Machine Learning to Classify Lung Cancer Tumor Cells from Gene Expression Data

KELLY ZHANG, Massachusetts Institute of Technology

Due to the complex nature and high dimensionality of gene expression data, detection of cancer from gene expression data continues to pose a challenge in the field of medicine. After decades of research, the exact biomarkers of cancer are still unknown and prediction methods lack certainty [1]. Here we focus on the classification of lung cancer tumor cells based on 21 candidate genes, proposing and comparing two machine learning models. First, we used principal component analysis to reduce the dimensions of our gene expression data. Then, we trained and tested multiple support vector machines with different kernels on various combinations of principal components. Finally, we implemented a neural network and compared it to the support vector machine models. Our results and analysis demonstrate that machine learning can be used to detect and classify lung cancer tumors from gene expression data.

Additional Key Words and Phrases: bronchus and lung cancer, tumor cells, normal cells, principal component analysis, support vector machine, neural network

ACM Reference Format:
Kelly Zhang. 2019. Machine Learning to Classify Lung Cancer Tumor Cells from Gene Expression Data. 1, 1 (May 2019), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The rapidly increasing quantity of gene expression data provides a wealth of information from which we can make significant scientific and medical discoveries. Tremendous potential exists for computational methods to analyze the data for disease detection and prediction. Specifically, examination of gene expression data can indicate cell state, providing the ability to partition cancerous and normal cells [1]. Multiple algorithms already exist to distinguish normal cells from abnormal cells using gene expression, but these methods generally lack accuracy and certainty as high dimension and noise pose major problems for this area of exploration [2].

To overcome these obstacles, we propose using dimensionality reduction and machine learning techniques. Additionally, we focus our efforts on lung cancer, the leading cause of cancer death, where the following twenty-one candidate genes have been identified: APEX1, CHRNA3, CXCR2, CYP2E1, HYKK, ATM, CLPTM1L, CYP1A1, AXIN2, PON1, REV3L, CHRNA5, ERCC2, SOD2, FGFR4, OGG1, TP53, TERT, ERCC1, MIR146A, and MIR196A2 [3]. Analyzing these candidate genes significantly decreases the dimensions of our data, narrowing the scope of our feature space and allowing focus on genes that are the most influential.

In this work, we implemented and compared two machine learning approaches for cancer detection and classification from gene expression data. The first method reduces the dimensions of the gene expression data using principal component analysis (PCA) and then performs classification on the reduced data with a support vector machine (SVM). The second method performs both dimension reduction and classification via a neural network. Analysis and comparison of these two models allow us to classify lung cancer tumor cells and normal cells for potential medical applications such as disease detection and prediction.

Author's address: Kelly Zhang, kellyz@mit.edu, Massachusetts Institute of Technology.

XXXX-XXXX/2019/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DATA AND METHODS

2.1 Data Collection

We analyzed RNA-seq transcriptome profiling gene expression data for bronchus and lung cancer from The Cancer Genome Atlas (TCGA). TXT files were parsed for 1146 patient samples to collect gene expression data for the 21 candidate genes based on Ensembl ID. 1038 samples were tumor cases, and 108 samples were normal tissue cases.

2.2 Classification

Two machine learning models were tested and compared:

(1) Principal component analysis dimensionality reduction paired with support vector machine classification
(2) Neural network classification

2.2.1 Metrics for Model Analysis. Accuracy, true positive rate, true negative rate, false positive rate, and false negative rate were reported for both models using the following definitions:

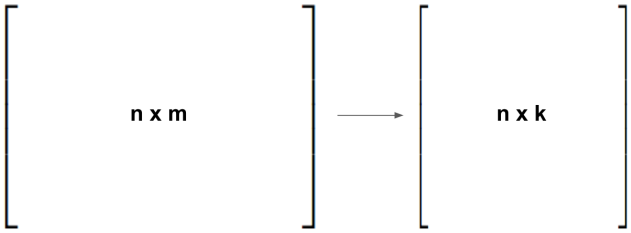
- accuracy (acc): ratio of correct predictions to total predictions made
- true positive rate (TPR): ratio of correct tumor classifications to actual tumor labels
- true negative rate (TNR): ratio of correct normal classifications to actual normal labels
- false positive rate (FPR): ratio of incorrect tumor classifications to actual normal labels
- false negative rate (FNR): ratio of incorrect normal classifications to actual tumor labels

2.2.2 Principal Component Analysis and Support Vector Machine. Principal component analysis was implemented using a Python library in the `sklearn` package. PCA is a dimensionality reduction technique for feature extraction that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components, which are linear combinations of the initial feature

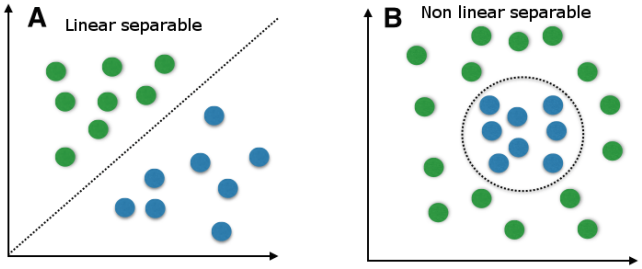
space. The first principal component is calculated such that it accounts for the greatest possible variance in the data set. This implies feature extraction since variables with greater variance will receive more weight in the linear combination. The second principal component is calculated in the same way with the constraint that it is uncorrelated, or perpendicular to, the first principal component and that it accounts for the next highest variance. This continues until a total of k principal components have been calculated. Thus, PCA converts an n x m matrix of n samples with m features into an n x k matrix where $k < m$ and k is the number of principal components. In this study, we used PCA to reduce the feature space from twenty-one candidate genes to three principal components. The reduced data was divided into eighty percent training data and twenty percent testing data. For data visualization purposes, the three principal components were plotted on a three-dimensional plot, and the data points were color-coded by label. Since PCA is a linear transformation, the labels of "tumor" and "normal" cells still apply to the reduced data.

A support vector machine was then trained on the subset of training data using a Python library from the `sklearn` package. An SVM is a supervised machine learning model that learns from a training data set to determine the optimal hyperplane that will separate the training data and correctly classify new data. Although SVMs learn linearly, they can perform non-linear classification using the kernel trick. We train SVMs using both the radial basis function (rbf) kernel and linear kernel.

Following SVM, cross-validation was performed using the testing data, and comparison metrics were analyzed. The same process of PCA followed by SVM was done for all pairwise combinations of the first three principal components. Figure 1 outlines the architecture for this method.



(a) PCA dimensionality reduction of n x m matrix to n x k matrix where $k < m$. In this paper, $n = 1146$, $m = 21$, and $k = 3$.



(b) SVM separating labeled data based on A) linear kernel and B) radial basis function kernel.

Fig. 1. Two step method: PCA, then SVM

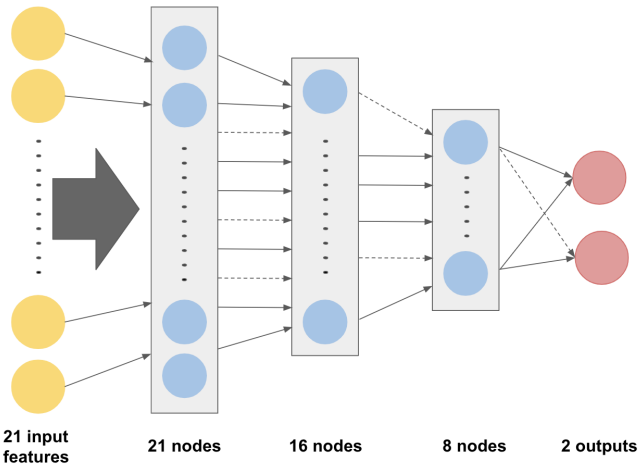


Fig. 2. Neural network architecture. Dotted arrows indicate connections being "dropped" to illustrate dropout layers.

2.2.3 Neural Network. We used a neural network with four fully connected (FC) layers. The first three FC layers were followed by ReLU activation and dropout layers while the last FC layer was followed by a softmax operation, constituting the output layer. The size of the fully connected layers were 21, 16, 8, and 2, respectively. Figure 2 illustrates this network architecture. Our neural network conducts feature extraction by learning the correct weights in its network to emphasize features that are more important to classification and disregard features that are less important. These decisions are passed through the network until the output layer, where the class probabilities are predicted for classification.

Cross entropy was used as the loss function with L2 regularization. We performed a hyperparameter grid search to find the optimal values of 0.2, 0.000001, and 0.001 for dropout rate, L2 lambda, and learning rate, respectively. We used cross validation to train the neural net and to test performance. During training, the batch size and number of epochs were set to 32 and 100, respectively. The data was split into seventy percent training data, ten percent validation data, and twenty percent testing data.

3 RESULTS

3.1 Principal Component Analysis and Support Vector Machine

Analysis of the previously defined model metrics demonstrates that support vector machines reported high accuracy, high TPR, low TNR, high FPR, and low FNR. Table 1 below shows the acc, TPR, TNR, FPR, and FNR of SVMs with radial basis function kernels trained on the listed combinations of principal components. Similarly, Table 2

shows the acc, TPR, TNR, FPR, and FNR of SVMs with linear kernels trained on the same combinations of principal components.

	Acc	TPR	TNR	FPR	FNR
PC1/PC2/PC3	0.9304	0.9904	0.3636	0.6364	0.0096
PC1/PC2	0.9391	0.9760	0.5909	0.4091	0.0240
PC1/PC3	0.9304	0.9856	0.4091	0.5909	0.0144
PC2/PC3	0.9043	1.0	0.0	1.0	0.0

Table 1. Acc, TPR, TNR, FPR, and FNR for SVMs with radial basis function kernel trained on the listed combinations of principal components.

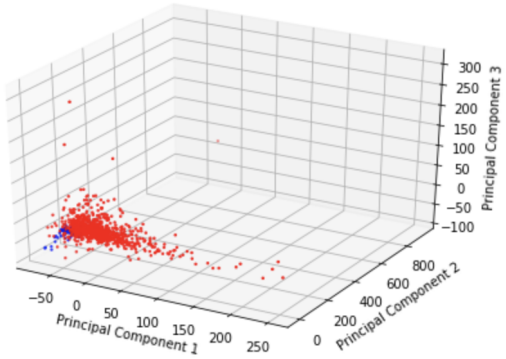
	Acc	TPR	TNR	FPR	FNR
PC1/PC2/PC3	0.8957	0.9904	0.0	1.0	0.0096
PC1/PC2	0.9043	1.0	0.0	1.0	0.0
PC1/PC3	0.8957	0.9904	0.0	1.0	0.0096
PC2/PC3	0.9043	1.0	0.0	1.0	0.0

Table 2. Acc, TPR, TNR, FPR, and FNR for SVMs with linear kernel trained on the listed combinations of principal components.

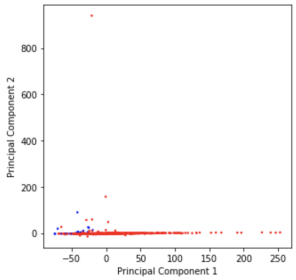
Comparison of tables 1 and 2 shows that SVMs with radial basis function kernels generally outperformed SVMs with linear kernels. For PC1/PC2/PC3, SVM with rbf kernel had an accuracy of 0.9304 while SVM with linear kernel only had an accuracy of 0.8957. Similarly, for PC1/PC2, SVM with rbf kernel had an accuracy of 0.9391 while SVM with linear kernel had an accuracy of 0.9043, and for PC1/PC3, SVM with rbf kernel had an accuracy of 0.9304 while SVM with linear kernel had an accuracy of 0.8957. Thus, the accuracies of SVMs with rbf kernels are higher than their respective counterparts with linear kernels with the exception of SVMs trained on PC2/PC3, which resulted in the same accuracy of 0.9043 when using a radial basis function kernel or linear kernel.

Although the accuracies are higher for SVMs with rbf kernels, SVMs with linear kernels reported higher TPRs and lower FNRs. However, there was a recurring problem of high FPRs across all SVMs, regardless of kernel type. The issue is more prevalent when using the linear kernel as the FPR is 1.0 for SVMs trained across all combinations of principal components. Consequently, the TNR is 0.0 for SVMs using the linear kernel, suggesting that all samples are being classified as tumor cells and no samples are being classified as normal cells. This indicates that the data may not be linearly separable. SVMS with rbf kernels have lower FPRs, with the lowest FPR being 0.4091 for PC1/PC2.

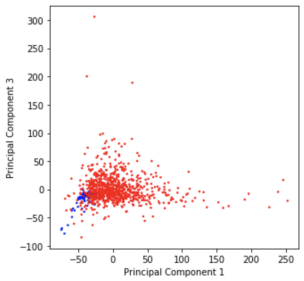
Plots of principal components are presented in Figure 3. Figure 3a plots the first three principal components in a three-dimensional plot. Figures 3b, 3c, and 3d plot pairwise combinations of the first three principal components in two-dimensional plots. Red points represent tumor cells while blue points represent normal cells. There appears to be no clear clustering of tumor cells and normal cells in the reduced data.



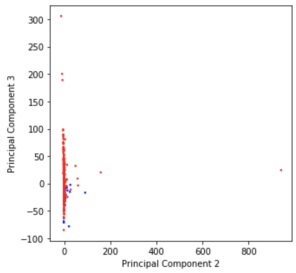
(a) PC1 vs PC2 vs PC3



(b) PC1 vs PC2



(c) PC 1 vs PC3



(d) PC2 vs PC3

Fig. 3. Plots of principal components with color-coded labels. Red points represent tumor cells, and blue points represent normal cells.

3.2 Neural Network

The same metrics of acc, TPR, TNR, FPR, and FNR were used to evaluate the performance of the neural net, as shown in Table 3. A high accuracy of 0.9737 was achieved. High TPR and TNR were reported at 0.9904 and 0.8182, respectively, while low FPR and FNR were reported at 0.1818 and 0.0096, respectively.

	Acc	TPR	TNR	FPR	FNR
Neural Net	0.9739	0.9904	0.8182	0.1818	0.0096

Table 3. Acc, TPR, TNR, FPR, and FNR for neural net.

The receiver operating characteristic (ROC) curve in Figure 4 illustrates the trade-off between sensitivty (TPR) and specificity

(1-FPR) in our neural network model. The area under the ROC curve is 0.99, indicating high performance.

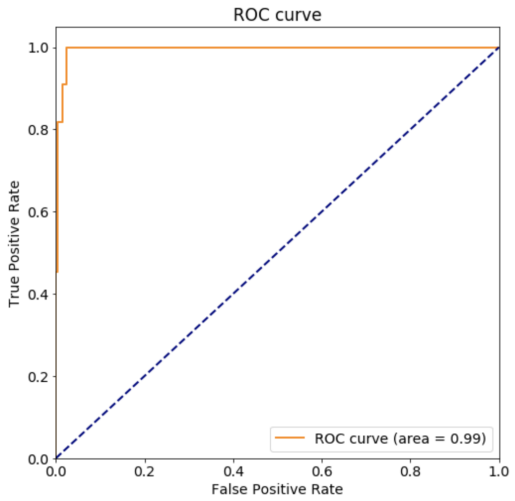


Fig. 4. Receiver operating characteristic (ROC) curve.

The precision-recall (PR) curve in Figure 5 displays the trade-off between precision (positive predictive value) and recall (TPR). Although the curve is jagged, the area under the curve is 0.95 and F_1 is 0.86, indicating high performance.

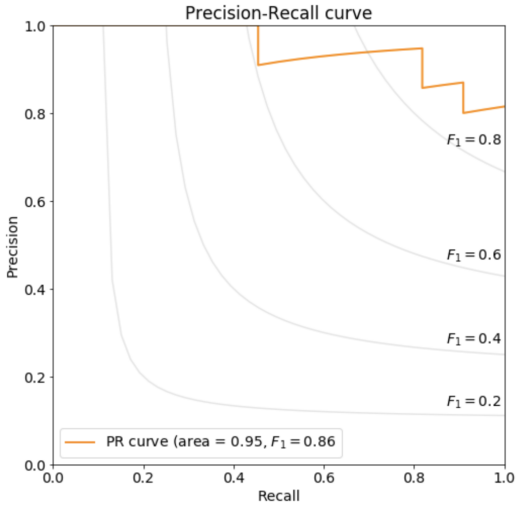


Fig. 5. Precision-recall (PR) curve.

4 DISCUSSION

The abundance of gene expression data provides the potential for significant progress in the classification and detection of lung cancer. We compared the classification power of PCA paired with SVM to a four-layer neural network. Based on the metrics of accuracy, TPR, TNR, FPR, and FNR, the neural network model is far superior to the SVM models for classification of lung cancer tumor cells. The neural network not only achieved the highest accuracy, but it also reported the lowest FPR. This is important to consider because we do not want to misdiagnose healthy individuals with cancer. The neural network also reported a low FNR, which is similarly important because we do not want to misdiagnose a sick individual as healthy. Although the SVM models also had low FNRs, they also had high FPRs, suggesting bias toward tumor labeling in the classification model. Most notably, SVMs utilizing linear kernels classified all samples as tumors. SVMs with rbf kernels performed better but not as well as the neural net. Thus, the neural net performed best, showing promise for the use of machine learning in tumor detection and potentially prediction.

One limitation to this study is the small data set. Although there are over 30,000+ cases on TCGA, only 1146 samples are RNA-seq data in TXT format from lung cancer patients. Other types of data and file formats were not usable given the resources available. Additionally, the data set was unbalanced. Out of the 1146 samples, only 108 samples were normal tissue cases. Thus, there were about ten times as many tumor cases as normal tissue cases.

Future work can be done to improve upon these models. We can explore techniques to handle imbalanced data as well as aim to acquire more RNA-seq samples as the TCGA database continues to grow. It is also of interest to apply these models to other specific cancers or even develop a pan-cancer model to classify any tumor based on gene expression data from a set of specific candidate genes. Finally, we would like to assess the interpretability of our neural network to determine which features were most important when making classification decisions.

5 CONCLUSION

In this paper, we implemented two machine learning models to classify lung cancer cells and normal cells and subsequently evaluated their performance. First, we performed principal component analysis paired with a support vector machine. PCA reduced the dimensions of our gene expression data while SVM classified the reduced data. SVMs with radial basis kernels performed better than those with linear kernels. However, SVMs generated a bias for tumor classification regardless of the kernel type. Then, we built and trained a neural network with four fully connect layers. The neural network reached 97.37% accuracy and outperformed all SVMs across all metrics of accuracy, TPR, TNR, FPR, and FNR. The success of the neural network for cancerous cell classification opens the door for future work in the exploration of deep learning for cancer classification, detection, and eventually, prediction.

REFERENCES

[1] Boyu Lyu and Anamul Haque. 2018. *Deep Learning Based Tumor Type Classification Using Gene Expression Data.* https://doi.org/10.1101/364323

[2] Padideh Danaee, Reza Ghaeini, and David A. Hendrix. 2016. *A Deep Learning Approach for Cancer Detection and Relevant Gene Identification.* In Biocomputing 2017. https://doi.org/10.1142/9789813207813_0022

[3] Junjun Wang, Qingyun Liu, Shuai Yuan, Weijia Xie, Yuan Liu, Ying Xiang, Na Wu, Long Wu, Xiangyu Ma, Tongjian Cai, Yao Zhang, Zhifu Sun, and Yafei Li. 2017. *Genetic predisposition to lung cancer: comprehensive literature integration, meta-analysis, and multiple evidence assessment of candidate-gene association studies.* Scientific Reports 7, 1. https://doi.org/10.1038/s41598-017-07737-0