CPSC 68 Final Report: Comparative Performance Analysis of Key Classification Algorithms on Microarray Data

Robert Hwang RHWANG1@SWARTHMORE.EDU
Swarthmore College, 500 College Ave, Swarthmore, PA 19081 USA

Michael Song MSONG2@SWARTHMORE.EDU
Swarthmore College, 500 College Ave, Swarthmore, PA 19081 USA

Abstract

Selection and classification of gene expression data is a logistically and computationally intensive task, which makes the problem an ideal candidate for machine learning algorithms. K-nearest neighbors, random forests, and support vector machines are among the most studied and analyzed for the selection and classification of such expression data, specifically microarray data. Several studies have battled over which algorithms may be viable in gene selection and classification, while continuing to push the current boundaries of data science. We investigate the relative performances of three widely implemented supervised machine learning algorithms on three microarray gene expression datasets. Support vector machine classifiers far outstripped random forest classifiers and k-nearest neighbors when run on a robust breast cancer gene expression datasets, but all three classifiers performed poorly on a much less robust multiclass mosquito dataset, which contained approximately 20 percent of the data contained in the previously mentioned dataset. The results of our study are relevant to current debates in the machine learning community on how to determine algorithmic optimality, and they also hold significant consequences for data analysis in a wide range of disciplines and studies, not limited to clinical trials, evolutionary biology and image classification.

CPSC 68 Bioinformatics Final Report Proceedings, Swarthmore College, Fall 2014.

1. Introduction

Machine learning is a powerful tool in the current data analysis arsenal, and its applications are spread across many different scientific disciplines and fields of study. Specifically, classification and clustering algorithms have emerged as significant helpers for identifying strong correlations, associations, and separations in large datasets, especially for gene expression data(Li et al., 2001). Microarrays have emerged as the most cost effective and efficient tool for the measurement of the expression levels of specific genes (R. & de Andrs S., 2006). The ultimate goal for gene selection and classification is to find the smallest set of genes that share functional value, as this will reveal the most information into how the gene operates at the macro level in the organism (R. & de Andrs S., 2006). Three algorithms have been extensively examined as possible contenders for optimal gene expression data classification- K-nearest neighbors, random forests, and support vector machines. Prior research has strongly established argumentative bases for each algorithm, as the focus for all of these studies has been algorithmic applications to diagnostic datasets (I. et al., 2002). However, it is vital to see and understand the comparisons that can be drawn. Intuitively, machine learning methods that seek to classify gene expression data should ideally be unsupervised, as there is no prior functional knowledge of unknown genes to assign labels, given the large size of genomic datasets. While this is true, supervised learning methods have validity as equal, if not better, in use for clustering similar genes hierarchically(Brown et al., 2000). In Schipp 2002, supervised machine learning methods were successfully utilized to predict clinical outcomes of diffuse large B-cell lymphoma by clustering gene expression profiles (Shipp et al., 2002). Due to a wide range of possible clinical and molecular variables, selecting the best possible algorithm is essential for cancer diagnosis at the genomic level (S. et al., 2002).

As the complexity of models and computational ability increases, overfitting becomes a much more relevant problem. General consensus on which supervised learning algorithm is the most effective for gene expression classification is unclear. A comparative study performed by Dudoit in 2002 found that the nearest-neighbor classifier and other simple classifiers were the best performing for lymphoma and leukemia datasets (S. et al., 2002). A more recent, but just as thorough study suggests that random forest classifiers are heavily outperformed by support vector machines, in terms of microarray-based cancer classification (Statnikov et al., 2008). Applying these algorithms to unique datasets will reveal the performance of each algorithm in terms of accuracy and efficiency relative to each other. In this study, we used gene expression data from colon cancer patients, breast cancer patients, and malaria infected mosquitos, and used 5-fold cross validation to separate the data into training and testing sets. A confusion matrix was calculated for each test, and the recall score, F1 score, and precision were recorded. Given the exhaustive grid search implemented to select the most optimal parameters for each algorithm, and the findings outlined in Statnikov 2008, we hypothesize that support vector machines will outperform random forests and k-nearest neighbors due to the extensive parameter tuning in conjunction with the variation among the datasets being classified.

2. Methods

2.1. SciKit Algorithm Selections

All libraries and documentation for the machine learning algorithms used in this study were obtained from scikit.learn. The algorithms were chosen in line with our original experimental goal for examining the efficiency and effectiveness on various classification algorithms. The Random Forest algorithm is an example of ensemble learning while SVMs classify using the Kernel Trick and Max-Margin Hyperplanes. K-Nearest Neighbors is a much more simplistic algorithm that serves as a solid ground basis for comparison.

2.2. Input Data and Microarray Datasets

Gene expression datasets were obtained from previous studies, which include colon cancer, breast cancer, and mosquito infection. The data sets were chosen because they are different sizes and have a various number of labels. The size of the colon cancer and mosquito data sets are much smaller than the Breast Cancer data set. The colon cancer data set is a binary classification case taken from Alon 1999, a study that aimed to recognize patterns of gene expression among colon cancer patients (U. et al., 1999). The breast cancer dataset is a large multilabel dataset taken from a study conducted by The Cancer Genome Atlas Network that aimed to identify molecular signatures that were indicative of breast cancer subtypes (Network, 2012). The mosquito dataset is a smaller multilabel dataset taken from Dimopoulos 2002, a study that sought to do a complete cross comparison of $invivo$ and $invitro$ gene expression responses to environmental stressors, particularly oxidative stress and infection (Dimopoulos et al., 2002).

2.3. Microarray Data Parsing

Parsing algorithms were implemented to create Comma Separated Value files for each dataset, and a separate CSV file was created to contain the labels for all indexed data. The library modules are listed at the top of the algorithms.py file, and all code modules are commentated, including the parsing algorithm for the datasets. Each of the data sets were read in and placed in two two-dimensional NumPy arrays. One array would contain the x_i values or raw data, and the other would contain the y_i values or the labels for each x_i value. Our parsing methodology was taken from the corresponding papers directly. We did not have to preprocess any of the data except for the Breast Cancer data set. Many values were filled in as null but we were able to change the null values to 0 instead.

Once all of our data and labels were placed in NumPy arrays, we had to find the optimal classifier parameters to use in cross validation using SciKit modules. In order to find the best classifier parameters, we perform an exhaustive grid search on the available parameters. We then pass in the classifier with the best parameters into SciKits cross_val_score function, which returns a confidence interval. The function splits the data, fits the model, and computes the accuracy by percentage 5 consecutive times with different splits each time. In addition to the accuracy, we also compute a confusion matrix and a recall score. We obtain the classifier predictions through cross validation by using the cross_val_predict function. Specific details of the settings tuned for each algorithm are listed in the subcategories below.

2.4. Algorithms Tested

2.4.1. K-Nearest Neighbors

K-Nearest Neighbors is a classification algorithm that uses supervised machine learning. Given a training set with the corresponding labels, we can predict the label of each test point based on its k nearest points which can be calculated using a Euclidian Distance. Each of these k points votes based on a certain voting scheme. The voting scheme can vote based on the majority labels among the points, or give more weight to the points closer to the test point. Now using the final votes, the classifier predicts the label of the test point.

If we chose a majority voting scheme, a larger K will avoid overfitting but will underfit the data instead. However, this is mitigated by a weighted voting scheme because points closer to the test point will be weighted more. This, however, creates another problem with the runtime. With a large K, the algorithm will have an intensive runtime and use up large amounts of memory because it needs to calculate the distances from the K points to the test point.

This is the Majority Voting Scheme. The equation computes how often we see each label in the k nearest points.

$$p_i = \arg\max_c \sum_{j \in knn} I(y_j == c) \tag{1}$$

This voting scheme is weighted. It incorporates the distance between the test point x_i and the neighbor point x_j.

$$p_i = \arg\max_c \sum_{\{x_j, y_j\} \in knn} I(y_j = c) \times \frac{1}{1 + dist(x_j, x_i)} \tag{2}$$

+ Simplicity: only need to pick a K and distance function
+ Generalizes to multiclass problems and regression
– Intensive testing time and memory intensive

2.4.2. Random Forest

The Random Forest algorithm is a type of ensemble learning. In order to reduce overfitting, the algorithm makes use of Bagging (Bootstrap Aggregating). This is the idea where we take multiple random resamples for the data set by replacement, and then train strong classifiers on each of these samples. Each classifier is then given a vote, which reduces overfitting.

In Random Forest, we take several resamples and train a decision tree based on the best splits from a subset of a random splits for each resample. From the ensemble of random decision trees created, we were able to predict the label of the test point based on an ensemble voting scheme. In the exhaustive grid search, we chose to tune the number of decision trees to represent the classifier and the max features. Tuning the max features has significant implications on runtime and performance, since splitting on too many features may leading to overfitting, while splitting on two few features will return poor classification results.

+ Runs efficiently on large databases
+ Handles thousands of input variables without variable deletion.
+ Gives estimates of what variables are important in the classification
– Overfits for some datasets with noisy classification/regression tasks

Algorithm 1 Random Forest Training Phase

Input: data x
Initialize $numClassifiers = 0$
repeat
  $resample$ = bootstrap(x)
  for max_depth iterations **do**
    choose a random feature
    choose the best split on that feature
  end for
  add tree to ensemble
until $numClassifiers$ reached

2.4.3. Support Vector Machines

Support vector machines are a supervised machine learning method that selects a maximum margin separating hyperplane in an appropriately chosen feature space, separating the data into positive and negative classes . Locating the hyperplane with the most room for error is done by finding the parallel separating hyperplanes with the largest distance between them and taking their midpoint(Brown et al., 2000). If the data is not linearly separable, then it can be mapped into a higher-dimensional space by choosing a different basis. In order to keep computation load reasonable, SVMs use a kernel function to define the mapping from one basis to another. Another way is to use a soft margin classifier. The algorithm would allow some misclassifications, but a penalty term would be added based on Equation 3.

+ Effective in high dimensional spaces
+ Memory Efficient (only uses subset of training pts)
+ Versatile: different Kernel functions
– Likely to give poor performances for num of features greater than sample size.
– Do not directly provide probability estimates

$$Penalty = \lambda(\frac{1}{margin})^2 + \sum_{x \in misclassified} dist(x, hyperplane) \tag{3}$$

2.5. Cross Validation Experimental Design

Each dataset will be split and tested for each algorithm using 5-fold cross validation. Algorithms that have consistently large error bars relative to the others will be classified as more inconsistent, while consistently small error bars relative to the others will be classified as more consistent. A confusion matrix, alongside error approximations for accuracy will be recorded for each algorithm, alongside the f-score, recall score, and precision.

3. Results

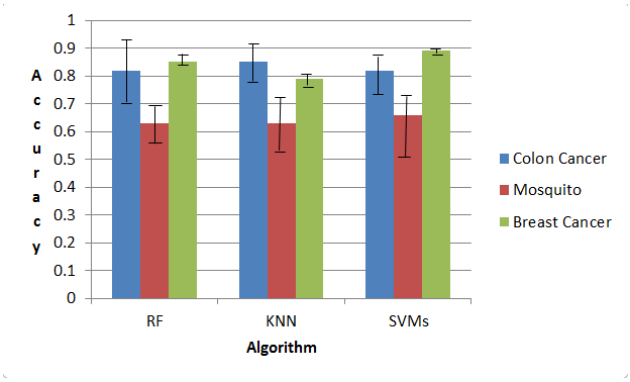

Figure 1. The accuracy of random forests, k-nearest neighbors, and support vector machines was calculated when tasked with classifying three different microarray datasets. Colon cancer dataset was binary classification, and mosquito and breast cancer datasets were multiclass classification. Each error bar represents one standard deviation.

3.1. Algorithm Accuracies

The performance result of each algorithm are shown in Figure 1. Overall, support vector machine classifiers had the least error when run on the colon cancer dataset and the breast cancer dataset, (0.82 +/- 0.13 and 0.89 +/- 0.01 respectively). According to the confusion matrices, all three algorithms performed poorly when run on the mosquito gene expression dataset, a far less robust multiclass dataset relative to the breast cancer dataset (Figure 2). Specifically, clusters 1A and 1B in the mosquito dataset are similar in that they share an in vivo property, and according to the numerical data on true positives and false positives from the confusion matrix of all three algorithms , all have difficulty distinguishing these two clusters, especially random forests (Figure 2). However, the most difficult classification task appeared to be the M label, which rarely peaked over half of true positives (Figure 2).

3.2. Classification Report

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

In Table 1, we present the average values for the recall score, precision, and F1 score from each algorithm, separated by dataset. These averages were taken from the sum of the label scores, divided by the total number of labels. The F1 score, which presents a weighted average of the precision and recall scores (1 is the best, 0 is the worst), is much lower for all three algorithms when run on the mosquito dataset (Equation 6). Similar to the implications from the confusion matrices, these values further support that these algorithms performed relatively poorly. The score composition appears to have been reversed as well, when comparing the mosquito dataset results to the breast cancer and colon cancer dataset results. Whereas the average precision is consistently higher compared to the average recall in the breast cancer and colon cancer datasets, the opposite is true for the mosquito dataset, meaning the ratio of true positives to all predicted positives is higher in the more robust datasets, and the ratio of true positives to all actual positives is higher in the less robust dataset (Equations 4 and 5).

4. Discussion

4.1. Support Vector Machines Performed Best

We compared the relative performance of different machine learning classification algorithms on microarray datasets of varying robustness. While support vector machines performed the best over random forests and k-nearest neighbors on our largest and most complete dataset, all three algorithms had poor performance on the less robust multiclass dataset. Support vector machines required extensive grid parameter tuning compared to random forests and k-nearest neighbors causing the runtime to be significantly slower, especially when running the classification on the large microarray dataset. We present possible explanations as to the tradeoffs involved in algorithmic selection, and we propose that selecting the best algorithm for the given characteristics of a microarray gene expression dataset is extremely complicated and has many facets that include, but are not limited to parameter tuning and dataset characterization.

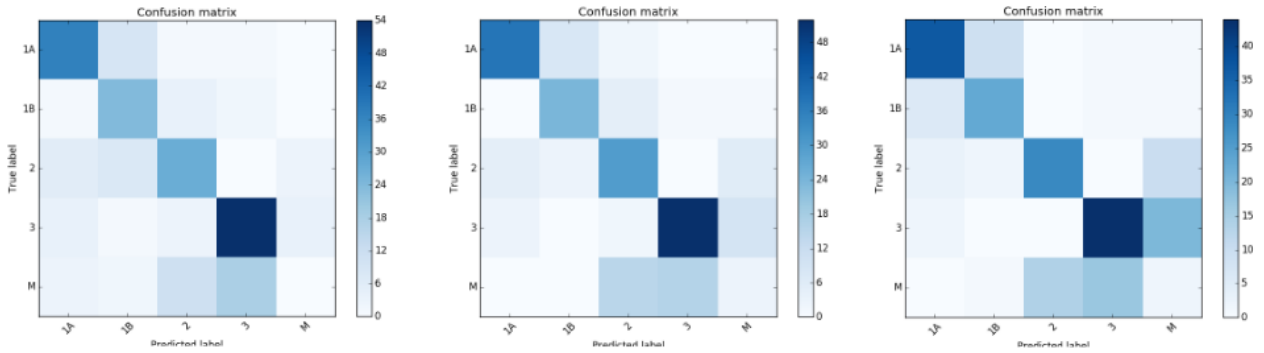

Figure 2. The confusion matrices of all three algorithms on the mosquito dataset. The horizontal axis is predicted labels and the vertical axis is true labels. (From the left: SVM, KNN, RF)

Table 1. Average classification report metrics of random forests, k-nearest neighbors, and support vector machines on microarray datasets.

TEST	AVG PREC.	AVG RECALL	AVG F1
KNN/COLON	0.86	0.82	0.83
RF/COLON	0.79	0.78	0.78
SVM/COLON	0.82	0.79	0.80
KNN/MOSQUITO	0.52	0.59	0.55
RF/MOSQUITO	0.57	0.58	0.58
SVM/MOSQUITO	0.60	0.62	0.61
KNN/BREAST	0.84	0.72	0.74
RF/BREAST	0.90	0.81	0.83
SVM/BREAST	0.89	0.88	0.88

4.2. Model Selection and Parameter Tuning

Currently, there exists disagreement in published literature on whether random forests or support vector machines are more sensitive to poorly optimized tuning. It is generally agreed that these two algorithms are preferred for classification tasks on more complex and robust datasets over the K-nearest neighbors algorithm, but classification performance has been largely measured by the viability of each algorithm on

binary and multicategory tasks (Statnikov et al., 2008). Interestingly, the K-nearest neighbors algorithm performed the best on the binary classification task, with high scores on the classification report on all three metrics. This may be due to less possibility of overfitting, due to the simplicity of the model compared to random forests and support vector machines. Even though we tuned the support vector machine classifier parameters with a linear kernel alongside the RBF kernel, K-nearest neighbors may generalize better to binary data. Our results support the notion that support vector machine classifiers are best suited for large complex multiclass datasets, but we cannot discount the random forest algorithm. While the number of classifiers involved in ensemble learning has been seen to greatly impact accuracy, more studies must be done to validate this concern (Bonab & Can, 2016). Random Forest and KNN are related in that theyre both weighted neighborhood schemes, but support vector machines rely much more on the support vectors that make up the hyperplane.

5. Conclusion

Overall, we were able to complete a comprehensive comparison of the potentials of three unsupervised machine learning classification algorithms to complete classification tasks on binary and multiclass gene expression data. Our original hypothesis that support vector machines would 1) Require the most tuning and 2) Perform the best was partially correct, but the K-nearest neighbors algorithm performed similarly, if not better than the support vector machine classifier on simple binary data. In general, the support vector machine classifier surpassed the random forest algorithm and the K-nearest neighbor algorithm when dealing with the significantly larger and more complex multiclass dataset, as shown through smaller error bars reported by cross validation, and higher performance metrics in terms of accuracy, precision, recall score, and F1 score. The results from this study suggest that there is no specific algorithm that will generalize well to all classes and sizes of microarray data. Until future studies show that a best algorithm exists, it is best to choose a classification algorithm based upon a holistic overview of the type and quality of data being classified.

6. Future Directions

The efficiency of our parameter tuning could be increased to fit the testing of a wider range of parameters with similar, if not better runtime performance. Alongside the exhaustive grid search function, SciKit provides a randomized grid search, which unlike the exhaustive grid search, does not attempt every possible parameter specified. Instead, it samples by taking a fixed number of parameter settings and samples with replacement. This would allow for a much larger parameter search space, while maintaining the goal of maximizing performance on the testing data. It was evident in our experiments that parameter tuning makes a significant difference for performance. While some prior studies advocated for default settings for some of these algorithms, it was generally better to tune in order to recognize the full performance potential. While we extensively tuned our support vector machine classifier, we could improve on the tuning for our random forest classifier. In addition to max features and number of trees, we could attempt to tune minimum sample leaves, since a small number of leaves is much more sensitive to noise in the data. This is especially important for large datasets, which may be significant in the context of the breast cancer dataset we examined.

Acknowledgments

We would like to thank Professor Soni for his guidance and mentorship throughout this project. We would also like to acknowledge the Cancer Genome Atlas Network, the individuals involved in Alon 1999, and the individuals involved in Dimopoulos 2002 for providing the data that was used in this study.

References

Bonab, Hamed R. and Can, Fazli. A theoretical framework on the ideal number of classifiers for online ensembles in data streams. *Proceedings of the 25th ACM International on Conference on Information and Knowledge*, pp. 2053–2056, 2016.

Brown, M. P. S., Grundy, W. N., Lin, D. Cristianini, Sugnet, N., W., C., Furey, T. S., and D., ... Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262–267, 2000.

Dimopoulos, George, Christophides, George K., Meister, Stephan, Schultz, Jrg, White, Kevin P., Barillas-Mury, Carolina, and Kafatos, Fotis C. Genome expression analysis of anopheles gambiae: Responses to injury bacterial challenge and malaria infection. *Proceedings of the National Academy of Sciences of the United States of America*, 99(13):8814–8819, 2002.

I., Guyon, Weston, J., and et al, Barnhill S. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(389): doi:10.1023/A:1012487302797, 2002.

Li, L., Weinberg, C. R., Darden, T. A., and Pedersen, L. G. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17(12):1131–1142, 2001.

Network, The Cancer Genome Atlas. Comprehensive molecular portraits of human breast tumors. *Nature*, 490 (7418):61–70, 2012.

R., Daz-Uriarte and de Andrs S., Alvarez. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(3):doi: 10.1186/1471–2105–7–3, 2006.

S., Dudoit, J., Fridlyand, and P., Speed T. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

Shipp, Margaret A., Ross, Ken N., Tamayo, Pablo, Weng, Andrew P., Kutok, Jeffery L., Aguiar, Ricardo C. T., Gaasenbeek, Michelle, Angelo, Michael, Reich, Michael, and et al, Geraldine S. Pinkus. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(68):68–74, 2002.

Statnikov, A., Wang, L., and F, Aliferis C. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *Bioinformatics*, 9(319):1–10, 2008.

U., Alon, N., Barkai, A., Notterman D., K., Gish, S., Ybarra, D., Mack, and J., Levine A. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, 1999.