Machine Learning Techniques for Thyroid Cancer Diagnosis

CS229 Final Report, Fall 2015

Chenjie Yang
Department Electrical Engineering
Stanford University
Stanford, CA
yangcj@stanford.edu

Jingtao Xu
Department Electrical Engineering
Stanford University
Stanford, CA
jiangtaox@stanford.edu

Tiffany Liu
Department Electrical Engineering
Stanford University
Stanford, CA
tiffliu@stanford.edu

Abstract— Drawing inspiration from Alexander's paper[1] on classification of thyroid cancer, we are interested in replicating and possibly improving the predictive results of a learning model for detecting thyroid cancer from gene expression data from thyroid nodules. This data set is the same data used in the paper by Alexander[1]. We will develop our own gene expression classifier by applying different feature selection methods and supervised learning algorithms to predict whether thyroid nodules are malignant or benign using gene expression data. With a L2 regularized linear support vector machine, we achieved a maximum predictive accuracy of 81.89%.

Keywords—machine learning techniques; principal component analysis; logistic regression; support vector machine

I. INTRODUCTION

Tumors found in thyroid cancer are commonly presented as thyroid nodules, but not all thyroid nodules are cancerous. About 5-15% percent of thyroid nodules can be malignant. Diagnosis of thyroid cancer is typically conducted by fine needle aspiration. However, 15-30% of aspirations yield indeterminate cytologic findings and lead to diagnostic thyroid surgery. Most of the people referred for diagnostic thyroid surgery prove to have a benign form of the disease after histopathological review.[2] Histopathological review classifies if a nodule is benign or malignant by examining whole tissues for abnormalities, whereas cytological review examines tissue fragments or cells for abnormalities. This kind of surgery, which is not necessary for those patients, exposes them to risk of serious surgical complications and causes patients to have levothyroxine replacement therapy for life.[1] Thus, improvement of diagnostic evaluation for patients with indeterminate cytologic findings is critically needed.

A. Related Work

Gene expression data is now readily available for many diseases and has been used extensively to develop classifiers to help physicians diagnose and treat the disease. Alexander's paper showed the potential for diagnosing thyroid cancer using gene expression data. The investigators extracted genes that are highly related to thyroid cancer and built a gene-expression classifier using linear kernel support vector machine[1] (LSVM). The model has a high sensitivity when classifying malignant samples, but benign samples are not classified with high accuracy. We attempt to increase the predictive accuracy both for malignant and benign samples by applying feature selection methods (i.e. principal component analysis (PCA) and forward search) to machine learning algorithms (i.e. logistic regression, SVM, SVR and boosting). The paper is organized as the following. Section II describes our data set. Section III describes our process in developing a gene classifier. Section IV reviews our results and comments on future work.

II. DATA SET

Data was collected by study conducted by Alexander. The study collected genetic data from fine needle thyroid nodule aspirates from thyroid nodules >1cm across the United States over a 19-month period. More detailed information is stated in Alexander's paper. They collected 2812 samples, of which 577 were indeterminate cytologically. Of the 577, 413 were viable for resection, so the patients underwent thyroid surgery to determine if their thyroid nodules were malignant or benign. 413 resection samples were classified by histopathological review. Of the 413, 265 was considered to be a valid data set. In addition, the paper also evaluated a randomly selected subset of 47 cytologically benign and 55 cytologically malignant surgical samples from an independent group of patients. For our gene expression classifier, we used the 47 and 55 randomly selected independent subsets as our training data, so we had a training data size of 102.

While we were training models, we found that there were 3 mislabeled samples in our training data. These 3 samples were cytologically labeled as malignant, but histopathologically labeled as the opposite. We choose to remove the data rather than correct them because we wanted our model to be cytologically accurate. Therefore, we ultimately used a training set with 99 samples.

Then we used 265 samples classified as cytologically indeterminate as our testing data. The true diagnosis of the cytologically indeterminate samples was determined histopathologically. We split the data in this fashion because we hypothesize that by training the model with data from cytologically confirmed results, a more accurate model for classifying the cytologically indeterminate aspirates can be built. Future work can be done to address samples that have conflicting cytological and histopathological results. Data has been preprocessed by robust multiarray average (RMA) method, so our data set is just an estimate of the expression measure for each gene using all the replicated probes for that gene. The data has 173 features overall.

III. Learning Algorithm Setup

To create our gene expression classifier, we first applied naïve learning models to gain intuition about what is the best way to handle our data.

A. *Naïve Baseline Models*

We first applied PCA to conduct our initial phase of pre-processing. PCA finds orthonormal basis for data and sorts dimensions in order of "importance", thus getting more compact description of features and discarding lower significant, redundant dimensions of features. After applying PCA on the training set, we selected the most important 71 components (which accounts for around 99% of the variance) out of the original 173 dimensions of feature. We created some naïve baseline models. The naïve baseline models helped us understand the structure of our data to build a strategy for classifying. The simplest models we used are logistic regression and support vector machine (linear kernel). In developing our intuition, we tried some slightly more complicated models: support vector regression, adaboost, and neural networks. For methods which have model parameters, we used 10-fold cross validation to choose parameters. We implemented the machine learning algorithms using MATLAB. For SVM, we used LIBSVM[3]. The initial results are displayed in Table 1.

Algorithm	CV Accuracy	Test Accuracy
Logistic Regression	/	80.00%
Support Vector Machine	97.06%	80.38%
Support Vector Regression	96.08%	81.13%
AdaBoost	95.10%	73.21%
Neural Networks	97.98%	79.25%

Table 1 Cross validation accuracy and test accuracy results for different learning models

As shown in the table, only support vector regression improved the testing accuracy to 81.13%. Our test accuracy seemed to be limited to around 80%. Also, note that there's a large gap between cross validation error and test error, which indicates over-fitting. So for the next step we experimented with feature selection methods. We examined the learning curve with respects to varying feature set size to understand the effect of each feature selection method. We used a linear support vector machine in the following section to test our data set.

B. *Learning Curve*

1) *Principal Component Analysis*

We varied the size of dimensions compressed by principal component analysis (PCA) to see how the training error and test error changes, demonstrated in Figure 1 below.

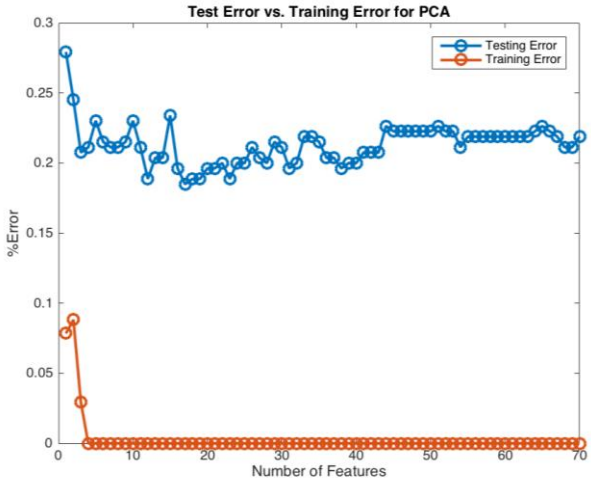

Figure 1 Test Error vs. Training Error for PCA

As training error quickly drops to 0, testing error increases slightly as the feature set grows: a sign of over-fitting. It is possible that the preliminary feature selection process is choosing extraneous features that lead to over-fitting. As we known, PCA can efficiently discard redundant dimensions of features, but it cannot find extraneous features because PCA does not use the knowledge of data labels. Next, we experimented with forward search.

2) *Forward Search*

Instead of using the conventional greedy algorithm of forward search, we used another version: Restricted Forward Selection (RFS) algorithm[4]. The RFS algorithm is similar to the greedy algorithm except that at each step to insert an additional feature into the subset, conventional greedy algorithm considers all the remaining features, while RFS only considers part of them (the set of features it considers decreases in each step). RFS is shown to be much more efficient and gets performance very close to the conventional greedy algorithm[4].

In each step of forward search, we use leave-one-out cross-validation (LOOCV) error to evaluate feature set and choose the best feature to be added in. Figure 2 shows the learning curve using forward search, again we used a linear support vector machine.

From the figure, we know that the error of training and testing drops quickly for the first 3 steps. But after that the

training error drops to 0, the test error increases slightly as the feature set grows. This demonstrates that the forward search method does not reduce over-fitting either.

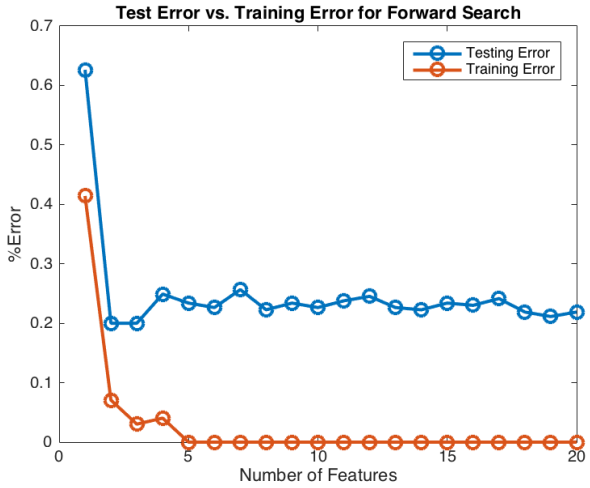

Figure 2 Test Error vs. Training Error for Forward Search

Of the two different feature selection techniques we tested, we found that forward search and PCA helped us discard some extraneous features. However, the test error is still around 80%, and there was a big gap between testing error and training error. Intuitively, both training error and testing error should be big but the gap between them should be small when we have low model complexity (i.e. small number of features). As the model complexity grows, both training error and testing error should drop rapidly. When the number of features increases above a certain threshold, the testing error starts rising because additional features generate no useful information at all and can cause overfitting. In our learning curves, the constant big gap between training error and testing error suggested that overfitting still exists in our model.

C. Regularization

It is possible that the high model complexity leads to the over-fitting. To further reduce the complexity of our models, we applied regularization to our model because it is known to help prevent overfitting. In our case, we chose to use L2 regularization. We used PCA to select features and the feature set size is 18. Then we applied linear SVM and logistic regression both with L2 regularization and without regularization to our data set. The result is shown in the following table. The test accuracy slightly increases, but not by much. The learning curve with and without regularization (using SVM) is shown in Figure 3.

Algorithm	Test Accuracy Without Regularization	Test Accuracy With L2 Regularization
Logistic Regression	80.38%	81.51%
SVM	80.75%	81.89%

Table 2 Test accuracy result of L2 regularized models

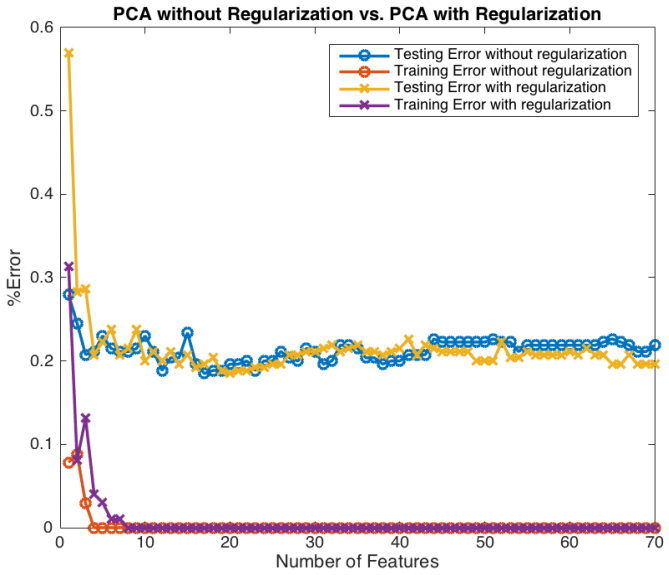

Figure 3 Test Error vs. Training Error for PCA with and without L2 regularization

D. Reweighting

After taking a closer look at the testing results, we found that while the testing errors of benign sample were relatively small, our model misclassified a significant percentage of malignant samples. We would like the testing error of malignant samples to be as small as possible since the ability to differentiate malignant samples from others is the main goal of the classifier. Therefore we tried to adjust the decision boundary of our model by adding a penalty to any misclassification of malignant samples during training. The penalty is kept below a threshold because overall test accuracy would drop significantly if we imposed a large penalty. Table 3 contains the testing results of our model after a penalty is added.

Algorithm (benign penalty: malignant penalty)	Total test accuracy	Malignant samples testing error	Benign samples testing error
L2 logistic regression (1:1)	81.51%	36.47%	10.00s%
L2 logistic regression (1:5)	80.38%	28.24%	15.56%
L2 logistic regression (1:10)	76.98%	25.88%	21.67%
L2 regularized linear SVM (1:1)	81.89%	37.65%	8.89%
L2 regularized linear SVM (1:5)	80.00%	30.59%	15.00%
L2 regularized linear SVM (1:10)	77.36%	24.7%	21.67%

Table 3 Test accuracy of L2 regularized models with penalty

IV. Results

A. Summary of Reults

We tried several methods to reduce over-fitting, but they didn't work very well. Feature selection methods such as PCA and forward search contribute little to the improvement of over-fitting. Regularization did help a little bit in reducing over-fitting but it was still far from ideal. We think this may be because of the small sample size used for training data.

B. Future Work

For future work, we could try other methods to develop a learning model for gene classification such as Bayesian networks. Bayesian networks have a unique advantage by incorporating prior distributions into the classification process. Therefore giving us a method to statistically capture the gene to gene interaction that is not accounted for in our current models.

V. Acknowledgment

We would like to acknowledge Olivier Geveart for providing us with the paper and data set to explore this topic. We would like to thank and acknowledge Irene Kaplow for help and guidance throughout this project.

VI. References

[1] Alexander, E. K. *et al*. Preoperative diagnosis of benign thyroid nodules with indeterminate cytology. *N. Engl. J. Med*. **367**, 705–715 (2012).

[2] "Thyroid Neoplasm." *Wikipedia*. Wikimedia Foundation, 15 Apr. 2015. Web. 30 Nov. 2015.

[3] Chang, Chih-Chung, and Chih-Jen Lin. "Libsvm." *TIST ACM Transactions on Intelligent Systems and Technology ACM Trans. Intell. Syst. Technol.* 2.3 (2011): 1-27. Web.

[4] Deng, Kan. *OMEGA: On-line Memory-based General Purpose System Classifier*. Pittsburgh, Pa.: Carnegie Mellon U, The Robotics Institute, 1998. Print.