Comparing Support Vector Machine and Random Forest for Predicting Lung Cancer using Single Nucleotide Polymorphisms

Angeline Madrid
Northeastern University

ABSTRACT

Lung cancer risk prediction is particularly difficult in that environmental factors (e.g., smoking) play a huge role in whether or not lung cancer develops. In addition, it is believed that lung cancer is not a disease based on a single gene but is probably associated with the joint effects of many genes. (Yoo, 2012) Both the random forest (RF) and the support vector machine (SVM; Hasibuan, 2014; Fallon, 2013) methods have been shown to be particularly good at identifying multiple predominant features in biological datasets. RF is a classification method that has gained recent popularity for predicting lung cancer risk based on single nucleotide polymorphisms (SNPs). While others have used SVMs to classify SNPs in many different organisms, very few have been applied to classifying SNPs associated with lung cancer. (Hasibuan, 2014; Fallon, 2013) Here, we train two classifiers (SVM and RF) using the class labels of "lung cancer" and "no lung cancer" to identify those features most associated with lung cancer. Our best RF model and SVM models resulted in prediction accuracies of 82.9% and 83.1%, respectively. Both models also determined segment size as the feature with the highest importance. While the RF model had a slightly lower accuracy than the SVM model, creating the RF was much faster and more efficient.

INTRODUCTION

SNPs are a common variation in a DNA sequence where the difference occurring is due to a modification in a single nucleotide either through a change in the base pair, or an insertion or deletion of a base pair. In predicting lung cancer risk, several SNPs have been identified as being associated with lung cancer risk. (Li, 2012). While the major cause of lung cancer is environmental (e.g., smoking), genetic factors may increase the risk of smokers developing cancer by as much as 2.4 times. There is a relatively substantial portion of those who develop lung cancer but are nonsmokers implying a genetic component exists that increases the risk.

Random Forest

A random forest classifier is built on growing multiple decision trees, where each tree classifies on a subset of the data, called the out-of-bag (oob) portion of the data. Each subset of the data is created randomly with replacement from the original dataset. Each decision tree is grown as big as possible without pruning and chooses a class, and the forest then selects the class chosen by the most trees. The random forest is essentially averaging across high variance and low bias trees in order to produce a low bias, low variance classifier. (Breiman, 2001)

The low correlations between trees and the increased strength of each tree are optimized in a random forest in order to keep error rates to a minimum. The parametrization on the number of features used for each individual tree is a balancing act between these two characteristics of the error rate. If the subset of features is large then both the correlation and strength are increased, while smaller subsets of features decrease both.

A random forest classifier is able to immediately identify those features essential to the classification by using the importance metric. This measure has two components based on permuting on the oob subsets and decreasing impurities of the nodes as they split. The prediction errors of each oob portion are averaged and normalized by the standard deviation of the differences. The second component of the importance measure is called the Gini index,

$$G = 1 - \sum_j p_j^2 ,$$

where p_j is the probability at class j. Much like entropy, this is one way of measuring impurity. When all classes in the table have equal probability the Gini index is at a maximum. If there is only one class, the Gini index is zero. The lower a Gini index (or higher decrease) is the greater the role feature has in splitting the data. (Teknomo, 2009)

The highlights of a random forest classifier include:

- No overfittting
- Highly accurate
- Efficient at handling large datasets and feature sets
- Identifies which features are central in the classification.

Support Vector Machine

An SVM is a linear classifier that makes new assignments on binary classes. Its decision function (how it makes assignments) is based on a linear separation of the input space of the training data. The decision function is $f(x)=\langle w,x\rangle + b$, where $w = \Sigma(\alpha_i y_i x_i)$, a linear combination of the training points with non-negative coefficients. The decision function attempts to create a maximum margin hyperplane that divides the data according to the binary class to which they belong by mapping the data in a higher order feature space. Those points that satisfy $\langle w,x\rangle + b=0$ lie on the hyperplane; those that satisfy $\langle w,x\rangle + b>0$ lie above the hyperplane; and those that satisfy $\langle w,x\rangle + b<0$ lie below the hyperplane. By mapping to a higher dimensionality, the SVM is then able classify on non-linear data as a linear classifier. The values of w are those points that define the hyperplane, and the values of x are those from the training dataset. (Boser et al., 1992; Cortes and Vapnik, 1995)

When classifying nonlinear data, a kernel function is used instead of the dot product $\langle w,x\rangle$. In higher dimensional space, the kernel function is the dot product. Some examples of kernel functions include Gaussian, polynomial, and radial basis. These kernel functions, just like the dot product, exploit the similarity between the data by

using the information about the dot products between data items to define possible patterns in the data.

Validation Method

A common way for validating classifiers is to use the method of 5-fold cross validation. The original sample will be randomly partitioned into 5 equal subsets. Of these, a single subset is kept the test set and the remaining 4 subsets will be used as the training data. The cross-validation will then be repeated 4 times with each of the 5 subsamples used exactly once as the test data. The 5 results from the folds will then be averaged. This method is advantageous because all observations are used for both training and testing, and each observation is used for testing exactly once.

Performance Measure

To measure performance, receiver operating characteristic curves (ROC) and the area under those curves (AUC) are typical metrics. The ROC is a plot of true positive rate (TPR) versus the false positive rate (FPR). Perfect classification is when the TPR is 100% and the FPR is at 0%. This occurs in the upper left corner of the ROC curve. The closer the curve gets to this corner of the plot, the better the performance of the classifier. A curve that is a straight line from lower left to upper right indicates random performance, where this classifier does no better than a coin flip. The AUC ranges from 0 to 1, where 0.5 represents a classifier that does no better than randomly guessing. Those classifiers with AUC values that are close to 1 are highly performing classifiers. The AUC value of a classifier essentially represents the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. (Fawcett, 2006)

The Cancer Genome Atlas

Here, we use a dataset from The Cancer Genome Atlas (https://tcga-data.nci.nih.gov/tcga/) containing a collection of SNP data from lung adenocarcinoma tissue along with their corresponding normal tissue samples for training RF and SVM models. Each model is trained using 5-fold cross validation and is evaluated by ROC and AUC measures.

METHODS (CODE WITH DOCUMENTATION)

Data Set

The data were obtained from The Cancer Genome Atlas (https://tcga-data.nci.nih.gov/tcga/) a National Institutes of Health data portal that stores genomic data from tumorous and non-tumorous tissue. It contains a set of individuals where each individual is a collection of SNP data from lung adenocarcinoma tissue along with their corresponding normal tissue samples. There are 110 individuals with each containing a few hundred samples (several per chromosome). There are a total of 220 files: a tumorous sample and normal sample for each individual. The data fields include Sample, Chromosome, Start, End, Num_Probes, and Segment_Mean. These are described in Table 1.

Table. 1 Data fields and descriptions in TCGA data set.

Data Field	*Description*
Sample	The sample name (i.e., the ID of the individual)
Chromosome	The number of the chromosome containing the segment
Start	The start position of the segment containing the SNP
End	The end position of the segment containing the SNP
Num_Probes	The number of probes used in the hybridization in the experiment to extract the segment
Segment_Mean	A measure of the ratio of intensities of the tumor tissue to normal tissue.

Preprocessing

The data were pre-processed in order to create a list of features for training and testing the classifiers. Those features are CHROMOSOME NUMBERS, NUMBER OF PROBES, COPY NUMBER (segment mean), NUMBER OF GENES, VALUE OF THE MOST FREQUENT GENE, and LENGTH OF SEGMENT (segment size). The CHROMOSOME NUMBER is just the number of the chromosome where the SNP is located. The NUMBER OF PROBES is the integer number of probes used in the sequencing experiment. The COPY NUMBER is the log2 ratio of the tumor intensity to the normal intensity (i.e., the segment_mean). The copy number refers to the number of copies of one or more sections of the DNA. As gene duplication occurs sometimes extra copies of a DNA segment are made altering the structure of the DNA. This copy number variation (CNV) results in the cell having an abnormal or, for certain genes, a normal the number of copies of genes. CNV has been linked to cancer progression. CNVs are measured based on the signals of the tissue intensities that are captured during the experiment. This measurement is the segment mean and represents the log2 ratios of those intensities (Olshen 2004). The values for this feature will be converted to an absolute copy number by the transformation (2^seg_mean)*2. The NUMBER OF GENES is the count of the number genes found in the given SNP segment. The VALUE OF THE MOST FREQUENT GENE is the highest count of the gene most frequently found in that SNP segment. The LENGTH OF SEGMENT is the size of the segment (End position minus Start position).

To calculate the NUMBER OF GENES and VALUE OF MOST FREQUENT GENE, `parsePos.R` parsed each file for chromosome number, and start and end position. These three variables were used to create input files to the UCSC genome browser (http://genome.ucsc.edu/cgi-bin/hgTables). The UCSC genome browser provided output files containing chromosome, gene name, and start and end position of that gene. Using `predDataAddGenes.R`, each of these UCSC output files were parsed for gene names, which were appended to the data set to be used for classification. In addition, for each tumorous sample a cancer label of "1" was added, and for each normal sample a label of "-1" was added. Using `mergFiles.R`, all the preprocessed data files are merged into one .csv file consisting of 65,051 samples with the six features listed above and a label of cancer or no cancer.

Classification Models

Prior to classification, the variables num_probe, num_genes, most_freq, and seg_size were normalized. Training and testing datasets were created based on a 75%-25% split. The models built are listed in Table2

Table 2. List of models

Model
A. RF model with 5-fold cross validation using the R package `caret`.
B. RF without cross validation using the R package `randomForest`.
C. SVM/linear kernel with 5-fold cross validation using the R package `caret`.
D. SVM/polynomial kernel with 5-fold cross validation using the R package `caret`.
E. SVM/radial kernel with 5-fold cross validation using the R package `caret`.

Two RF models (A and B) were built on the training data set one using 5-fold cross validation and one with out. Model A was trained using the `train()` function in the R package `caret`. The `trControl` parameter was set to "repeatedcv" with 5 repeats. Model B was trained using the `randomForest()` function in the R package `randomForest`. Prediction was made on the testing data set using the `predict()` function from each package. Variable importance was produced using the `varImp()` function. Calculations of ROC and AUC were done using `roc()` and `auc()` functions from the `pROC` package in R.

Model B was trained using the `randomForest()` function from the `randomForest` R package and `predict()` was used for the predictions on the testing data set. Variable importance was produced using the `importance()` function. Calculations of ROC and AUC were done using `performance()` function from the `ROCR` package in R.

Three SVM models (C, D, and E) were built on the training data set using 5-fold cross validation using the `train()` function in the R package `caret`. Model C was trained using the method "svmLinear", which uses a linear kernel. Variable importance was produced using the `varImp()` function. Prediction was made on the testing data set using the `predict()` function from the `caret` package. Calculations of ROC and AUC were done using `roc()` and `auc()` functions from the `pROC` package in R.

Model D was trained using the method "svmPoly", which uses a polynomial kernel. Prediction was made on the testing data set using the `predict()` function from the `caret` package. Variable importance, ROC and AUC were not calculated for model D.

Model E was trained using the method "svmRadial", which uses a radial kernel. Prediction was made on the testing data set using the `predict()` function from the `caret` package. Variable importance was produced using the `varImp()` function.

Calculations of ROC and AUC were done using `roc()` and `auc()` functions from the `pROC` package in R.

RESULTS

Accuracies

The accuracies of all the models are shown Table 3. The model with the highest accuracy of 0.8296 is Model B, the RF model without 5-fold cross validation.

Table 3. Accuracies of classification models.

Model	**Accuracy**
A. RF model with 5-fold cross validation using the R package `caret`.	0.8115
B. RF with no cross validation using the R package `randomForest`.	0.8296
C. SVM/linear kernel with 5-fold cross validation using the R package `caret`.	0.7885
D. SVM/polynomial kernel with 5-fold cross validation using the R package `caret`.	0.8309
E. SVM/radial kernel with 5-fold cross validation using the R package `caret`.	0.8244

Importance Variables

In Model A, Model B, Model C, and Model E the top three importance variables (with the largest mean decrease Gini values) were seg_size, num_probes, and num_genes. For Model A the three chromosomes with the largest mean decrease Gini values are chromosome 1, chromosome 6, and chromosome 8. For Model B the top three were chromosome 6, chromosome 8, and chromosome 5. For Model C and Model E the top three were chromosome X, chromosome 20, and chromosome 10.

ROC and AUC

The ROC curve for Model A is shown in Figure 1 with an AUC is 0.6234.

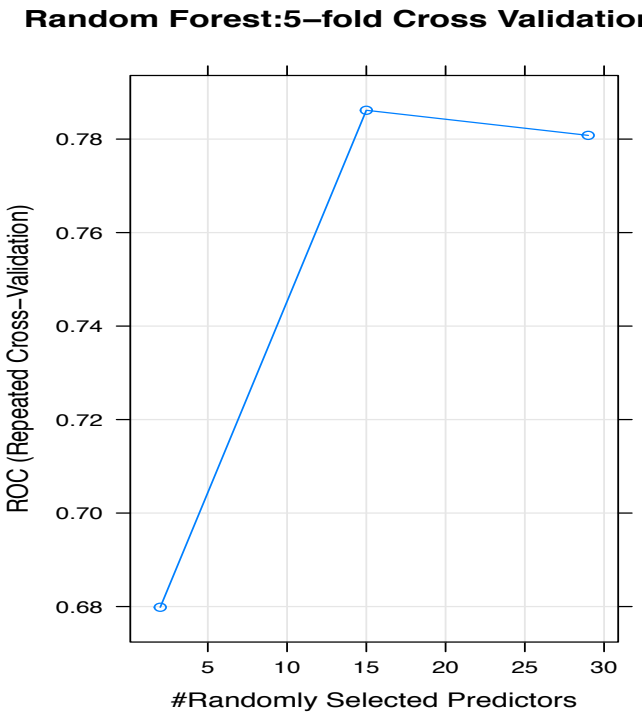

Figure 1. ROC curve for RF model using 5-fold cross validation.

The ROC curve for Model B is shown in Figure 2 with an AUC of 0.75.

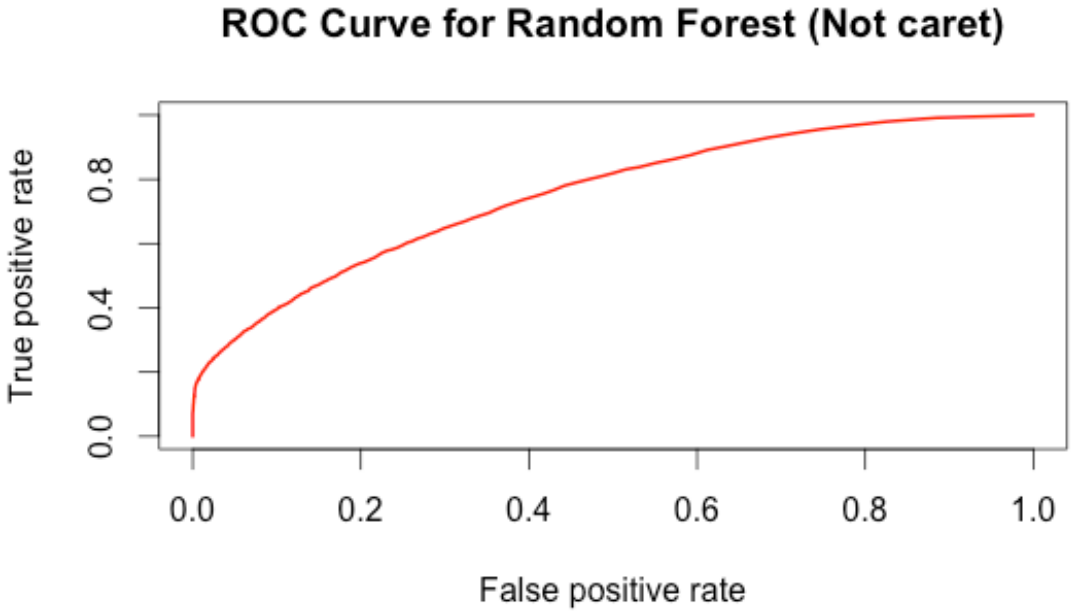

Figure 2. ROC curve for RF not using 5-fold cross validation.

The plot in Figure 3 is a sensitivity versus specificity curve for Model C with an AUC of 0.5.

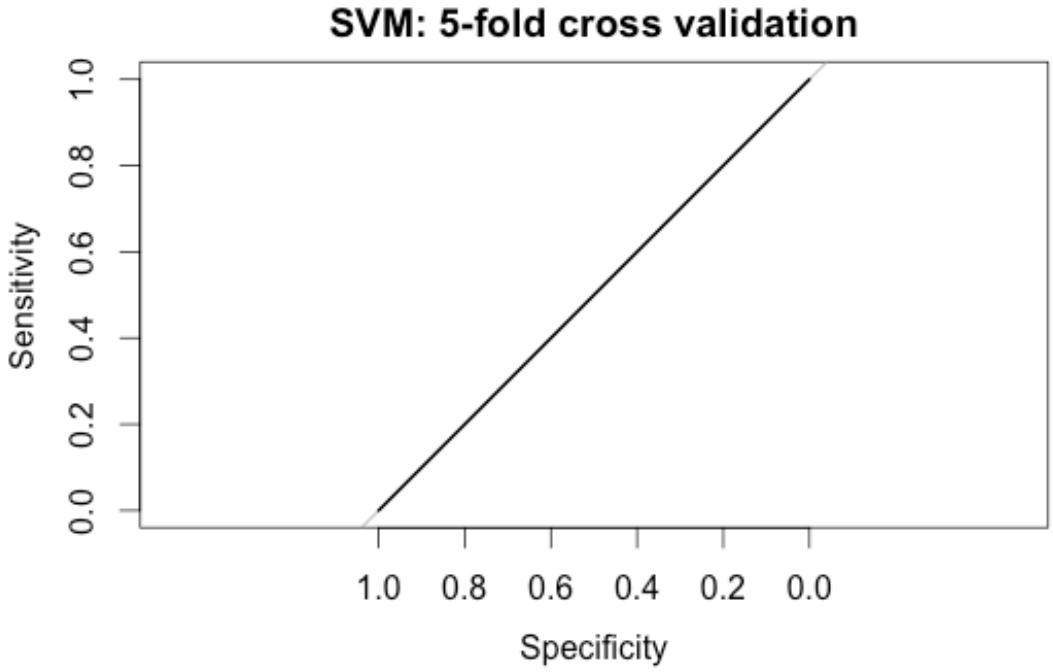

Figure 3. Sensitivity vs. specificity plot for SVM-linear model.

The plot in Figure 4 is a sensitivity versus specificity curve for Model E with an AUC of 0.5982.

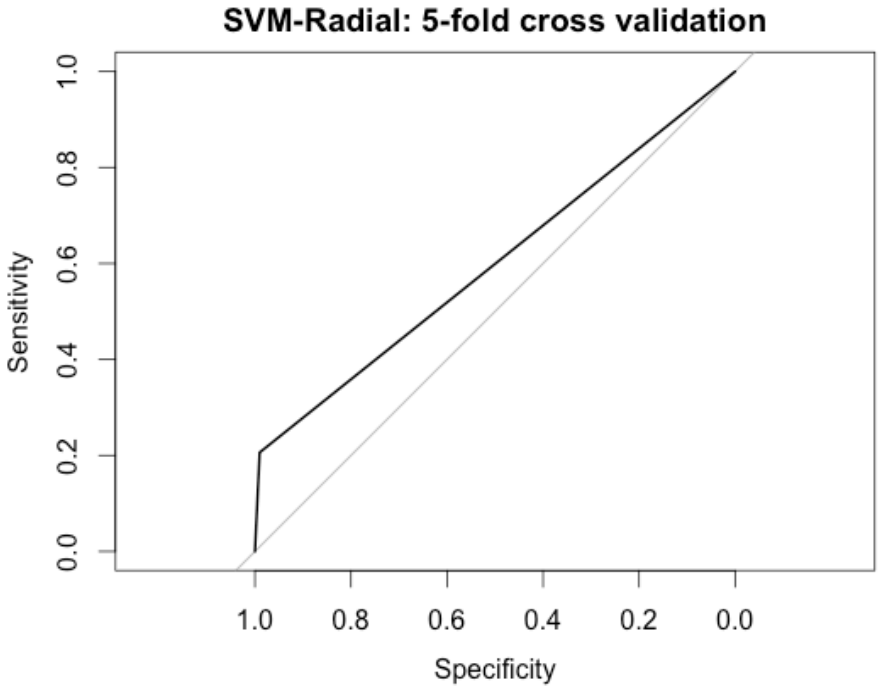

Figure 4. Sensitivity vs. specificity plot for SVM-radial model.

DISCUSSION

Random Forest

The two models built using RF had accuracies of 0.8115 and 0.8296 (with 5-fold cross-validation and without, respectively). The higher accuracy of the model not using 5-fold cross validation is interesting and implies that the model using 5-fold cross validation may be over-fitted. As discussed above, the RF method builds multiple trees on portions of the data with all the trees averaged at the end. The oob error estimate takes care of this internally. (Breiman, 2001) In addition, the model without 5-fold cross validation also had better ROC curve and higher AUC (0.75 versus 0.6234.

Both of these models listed chromosome 6 and chromosome 7 as variables of importance after seg_size, num_probes, and num_genes. Since all models created here (both RF and SVM) listed seg_size, num_probes, and num_genes as variables of importance, using chromosomes as features for prediction may not be ideal.

Support Vector Machine

The three models (C, D, and E) built using SVM had accuracies of 0.7885, 0.8309, and 0.8244 (with linear, polynomial, and radial kernels, respectively). Model C classified all observations as "cancer", and by doing so resulted in a fairly high accuracy. However, this model produced an ROC and an AUC (0.5) that are essentially identical to what would be produced through random guessing.

While the SVM-polynomial model had the highest accuracy, it took appreciably more time (days versus hours) to create than the SVM-linear or either of the RF models. While the SVM-polynomial model had the highest accuracy, it is too time-consuming. In addition, due to the long computing times, the importance variables, ROC and AUC metrics were not produced. Without these, it is difficult to determine whether this higher accuracy is meaningful.

Model E took, by far, the longest time to train at 10 days. This model and Model B (RF without cross validation) had very comparable results, e.g., similar sensitivities, specificities. In addition, each of their accuracies were within 95% CI of one another. Due to the similarities of the results of these two models and the large discrepancy in their computation time, Model B is the preferred model here. As a follow-up, it would be interesting to see whether Model E's training time could be reduced enough through distributed computation scheme that would make it the preferred model over any RF model.

REFERENCES

Boser, B. E., Isabelle M. Guyon , Vladimir N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. 1992.

Breiman, Leo. "Random Forests". Machine Learning 45 (1): 5–32. 2001.

Cortes, C. and Vapnik, V. Support-Vector Networks. Machine Learning, 20, 273-297. 1995.

Fallon, B. O., Whitney Wooderchak-Donahue, and David K. Crockett, A support vector machine for identification of single-nucleotide polymorphisms from next-generation sequencing data, Bioinformatics, Vol. 29 no. 11 2013, pages 1361–1366

Fawcett, T. An introduction to ROC analysis. Pattern Recognition Letters 27 (2006) 861–874

Hasibuan, L.S. Kusuma, W.A. ; Suwamo, W.B. Identification of single nucleotide polymorphism using support vector machine on imbalanced data. Advanced Computer Science and Information Systems (ICACSIS), 2014 International Conference. 375 – 379.

Rong Li, Jing Wu, Liwen Xiong, and Baohui Han. Lung cancer and benign lung diseases in patients with serious vitamin D deficiency in eastern China. Thoracic Cancer Volume 3, Issue 4, pages 303–306, November 2012.

Teknomo, Kardi. (2009) Tutorial on Decision Tree.
http://people.revoledu.com/kardi/tutorial/DecisionTree/

Yoo, W., et al., A Comparison of Logistic Regression, Logic Regression, Classification Tree, and Random Forests to Identify Effective Gene-Gene and Gene-Environ. International Journal of Applied Science and Technology Vol. 2 No. 7; August 2012.

APPENDIX

Attached Files

File	Description
`alldata_+genes.csv`	Input data file
`projModels_final.Rmd`	R working file for building classification models
`parsePos.R`	R script for parsing each file for chromosome number, and start and end position
`predDataAddGenes.R`	R script for parsing each UCSC output files for gene names, which were appended to the data set to be used for classification
`mergFiles.R`	R script for combining all the preprocessed data files into one .csv file