Identifying Significant Genes for Distinguishing Acute Leukemias

Peter Jourgensen
UCLA

1 Introduction

Classification of cancer tissue cells based on microarray expression levels has been of of great interest over the last 30 years. The struggle often resides in the fact that datasets tend to contain few samples in comparison to their number of features. Some approaches tend towards optimizing a model for a given selection of features, while others tend towards optimizing the selection of features before giving them to a model. There are many varying approaches to selecting genes, but a common goal remains to estimate the effect of gene interactions on final classification. With a high feature space, however, this results in a combinatorial explosion of interactions and too small a sample space to accurately estimate the significance of a given interaction. In this report, I discuss a new approach to selecting significant genes based on their interactions with others. I begin by applying an ensembling method to a neural network base model, whereby random subsets of training data are passed through the base model to ultimately make predictions on a held out set. The accuracies of the models are then transformed into weights, which can be used to "score" each individual gene. The hypothesis is that combinations of high scoring genes will serve as strong predictors for cancer class. Finally, I take various selections of the highest scoring genes and train a KNN model for prediction. Using the data from "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring" (Golub et al. 1999), this approach is able to achieve 97.2% classification accuracy on the collection of patients.

2 Methodology

The data I chose to work with involves classifying human acute leukemias from gene expression levels. It consists of 72 patients and 7,129 genes with quantified expression levels. I began by splitting the data into a training set of 38 patients and a testing set of the remaining 34 patients.

An ensembling method generally involves training a collection of weak learners on subsets of the data before weighing each individual learner and having them vote on final predictions. I chose to work with a densely connected neural network as a base model because it would be capable of capturing nonlinear interactions among a multitude of features. Each model was trained on a random selection of 30 samples and 50 genes. Overfitting is often a concern with neural networks, however, so proper parameterization is key. First, the number of hidden layers and the number of nodes within each layer were kept small. I used 3 hidden layers with descending dimensionalities of 25, 10, and 5 before reaching the output layer. Second, a high dropout rate was instituted for increased regularization. Dropout is a common tool used to reduce codependency among neurons and create a network with more robust features. Finally, the number of epochs was kept small. This means that the network gets a brief look at the data before making its final assessment. With the comparatively small subsets of training data and the parameterizations of the network set to combat overfitting, each model is intended to learn a small truth about the underlying data, with the collection of models ideally encapsulating the whole truth. More models should provide more information to the structure of the population, but I chose to continue with the analysis with a total of 4,680 models trained.

The next step in the procedure was determining how to weight each individual model. For this, a few approaches were tested for efficacy in final class predictions. The predictions of each model were labeled ALL/AML (+/- 1). These predictions would be multiplied by their weights and summed across all models. If the net sum was positive, that patient would be labeled as having ALL, while if it was negative, they would be labeled as having AML. The results and conclusions of these tests will be discussed in the following section of this report. The first weighting framework was to give every model, independent of accuracy, an equal vote on final prediction. The second framework used the model's accuracy as its weight. For the third framework, I had observed that nearly half of all models only predicted the majority class (ALL) for every patient, which produced an accuracy of 58.82%.

Therefore, I subtracted this quantity from each accuracy and passed the result through a Rectified Linear Unit activation function. This gave 0 weight to models below this threshold and a linearly increasing weight to those above it. Finally, I further developed the third set of weights by passing the values greater than 0 through an exponential function. This served to give slightly higher weight to more accurate models when compared with its linear counterpart. Ultimately, the fourth set of weights was chosen because it produced the highest overall accuracy.

The next step involved scoring the individual genes based on these weights. If a gene was used in a model, that model's weight would be added to the gene's score. The gene's score would be averaged across the number of times it appeared to normalize for different number of appearances. Only genes with scores significantly greater than 0 ($p < 0.05$) would be considered. Finally, the genes were sorted descending by score.

To test the efficacy of the scoring system, I simply took a KNN model and performed a grid search over the number of highest scoring genes and the number of neighbors in the model to optimize the parameters. 5-fold cross validation on the training set was used to measure model performance. I then tested the parameterization with the highest cross validation score on a held out testing set and the entirety of the data as well.

3 Results

As previously mentioned, I trained a total of 4,680 individual neural networks. With 50 genes selected at random for each model, each gene appeared in 33 distinct models on average. In regards to the performance of the individual models, the majority performed poorly by either only predicting the majority class or seemingly guessing at random. This was not a surprise, however, given that the majority of genes play limited role in distinguishing between acute leukemias and that the individual models were not optimized for performance. In summary, 49% predicted the majority class exclusively and 29% performed worse, leaving 22% [1] that appeared to have insight into the structure of the data.

The next step involved assessing which set of weights to choose for scoring the genes. This was done by comparing the ensemble accuracy of the 4 weighting options. Giving equal weights to each model resulted in always predicting the majority class, resulting in an accuracy of 58.82%. The outcome was the same for weighting each model by its accuracy. The next system involved only considering models above that threshold, which resulted in an accuracy of 82.21%. And finally, these accuracies above the threshold were passed through an exponential function to give slightly more bias to higher performing models. This resulted in an ensemble accuracy of 85.29%. This would be the final selection to be used for gene scoring.

[1] 1005 total models

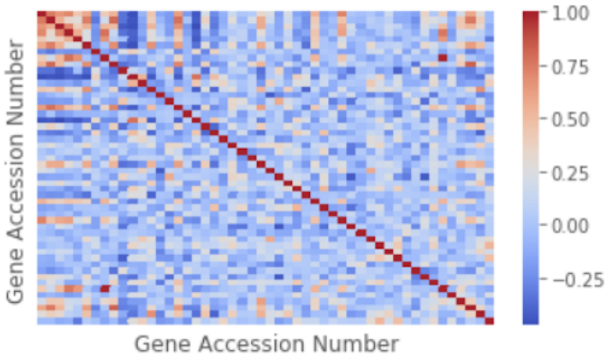

Figure 1: Heatmap illustrating correlations among 50 highest scoring genes

I then followed the previously discussed procedure for scoring the individual genes and limited further analysis to the top 50 genes. Among these genes, only 7 were among the top 50 most correlated with the class distinction. Further, **Figure 1** is a heatmap of the correlations between each of the top 50 highest scoring genes. While some of the genes appear coregulated, most only share weak relationships. This is a key result with respect to feature selection. Highly correlated features, while they all may be good individual predictors of a target class, do not provide additional information when used in combination. Weakly correlated features provide broader insight into the data when used together. However, this is contingent upon their efficacy in predicting a target class.

To test the efficacy of the gene scoring system, I optimized a KNN model for both the number of highest scoring genes to use and the number of neighbors to compare. After a grid search of the parameters with cross validation serving as the performance metric, it was determined that using the combination of 2 neighbors and the top 50 genes was optimal. When tested on a held out set, it achieved an accuracy of 94.1% [2]. It predicted the training set perfectly, giving an overall accuracy of 97.2% [3].

4 Conclusion

The success of the predictions of the KNN model justifies the use of bagging neural networks for feature selection. While assessing significance of singular genes via correlation or information gain has seen some success, cancer classification models have seen improvement when selecting features based on gene pairs or even triplets. The method proposed here does not directly score gene interactions, but it does effectively score each gene for its significance when interacting with

[2] 32 correct out of 34

[3] 70 correct out of 72

other genes. Genes that score highly fall into 2 categories: They are either strong individual predictors or they are weak, but provide key additional insight when combined with other genes. For future research, optimizing the neural networks and training more of them would be beneficial in improving the quality of the gene scores. It may also be of value to take a subselection of the highest scoring genes, compute their products for each pair and follow a similar analysis with the new set of features.

References

[1] Zhuo Sheng *Boosting and Bagging Neural Networks with Applications to Financial Time Series.* 2006.

[2] Chopra et al. *Improving Cancer Clasification Accuracy using Gene Pairs* 2010.

[3] Zhang et al. *Improving Accuracy for Cancer Classification with a new Algorithm for Gene Selection* 2012.

[4] Golub et al. *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring* 1999.