily a data analyzing technique that obtains linear transformations of a group of correlated variables such that certain optimal conditions are met (Jackson, 1991). The most important of these conditions is that the transformed variables are uncorrelated. Correlation of variables is essentially an indication of the strength and direction of a linear relationship between two variables (Weisberg, 1980) and it must be considered if redundant data is to be acknowledged and excluded.

PCR is a two-step process, which first uses PCA then applies a multivariate linear regression (MLR) procedure. This second step regresses the newly acquired data with the response variable. The objective of principal components analysis is to find a linear transformation of a set of n variables of X into a new set denoted by H, where the new set has certain necessary properties. These properties, which provide the rationale for using the H rather than the original X are: (i) the elements of H are uncorrelated with each other in the sample; and (ii) each element of H, progressing from $H_1 \, to \, H_2$ etc., accounts for as much of the combined variance of the X as possible, steady with being orthogonal to the preceding H. The new variables correspond to the principal axes of the ellipsoid formed by the scatter of sample points in the n dimensional space having the elements of X as a basis. The principal components transformation is thus a rotation from the original x coordinate system to the system defined by the principal axes of this ellipsoid. PCA is a useful method to solve problems including exploratory data analysis, classification, pattern recognition, and noise reduction, for example. It is used whenever uncorrelated linear combinations of variables are wanted which reduces the dimensions of a set of variables by reconstructing them into uncorrelated combinations. It combines the variables that explain for the largest part of the variance to form the first principal component. The second principal component explains for the next largest amount of variance, and so on, until the complete sample set variance is combined into smaller uncorrelated component categories. Each successive component explains portions of the variance in the total sample and all of the components are uncorrelated with each other.

Consider a data matrix X having N rows and M columns. Let $X_1, X_2, \dots, X_M$ be the variables. PCA is the fundamental technique for dimension reduction based on the principle of singular value decomposition of the data matrix. PCA relates to the second statistical moment of X, which is proportional to XX' and it partitions X into two matrices H and C. Each attribute can be expressed as a linear combination,

$$X_j = H \times C_j$$

where $H = (\psi_1, \psi_2, \ldots, \psi_N)$ is an $N \times N$ matrix of basis vectors and C_j is a $N \times 1$ column vector of weights related to jth variable. For the defined $N \times M$ data matrix $X = (X_1, X_2, \ldots, X_M)$, the observation model can be written in the form,

$$X = H \times C$$

where $C = (C_1, C_2, \ldots, C_M)$ is a $N \times M$ matrix of weights.

Another important point in the use of model is the choice of the basis vectors ψ_n. Many different ways to select these basis vectors exist, of which one is the principal component regression. In PCR, the basis vectors are selected to be the eigenvectors v_n of either the data covariance or correlation matrix. The correlation matrix can be estimated as,

$$R = \frac{1}{M} \times X \times X',$$

The eigenvectors and the eigenvalues can be solved from the eigen decomposition. The eigenvectors of the correlation matrix are orthonormal, and therefore, the ordinary least-squares solution for the parameters C becomes,

$$\hat{C}_{PC} = H' \times X$$

and the attribute estimates could be computed from,

$$X_{PC} = H \times \hat{C}_{PC}{}'$$

Matrix H contains the eigenvectors of R ordered by their eigenvalues with the largest first and in the descending order. The first column of H gives the direction that minimizes the orthogonal distances from the samples to their projection onto this vector.

ENSEMBLE REGRESSOR VIA BAGGING AND RANK AGGREGATION

This regression method is adapted from the classification method that was originally proposed by Datta et al (2010), which is a combination of bagging and rank aggregation in a single procedure. Bagging reduces the variance and improves the accuracy of weak classifiers. For every bootstrap sample (sampling with replacement) several regression algorithms are trained and a regressor with the best performance on out-of-bag samples are kept for predicting the testing data. The weighted rank aggregation is used for multi-objective optimization, where more than one performance measure is required. Each performance measure ranks the algorithm according to the performance and the ordered lists of algorithms are then aggregated to produce a single list which ranks algorithms according to the performance. The algorithm below is a step-by- step procedure on how the ensemble regressor is built. Assuming we have a training data consisting of n samples with the vector form$\{X_{(n\times p)}, y_{(n\times 1)}\}$.

1. Initialization. Set the number of bootstrap samples to draw. Let $j = 1$, select M regression methods along with K different performance measures to be predicted.

2. Sampling. Draw a bootstrap sample of the same size from the training samples using simple random sampling with replacement so that we can obtain $\{X_j^*, y_j^*\}$. Sampling is repeated until the samples from all the classes are present in the training set. Some

samples will be repeated more than once while others will be left out of the bootstrap sample, and such samples are called out-of-bag (O-O-B) samples.

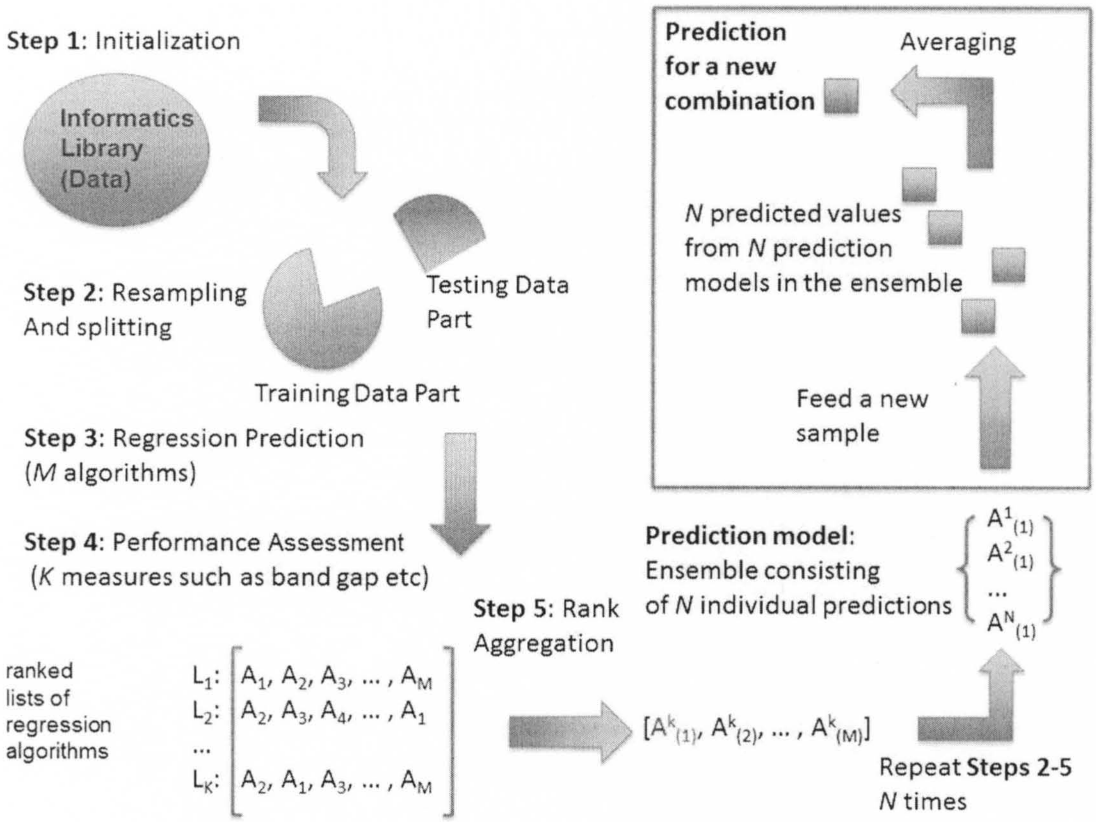
3. Prediction. Using the M regression algorithms, fit models to predict each of the K outcomes based on the bootstrap samples..

4. Performance assessment. The M models fitted in the prediction step are used to predict the classes of the OOB samples. Since we know the survival times of the samples, K different performance measures can be computed. Each measure will rank the regressiom algorithm according to the performance of the algorithm under the particular measure, producing K ordered list of size M, $L_1, \ldots, L_K$.

5. Rank Aggregation. Once we obtain the ordered lists from the performance measures, they are rank aggregated using the weighted rank aggregation procedure which determines the best regression algorithm.

Steps 2-5 are repeated many times say N times.

Predictions of a new sample. Predict N values of each of the features for a new combination using the N prediction models obtained before and average the answers to get the final predictions. Figure 3.1 shows the flowchart of both building the ensemble regressor function as well as using it to predict new samples.



Step 1: Initialization
Informatics Library (Data)
Step 2: Resampling And splitting
Testing Data Part
Training Data Part
Step 3: Regression Prediction (M algorithms)
Step 4: Performance Assessment (K measures such as band gap etc)
Step 5: Rank Aggregation
ranked lists of regression algorithms
L1: A1, A2, A3, ... , AM
L2: A2, A3, A4, ... , A1
...
LK: A2, A1, A3, ... , AM
[Ak(1), Ak(2), ... , Ak(M)]
Repeat Steps 2-5 N times
Prediction model: Ensemble consisting of N individual predictions
A1(1) A2(1) ... AN(1)
Prediction for a new combination
Averaging
N predicted values from N prediction models in the ensemble
Feed a new sample

Figure 3.1. A schematic flowchart for the regression problem, provided by Somnath Datta.

PREPROCESSING OF THE DATA

The data was first preprocessed using the pkDACLASS package proposed by Ndukum et al (2011). Basic preprocessing of the raw data involves baseline correction, denoising and binning. Baseline subtraction uses an algorithm to eliminate the baseline slope and offset from a spectrum by interactively calculating the best-fit straight line through a set of estimated baseline points (Ndukum et al). The baseline correction relies on a method that has been applied in PROcess package. The baseline is deducted by setting the bandwidth of "approx" method, in the routine bslnoff, to be 25% (Ndukum et al, 2010). For denoising, a cutoff point h is chosen such that the features selected match to real peptide peak. The principle is based on keeping features with intensities greater than a certain threshold h. The threshold should be large enough to eliminate initial noisy region but small enough to keep any peak that could match to real observable proteins or peptides. The graphs below show the how the mass spectrometry raw data looks like before and after the baseline correction and denoising.

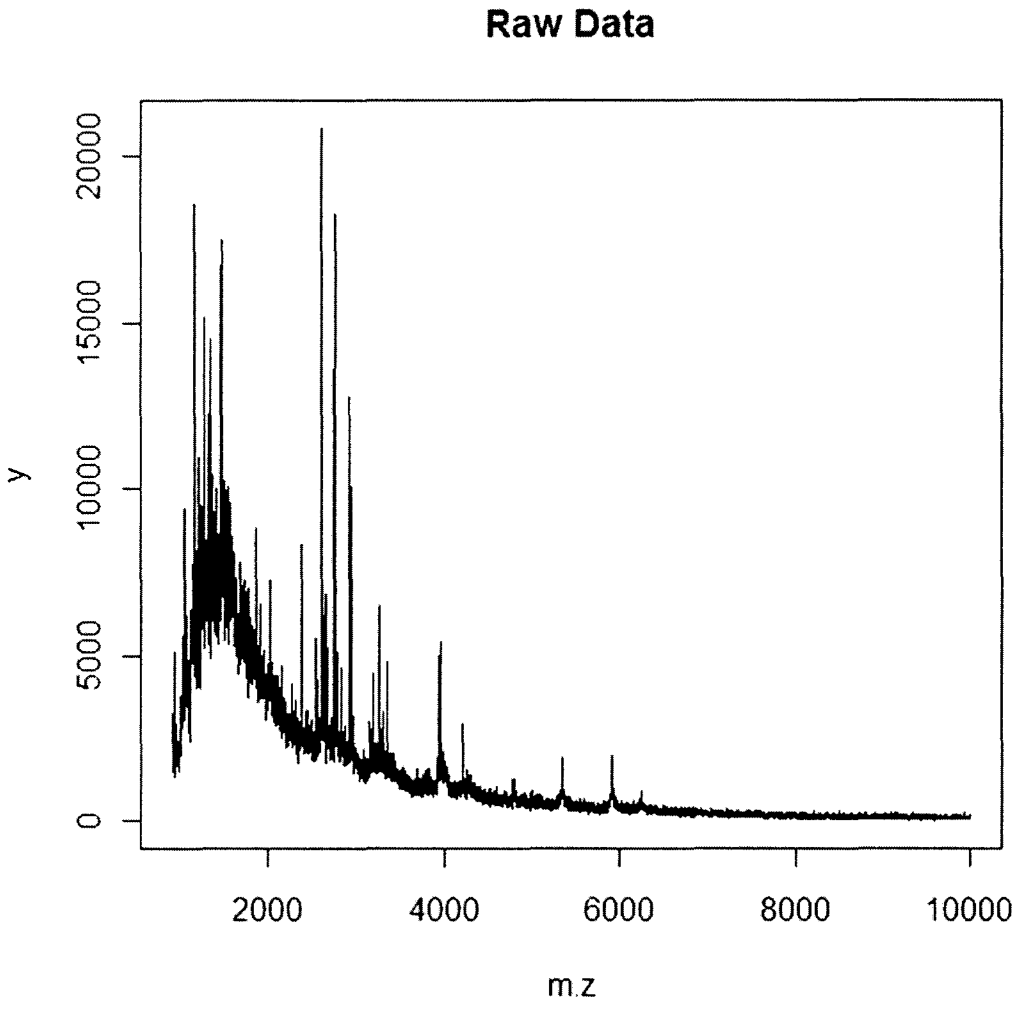


Figure 3.2 : Graph showing the raw MS Data

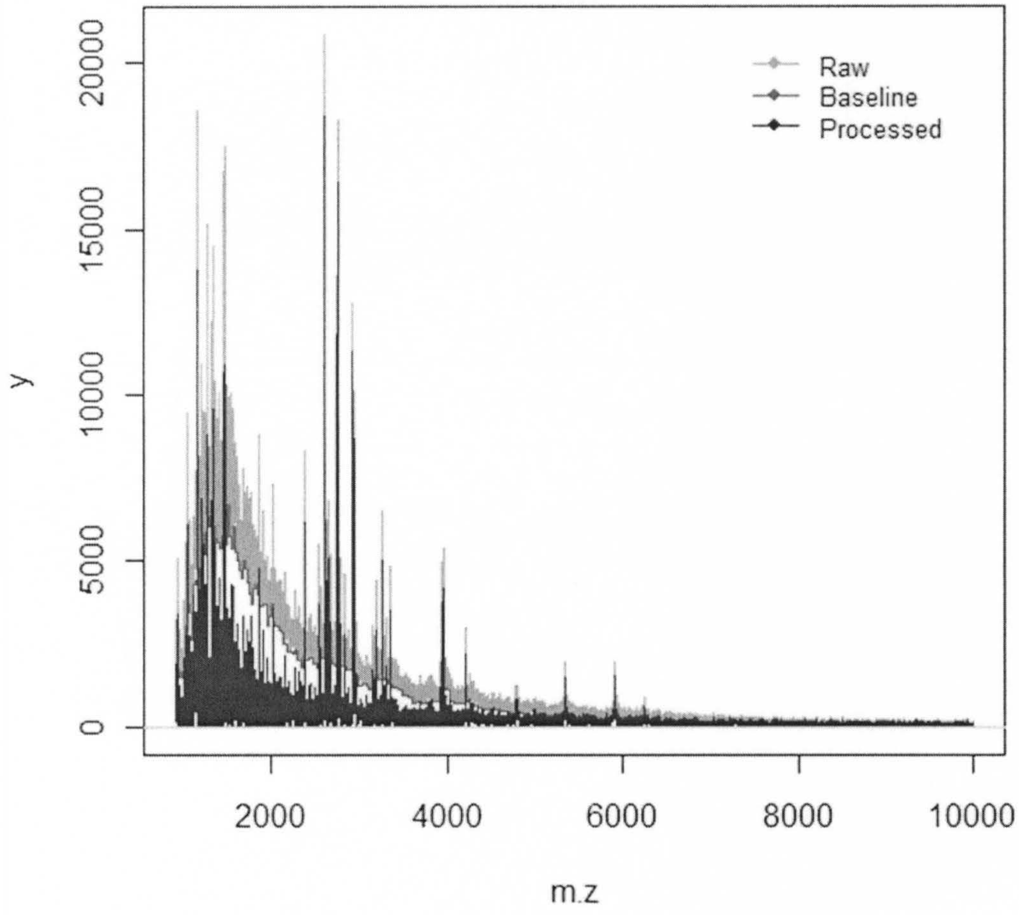


Figure 3.2: Graph showing after Baseline correction.

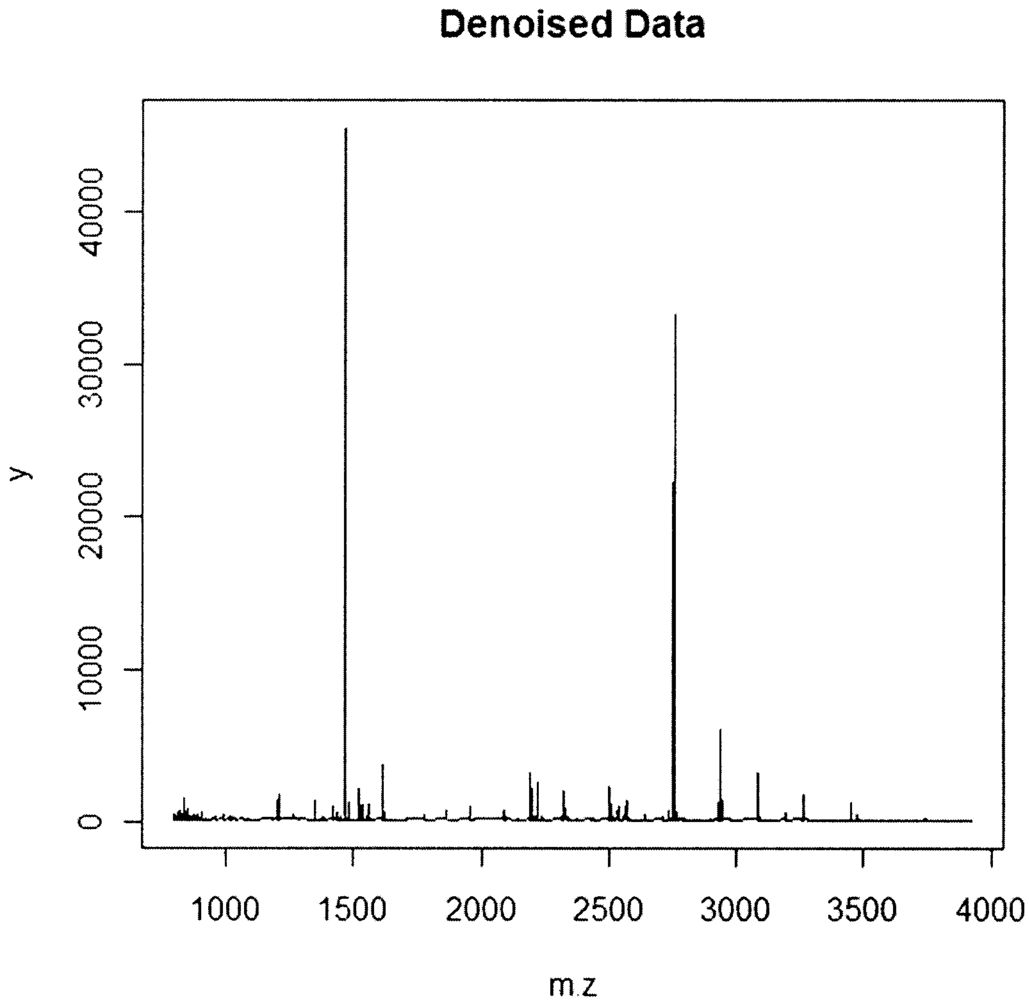


Figure 3.4: Graph showing after denoising.

RESULTS AND DISCUSSION

The data consists of 25 samples, and all 25 spectra are standardized and denoised by the use of the package pkDACLASS (Ndukum et al, 2010). As mentioned above five methods of model fiting PCR, PLS, SPLS, LASSO and Ensemble is used for each of the feature set. The algorithms were selected with the number of bootstrap samples equal to 101. The performance of each method is compared by computing the average MSEP and MAE of one hundred training and testing datasets. The ensemble gives the lowest MSEP and MAE and is similar to SPLS which is the best individual algorithm. The results of the average of 100 training and testing datasets are shown in the table below. The samples were divided into training and testing data each consisting of 14 and 11 samples respectively. 100 different training and testing datasets were randomly created.

	MSEP	MAE
PCR	0.56938	0.52794
PLS	0.56279	0.55865
SPLS	0.54552	0.50027
LASSO	0.61878	0.57916
ENSEMBLE	0.53467	0.50027

Table 3.1 : Average of Performance measures from 100 training and testing datasets.

CHAPTER 4

CONCLUSIONS AND FUTURE RESEARCH

CONCLUSIONS

For complex high dimensional datasets resulting high throughput experiments, it may be wise to consider many different classification algorithms combined with dimension reduction techniques rather than a single standard algorithm. Different algorithms with different performance measures give different results from one dataset to another. The algorithm proposed by Datta et al (2010) borrows elements from bagging and rank aggregation to create an ensemble classifier optimized with respect to several objective performance functions. The ensemble classifier is capable of adaptively adjusting its performance depending on the data, reaching the performance levels of the best performing individual classifier without knowing which one it is. In chapter two, a similar approach is carried out in the regression context. Here the dataset used is a Mass Spectrometry data and similar to classification methods, different regression models give different results. In the regression approach bagging and rank aggregation is used to create the ensemble regressor and the results show that the ensemble regressor is capable of adaptively adjusting its performance depending on the data, reaching the performance levels of the best performing individual regression model. For illustration purposed, the common classification algorithms and dimension reduction techniques and regression algorithms are used in this thesis. The procedure is implemented in R using available classification and regression routines to build the ensemble classifier and ensemble regressor.

FUTURE RESEARCH

In Chapter 3, it was investigated how Mass Spectrometry data can be used to do prediction analysis using the regression models. An interesting direction in this sense would be to study the effects of the covariates and also test the effect of regression. Furthermore the analysis can be extended to survival prediction with the use of right censoring. The data used is continuous data but categorical data can also be used in this context. For the performance measures, mode or median estimation and prediction errors can be used instead of mean which both of them seem to be consistent with majority voting. Simulations in the context of regression analysis can also be done.

REFERENCES

Agresti, A. (2002). *Categorical Data Analysis*, New York: Wiley-Interscience.

Allwein, E.L., Schapire, R.E., and Singer, Y. (2000). Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1, 113–141.

Antoniadis,A., Bigot, J. and Lambert-Lacroix, S. (2010). Peaks detection and alignment for mass spectrometry data. *Journal de la Sociètè Française de Statistique*, 151(1), 17-37.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and regression trees.* Monterey, CA; Wadsworth & Brooks/Cole Advanced Books & Software.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.

Breiman, L. (1996). Bias, variance and arcing classifiers. *Technical Report TR 460*, Statistics Department, University of California, Berkeley, CA.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5-32.

Breiman, L. (2002). Manual on Setting Up, using, and understanding Random Forests v3.1. http://www.stat.berkeley.edu/users/breiman/RandomForests/cc.home.htm.

Brier, G.W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 75, 1-3.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, 121–167.

Chandra, A. and Yao, X. (2006). Evolving hybrid ensembles of learning machines for better generalization. *Neurocomputing*, 69(7–9), 686–700.

Chang, C.C., and Lin, C.J. (2003). LIBSVM: a library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm. Tech. rep., Department of Computer Science, National Taiwan University.

Cohen, P. (2002). The origins of protein phosphorylation. *Nature Cell Biology*, 4 (5).

Datta, S., Pihur, V., and Datta, S. (2010). An adaptive optimal ensemble classifier via bagging and rank aggregation with applications to high dimensional data. *BMC Bioinformatics*, 11, 427.

De Jong, S. (1993). SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263

Dietterich, T.G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2), 139–157.

Dutkowski, D., and Gambin, A. (2007). On Consensus biomarker selection. *BMC Bioinformatics*, 8, Suppl 5:S5.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001). Rank aggregation methods for the Web. *In proceedings of the 10th international conference on World Wide Web Hong Kong Elsevier Science*, 613-622.

Eads, C.A., Danenberg, K.D., Kawakami, K., Saltz, L.B., Blake, C., Shibata, D., Danenberg, P.V., and Laird, P.W. (2000). MethyLight: a high-throughput assay to measure DNA methylation. *Nucleic Acids Res* 28, E32.

Fisher, R.A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 179-188.

Freund, Y. and Schapire, R.E. (1996). Experiments with a new boosting algorithm. Machine Learning: Proceedings of the Thirteenth International Conference (ICML '96), Morgan Kaufmann, San Francisco, CA, 148–156.

Galton, F. (1892). *Finger Prints* Macmillan, London.

Gamez-Pozo, A., Sanchez-Navarro, I., Nistal, M., Calvo, E., Madero, R., Diaz, E., Camafeita, E., Vara J.A.F., and et al (2009). MALDI profiling of human lung cancer subtypes. *PLoS ONE,* 4 (11), art. no. e7731.

German, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias-variance dilemma. *Neural Computation*, 4(1), 1–58.

Goldenberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning Reading*, MA, Addison Wesley.

Hanash, S. (2008). Disease proteomics. *Nature*, 422 (6928), 226-232.

Hansen, L.K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 993–1001.

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

Indurkhyn, N. and Sholom, M. (2001). Solving regression problems with rule-based ensemble classifiers. *ACM SIGKDD*, 287–292.

Jemal, A., Siegel, R., Ward, E., Murray, T., Xu, J., and Thun, M.J. (2007). Cancer statistics, *J Clin*, 57, 43-66.

Jimenez, C.R., El Filali, Z., Knol, J.C., Hoekman, K., Kruyt, F.A.E., Giaccone, G., Smit, A.B., and Li, K.W. (2007). Automated serum peptide profiling using novel magnetic C18 beads off-line coupled to MALDI-TOF-MS. *Proteomics - Clinical Applications*, 1 (6), 598-604.

Karpova M.A., Moshkovskii S.A., Toropygin I.Y., and Archakov A.I. Cancer-specific MALDI-TOF profiles of blood serum and plasma: Biological meaning and perspectives (2010) Journal of Proteomics, 73 (3), 537-551.

Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., and Meltzer, P.S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med*, 7, 673-679.

Kong, E. and Dietterich,T.G. (1995). Error-correcting output coding correct bias and variance. *In The XII International Conference on Machine Learning*, San Francisco, CA, 313-321.

Larkey, L.S. and Croft, W.B. (1997). Combining classifiers in text categorization, in H.-P. Frei, D. Harman, P. Sch¨auble and R. Wilkinson (eds.), *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* ACM Press, New York, 289–297.

LeBlanc, M., and Tibshirani, R. (1996). Combining estimates in regression and classification. *Journal of American Statistical Association*, 91(436), 1641-1650.

Lee, Y., and Lee, C.K. (2003). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, 19(9), 132-1139.

Liotta, L.A., and Petricoin, E.F. (2006). Serum peptidome for cancer detection: spinning biologic trash into diagnostic gold. *J Clin Invest*, 116, 26-30.

Makrantoni, V., Antrobus, R., Botting, C.H., and Coote, P.J. (2005). Rapid enrichment and analysis of yeast phosphoproteins using affinity chromatography, 2D-PAGE and peptide mass fingerprinting. *Yeast*, 22 (5), 401-414.

Marchevsky, A.M., Shah, S., and Patel, S. (1999). Reasoning with uncertainty in pathology: artificial neural networks and logistic regression as tools for prediction of lymph node status in breast cancer patients. *Mod Pathol*, 12, 505–513.

Marchevsky, A.M., Tsou, J.A., and Laird-Offringa, I.A. (2004). Classification of individual lung cancer cell lines based on DNA methylation markers: Use of linear discriminant analysis and artificial neural networks. *Journal of Molecular Diagnostics*, 6 (1), 28-36.

Mason, L., Bartlett, P., and Baxter, J. (2000). Improved generalization through explicit optimization of margins. *Machine Learning.*

Ndukum, J., Atlas, M., and Datta, S. (2011). pkDACLASS: Open source software for analyzing MALDI-TOF data. *Bioinformation* 6(1): 45-47.

Oda, Y., Nagasu, T., and Chait, B.T. (2001). Enrichment analysis of phosphorylated proteins as a tool for probing the phosphoproteome. *Nature Biotechnology*, 19 (4), 379-382.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11, 169–198.

Pekhimenko, G. Penalizied Logistic Regression for Classification. http://www.cs.cmu.edu/~gpekhime/Projects/CSC2515/project.pdf

Perrone, M.P. and Cooper, L.N. (1993). When networks disagree: Ensemble methods for hybrid neural networks. *Neural Networks for Speech and Image Processing*, Chapman-Hall, London, 126–142.

Phelps, R.M., Johnson, B.E., Ihde, D.C., Gazdar, A.F., Carbone, D.P., McClintock, P.R., Linnoila, R.I., Matthews, M.J., Bunn, P.A., Jr., Carney, D., Minna, J.D., and Mulshine, J.L. (1996). NCI-Navy Medical Oncology Branch cell line data base. *J. Cell. Biochem*, Suppl., 24: 32–91.

Pihur, V., Datta, S., and Datta, S. (2007). Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. *Bioinformatics*, 23(13), 1607-1615.

Pihur, V., Datta, S., and Datta, S. (2009). RankAggreg, an R package for weighted rank aggregation. *BMC Bioinformatics*, 10(62).

Reyzin, L. and Schapire, R.E. (2006). How boosting the margin can also boost classifier complexity. *ICML '06: Proceedings of the 23rd International Conference on Machine Learning,* ACM, New York, 753–760.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning,* 5(2), 197–227.

Schapire, R.E., Freund, Y., Bartlett, P. and Lee, W. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5), 1651–1686.

Schiller, J.H., Harrington, D., Belani, C.P., Langer, C., Sandler, A., Krook, J., Zhu, J., Johnson, D.H., and Eastern Cooperative Oncology Group (2002). Comparison of four chemotherapy regimens for advanced non-small-cell lung cancer. *N Engl J Med*, 346, 92-98.

Simon, R. (2005). Roadmap for Developing and Validating Therapeutically Relevant Genomic Classifiers. *J Clin Oncol*, 23(29), 7332-7341.

Skates, S.J., Horick, N., Yu, Y., Xu, F.J., Berchuck, A., Havrilesky, L.J., De Bruijn, H.W.A., Bast Jr., R.C., and et al. (2004). Preoperative sensitivity and specificity for early-stage ovarian cancer when combining cancer antigen CA-125II, CA 15-3, CA 72-4, and macrophage colony-stimulating factor using mixtures of multivariate normal distributions. *Journal of Clinical Oncology*, 22 (20), 4059-4066.

Skurichina, M. and Duin, R.P.W. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis and Applications,* 5(2), 121–135.

Smit, E.F., Van Meerbeeck, Jan P.A.M., Lianes, P., Debruyne, C., Legrand, C., Schramel, F., Smit, H., and et al. (2003). Three-arm randomized study of two cisplatin-based regimens and paclitaxel plus gemcitabine in advanced non-small-cell lung cancer: A phase III trial of the European Organization for Research and Treatment of Cancer Lung Cancer Group - EORTC 08975. *Journal of Clinical Oncology*, 21, 3909-3917.

Sozzi, G. (2001). Molecular biology of lung cancer. *Eur J Cancer,* 37, 63–73.

Spira, A., and Ettinger, D.S. (2004). Multidisciplinary Management of Lung Cancer. *N Engl J Med*, 350, 379-392.

Srivastava, D and Bhambhu L. (2010). Data Classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, 1.

ter Braak, C.J.F. and De Jong, S. (1998). The objective function of partial least-squares regression. *Journal of Chemometrics*, 12, 41-54.

Topchy, A., Jain, A.K. and Punch, W. (2004). A mixture model for clustering ensembles, in M. W. Berry, U. Dayal,C. Kamath and D. Skillicorn (eds.). *Proceedings of the*

Fourth SIAM International Conference on Data Mining, SIAM, Philadelphia, PA, 379–390.

Tsou, J.A., Hagen, J.A., Carpenter, C.L., and Laird-Offringa, I.A. (2002). DNA methylation analysis: a powerful new tool for lung cancer diagnosis. *Oncogene*, 21, 5450–5461.

Valentini, G., and Dietterich, T. (2004). Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *Journal of Machine Learning Research*, 5, 725–775.

Valentini, G. and Masulli, F. (2002). Ensembles of learning machines, in M. Marinaro and R. Tagliaferri (eds.). *Neural Nets: 13th Italian Workshop on Neural Nets, WIRN VIETRI 2002, Vietri sul Mare, Italy, May 30–June 1, 2002. Revised Papers,* Vol. 2486 *of Lecture Notes in Computer Science*, Springer, Berlin, 3–19.

Vapnik, V.N. (1998). *Statistical Learning Theory*, John Wiley and Sons.

Virmani, A.K., Tsou, J.A., Siegmund, K.D., Shen, L.Y., Long, T.I., Laird, P.W., Gazdar, A.F., and Laird-Offringa, I.A. (2002). Hierarchical clustering of lung cancer cell lines using DNA methylation markers. *Cancer Epidemiol Biomarkers Prev,* 11, 291–297.

Voortman, J., Pham, T.V., Knol, J.C., Giaccone, G., and Jimenez, C.R. (2009). Prediction of outcome of non-small cell lung cancer patients treated with chemotherapy and bortezomib by time-course MALDI-TOF-MS serum peptide profiling. *Proteome Science*, 7, 34.

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. In P.R. Krishnaiah, editor, *Multivariate Analysis*, 391-420. Academic Press, New York.

Yeung, K.Y., and Bumgarner, R.E. (2003). Multiclass classification of microarray data with repeated measurements: application to cancer. *Genome Bio*, 4, R83.

Yu, H., Yang, J., and Han, J. Classifying large data sets using SVMs with hierarchical clusters. (2003). *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Yu, J., Ni, M., Xu, J., Zhang, H., Gao, B., Gu, J., Chen, J., Zhang, L., Wu, M., Zhen, S., and Zhu, J. (2002). Methylation profiling of twenty promoter-CpG islands of genes which may contribute to hepatocellular carcinogenesis. *BMC Cancer*, 2, 29-42.

Zhang, C.-X. and Zhang, J.-S. (2008). A local boosting algorithm for solving classification problems. *Computational Statistics & Data Analysis*, 52(4), 1928–1941.

APPENDIX

R CODE

```
ensembleRegressor <- function(x, y, M=10,...){
rownames(x) <- NULL # to suppress the warning message about duplicate rownames
fit.individual=TRUE
algorithms = c("pls","spls", "lasso", "pcr")
validation = c("MSEP","MAE")
weighted = TRUE
distance ="Spearman"
mse.pred <- function (a, b) { mean((a-b)^2)}
mae.pred <- function (a, b) { mean(abs(a-b))}
nalg <- length(algorithms)
nvm <- length(validation)
fittedModels <- list()
n <- length(y)
for(k in 1:M){
        s <- sample(n, replace = TRUE)
        fs <- 1:ncol(x)
        training <- x[s, fs]
        testing <- x[-unique(s), fs]
        trainY <- y[s]
#########################################################################
############## train all algorithms on the subset ##############################
######## algorithms=c("pls", "spls", "lasso", "pcr", "elasticnet")
#######################
########################################################################
Res <- list()
for(j in 1:nalg) {
       Res[[j]] <- switch(algorithms[j],
       "pls"  = plsr(y ~ . , data = data.frame(y = trainY, x = training),validation = "none",
method = "oscorespls"),
       "spls" = spls(x, y, K = 14, eta = 0.1, scale.x= FALSE, scale.y=FALSE,
trace=FALSE),
       "lasso"= lars(x, y, type = "lasso", use.Gram = FALSE, normalize = FALSE),
       "pcr"  = pcr(y ~. , data = data.frame(y= trainY, x = training),validation = "none"))

attr(Res[[j]], "algorithm") <- algorithms[j]

} # Train Part For Loop
```

```
# predict using fitted models
predicted <- list()
for(j in 1:nalg){
        switch(algorithms[j],
        "pls"       = {predicted[[j]] <- predict(Res[[j]], testing, type = "response")},
         "spls"       = {predicted[[j]] <- predict(Res[[j]], testing, type = "fit")},
         "lasso"       = {predicted[[j]] <- predict(Res[[j]], testing)$fit},
          "pcr"        = {predicted[[j]] <- predict(Res[[j]], testing, type = "response")}
)
} # Prediction part for loop

# compute validation measures
scores <- matrix(0, nalg, nvm)
 rownames(scores) <- algorithms
colnames(scores) <- validation
truth <- y[-unique(s)]
for(i in 1:nalg)
 for(j in 1:nvm)
scores[i,j] <- switch(validation[j],
"MSEP"   = mse.pred(predicted[[i]], truth),
"MAE"    = mae.pred(predicted[[i]], truth)
)
convertScores <- function(scores){
scores <- t(scores)
 ranks <- matrix(0, nrow(scores), ncol(scores))
weights <- ranks
for(i in 1:nrow(scores)){
        ms <- sort(scores[i,], decr=FALSE, ind=TRUE)
         ranks[i,] <- colnames(scores)[ms$ix]
         weights[i,] <- ms$x
 }
 list(ranks = ranks, weights = weights)
}
# perform rank aggregation
convScores <- convertScores(scores)
if(nvm > 1 && nalg <= 6)
fittedModels[[k]] <- Res[[which(algorithms == RankAggreg(convScores$ranks,
                          nalg, convScores$weights, distance=distance,
verbose=FALSE)$top.list[1])]]
else
fittedModels[[k]] <- Res[[which.min(scores[,1])]]
```

```
} # End of for Loop Iteration 1:M

# how many times each algorithms was the best?
bestAlg <- unlist(sapply(fittedModels, FUN = function(x) attr(x, "algorithm")))
res <- list(models = fittedModels, M = M,
bestAlg = bestAlg, convScores=convScores)
class(res) <- "ensemble"
res
}

predictEns <- function(EnsObject, newdata, y=NULL){
          mse.pred <- function (a, b) { mean((a-b)^2)}
          mae.pred <- function (a, b) { mean(abs(a-b))}
          M <- EnsObject$M
           n <- nrow(newdata)
           predicted <- matrix(0, n, M)
           for(i in 1:M){
           testing <- newdata
           switch(attr(EnsObject$models[[i]], "algorithm"),
"pls"     = predicted[,i] <- predict(EnsObject$models[[i]], testing, type = "response"),
"spls"    = predicted[,i] <- predict(EnsObject$models[[i]], testing, type = "fit"),
"lasso"   = predicted[,i] <- predict(EnsObject$models[[i]], testing)$fit,
"pcr"     = predicted[,i] <- predict(EnsObject$models[[i]], testing, type = "response")
)
 }
 res <- list()
 if(!is.null(y)){ # compute validation measures
valM <- c("MSEP", "MAE")
MAE  <- mae.pred(predicted, y)
MSEP <- mse.pred(predicted, y)
ensemblePerformance <- matrix(c(MSEP, MAE),1,2)
colnames(ensemblePerformance) <- valM
rownames(ensemblePerformance) <- "ensemble"
}
res <- list(pred=predicted, ensemblePerf=ensemblePerformance)
class(res) <- "predictEnsemble"
res
}
```

CURRICULUM VITAE

10905 Keene Road
Louisville
Kentucky, 40241

Phone 586-873-2750
E-mail
jasmeet_shah87@hotmail.com
jasmit.shah@yahoo.com

Jasmit S Shah

Career Objective

A part-time position in the field of Research with a background of Statistics

Education

Masters of Science, Biostatistics August 2011

- Current GPA 3.83

Bachelor of Science, Mathematics May 2009

- *Minor: Chemistry*
- GPA 3.7/4.0

Work Experience

Student Assistant at University of Louisville School of Public Health: Jan 2010 – December 2010
Student Assistant University of Louisville Gheens and Aging Department: Aug 2009 – Dec 2009

Relevant Coursework

Survival Analysis
Bayesian Analysis
Categorical Data Analysis
Calculus I,II & III
Intro to Java
Advanced Computing
High Throughput Data Analysis
Biostatistics
Differential Equations
Applied Statistics and Probability

Honors

Dean's List January 2004 – Current

Top Student in Mathematics 2002

Silver Standard of the President's Award: 2002

Bronze Standard of the President's Award: 2001

Skills

Computer Skills of Ms Word, Excel, Access and PowerPoint

Statistical Software Skills of R and SAS

Programming Skills of Basic C++ and Java

Languages Known: English, Swahili, Gujarati and Hindi