Classification Methods for Different Types of Leukemia

Wei-Yu Tseng, Department of Statistics and Data Science

weiyut@stat.cmu.edu

Abstract

This paper will develop methods to classify two types of blood cancer, acute myeloid leukemia (AML) and acute lymphocytic leukemia (ALL), based on gene expressions. We performed the data analysis using Python programming language. Since the dataset contains more than 3000 different genes, we used dimension reduction technique, Principal Component Analysis, to reduce the computation cost for the analysis while keeping majority part of the information. The methods we applied here are Linear Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) to help us classify the types of leukemia. Both methods have decent precision, recall, and F-measure, but Linear SVM has slightly better performance than the KNN. Although the experiment is successful, we only have 72 observations and 36 both for training and testing dataset, we would like to know if the algorithm is still feasible when the data size is large.

1 Introduction

Although both acute myeloid leukemia (AML) and acute lymphocytic leukemia (ALL) are both cancers of the blood and bone marrow, the treatments are slightly different. Since both types share similar symptoms, doctors need to perform several tests and physical exams before reaching a diagnosis. However, since both cases are serious conditions that develop rapidly, and that the diagnosis processes are time and money consuming, we would like to know an easier way to not only help us distinguish the cancer, but also predict the cancer. In this paper, we will develop classification strategies based on gene expression of our samples to help us decide the types of leukemia.

2 Data

The Golub dataset we used in this analysis came was collected by the nonprofit, independent research institution, Whitehead Institute, who is affiliated with Massachusetts Institute of Technology.

This dataset contains 72 patients with two different types of leukemia and 3053 different genes of the patient.

Table 1 shows the number of observations in each type of leukemia.

Tumor Type	Acute myeloid leukemia (AML)	Acute lymphocytic leukemia (ALL)
Observations	25	47

Table 1: Number of observations in each type of leukemia for the entire dataset

We split our data into 50% as training dataset and 50% as testing dataset. Two datasets share similar structure in terms of the proportion of two leukemia types.

Table 2 shows the number of observations in each type of leukemia for training dataset, Table 3 shows the number of observations in each type of leukemia for testing dataset.

.

Tumor Type	Acute myeloid leukemia (AML)	Acute lymphocytic leukemia (ALL)
Observations	13	23

Table 2: Number of observations in each type of leukemia for training dataset

Tumor Type	Acute myeloid leukemia (AML)	Acute lymphocytic leukemia (ALL)
Observations	12	24

Table 3: Number of observations in each type of leukemia for testing dataset

We use the training dataset to help us construct models and testing dataset to verify the validation of them.

3 Methods

We first apply Principal Component Analysis to simplify our calculations, and the run two different classification algorithms, linear Support Vector Machine and K-Nearest Neighbor, to help us predict and classify the types of leukemia.

Principal Component Analysis

The purpose of Principal Component Analysis (PCA) is to obtain a lower dimensional representation of data. If the columns of the dataset are features, we can find a small number of variables that capture most of the information contained in a large number of features, and we can display the data in lower dimensions for visualization or clustering of the points, as well as reducing the calculation cost.

For our Golub dataset, we have 3053 features. However, since every feature is likely not equally important, and that model may not be stable when training with this many variables, we need PCA to help us reduce the dimension of our data.

After performing PCA on our training data, we would like to choose how many principal components (PCs) we would like to use. One way to help us make the decision is to look at the elbow point of the scree plot.

The scree plot (see Figure 1) shows the proportion of variance explained by each PC, and we can see that after starting at 6th PC, the percentage of variance accounted for each are similarly small compared to the first 5 PCs. In addition, top 5 PCs also explain more than 50% variance of the entire training dataset (see Figure 2). Therefore, we select top 5 PCs to conduct our analysis.

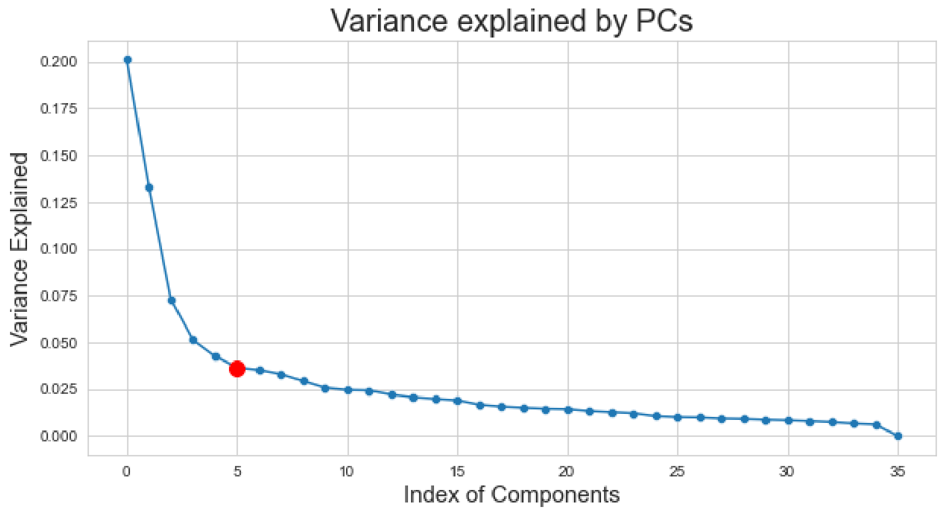

Figure 1: Scree plot (Variance explained by each PC)

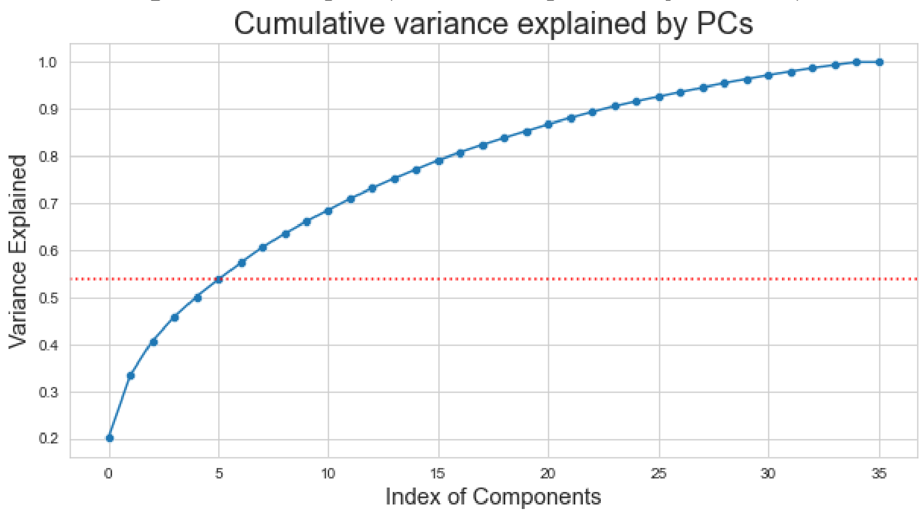

Figure 2: Cumulative variance explained by PCs

Method 1 - Linear Support Vector Machine

For our training data, the linear Support Vector Machine (SVM) marked each observation as belong to one of the two categories – ALL or AML and maximize the width of the gap between the two categories. The linear SVM then predicted new examples (testing data) to belong to a category based on which side of the gap they fall. In our case, since we used 5 PCs, our linear SVM created a 5-dimension hyperplane that separated two types of leukemia.

Method 2 – K-Nearest Neighbor

K-Nearest Neighbor algorithm first stored our training data into the model, and when we wanted to determine which type of leukemia our input data (testing data here) are, the algorithm looked at the states of the K training data points near it, in terms of Euclidean distance. The group that most K data points surrounding are in would be the assigned as the label of the data we tested. In our case, since we only have two groups and the size of our training data is relatively small, with only 36 observations, we believe setting K to 3 is adequate for our analysis.

4 Results

In this section, we used 4 metrics,

1. Accuracy: a ratio of correctly predicted observation to the total observations.
2. Precision: the ratio of correctly predicted positive observations to the total predicted positive observations.
3. Recall: the ratio of correctly predicted positive observations to all observations in actual class.
4. F-score: weighted average of Precision and Recall.

to evaluate the performance of our model. If all four measurements of the model give good results, we say the model perform well.

Method 1 - Linear Support Vector Machine

a. **Training Dataset**
Confusion Matrix

n = 36	True: ALL	True: AML
Predicted: ALL	22	1
Predicted: AML	1	12

Classification Report

Type	Precision	Recall	F-score
ALL (n = 23)	0.96	0.96	0.96
AML (n = 13)	0.92	0.92	0.92

The overall training accuracy for our linear SVM model is 94.44%. All measurements for both types of leukemia are greater than 0.9, suggesting that the model fit the training data well. However, since the model already saw these data, we would need testing data to help justify the performance of the model.

b. **Testing Dataset**
Confusion Matrix

n = 36	True: ALL	True: AML
Predicted: ALL	23	1
Predicted: AML	1	11

Classification Report

Type	Precision	Recall	F-score
ALL (n = 24)	0.96	0.96	0.96
AML (n = 12)	0.92	0.92	0.92

The overall testing accuracy for our linear SVM model is 94.44%. Similar to training data, all measurements for both types of leukemia are greater than 0.9, implying that the model classified unseen data well. This also suggests that the model is decent and valid.

Method 2 – K-Nearest Neighbor

a. **Training Dataset**
Confusion Matrix

n = 36	True: ALL	True: AML
Predicted: ALL	23	2
Predicted: AML	0	11

Classification Report

Type	Precision	Recall	F-score
ALL (n = 25)	1	0.92	0.96
AML (n = 11)	0.85	1	0.92

The overall training accuracy for our KNN model is 94.44%. All measurements for both types of leukemia are greater than 0.85, suggesting that the model fit the training data well. However, since the model already saw these data, we would need testing data to help justify the performance of the model.

b. Testing Dataset

Confusion Matrix

n = 36	True: ALL	True: AML
Predicted: ALL	22	2
Predicted: AML	2	10

Classification Report

Type	Precision	Recall	F-score
ALL (n = 25)	0.92	0.92	0.92
AML (n = 11)	0.83	0.83	0.83

The overall testing accuracy for our KNN model is 88.89%. The model seems to have hard time recognizing AML types of leukemia, with only 0.83 Precision, 0.83 Recall, and 0.83 F-score compared to ALL's 0.92's, though in general the model still appears to be powerful.

5 Discussion

Results went well even though we only selected top 5 principal components (Reduce our dimension from 3052 to merely 5) to perform our analysis. This is probably because our dataset is very informative. For further analysis, if we expand the classification categories (for example, include other leukemia types such as chronic myeloid leukemia (CML) and chronic lymphocytic leukemia (CLL)), the dataset may not be as informative, and we may need more principal components or even other more powerful dimension reduction methods (such as sparse PCA) before we conduct the analysis.

For two algorithms we applied on our dataset, linear SVM seems to be a better fit for the data. The predicting results we obtained from training and testing dataset are similar and also better than those of our KNN model. However, since we only selected 3 neighbors for our KNN model, the result may improve if we include the numbers of neighbors but may also increase the risk of overfitting.

In conclusion, both linear Support Vector Machine and K-Nearest Neighbor seem to be great algorithms to help classify and predict the types of leukemia, but linear SVM yield a more stable and precise analysis than KNN in our cases. However, since we only have 72 observations in our dataset and only 36 each for training and testing dataset, more data may be required to prove the effectiveness of our models and the selection of algorithms.

References

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, E. S. Lander1 (1999), *Molecular classification of cancer: class discovery and class prediction by gene monitoring*.

Technical Appendix

Python source code for this report:

https://github.com/weiyut/Genetics/blob/main/Genetics%20Final%20Project%20-%20Leukemia%20Classification.ipynb