Gene Expression Classification

Dmitriy Vasylyshyn   
Data Science Project

*Abstract*—The classification of different tumor types is very important in today’s cancer diagnosis and drug discoveries. Cancer classification using gene expression data is known to contain the keys of addressing the fundamental problems relating to cancer diagnosis. The discovery of DNA microarray technique has made simultaneous monitoring of gene expressions possible and gives scientists the ability to analyze the data and make accurate predictions of different type of tumors. This helps identify the necessary treatment for the patient. In this paper I will test three different classification algorithms, based on the end result, we should be able to identify models that best fit the current dataset.

# Introduction

“Microarray is a recent technology, helps to identify the patterns of gene expressions of multiple genes at a time in genomic level. It supports the researchers to analyze and investigate hundreds and thousands of genes in single experiment.”[1] With the help of this breakthrough technology scientists can discover and analyze gene patterns of many rare diseases and the research has been globally improving every day. There are many different classification techniques that can be applied on such datasets, for example based on the information from an article published by Heping Zhang, Chang-Yung Yu and Burton Singer “Cell and tumor classification using gene expression data: Construction of forests”. In this article they used a technique called Deterministic Forest. Forest-based classification and prediction is one of the most commonly used nonparametric statistical methods in many scientific and engineering areas, particularly in machine learning and analysis of high-throughput genomic data.[2] I have also found another very interesting article published on plos.org by Dominik Schaack, Markus Weigand and Florian Uhle[3]. The purpose of their analysis was to find the best fitting machine-learning technique to perform a diagnosis of a sepsis gene expression dataset. The techniques that they implemented were: Decision Trees, Random Forest, Support Vector Machines, and Deep-Learning Neural Network. Based on their results the average accuracy score for all the algorithms was 0.955. One specific part of their project which I found interesting is where they randomly reduced their dataset by 2,361 genes and used a residual of 3,571 genes, with that being said the models still performed at a similar performance level.

The dataset that is being used in this analysis comes from the University of California, Irvine. It is a gene expression dataset that consists of 5 different classes, it has 801 instances and 20.5k attributes. In this project I will prep the data and apply three different classification techniques. The goal is to identify the model that will provide the best classification results for the dataset at hand. I will be using three different classification techniques that in my opinion will provide optimal results. In this paper I will be going through detailed data descriptions and data cleaning techniques that were applied. I will discuss the obstacles that came up throughout the analysis. The methodology section will include the algorithms that are being applied followed by results and discussion of the results.

# Data Description

The dataset has 2 main labels: class and gene. Class represents the type of tumor and gene represents the correspondence of the cell to the protein; the data examples are provided in Table I. As mentioned in the introduction the dataset contains 5 different classes, 801 instances and 20.5k of genes.

Table I. Data Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Example Value** | **Description** |
| Class | Nominal (string) | BRCA | Type of Tumor |
| Gene | Numeric(integer) | 9.233 | Sequence of gene |

# Data Cleaning

While examining the data I observed many “0” gene values, which in the classification analysis can really skew the model’s decision making. I have decided to drop all the columns that contained any zero values. My dataset was now reduced to 9671 attributes the number of instances remained the same. At this point the dataset consists of 5 classes: BRCA, COAD, PRAD, LUAD, KIRC.

After running a test analysis, which in this case was Support Vector Classifier, the model produced a 100% accuracy score, which is not impossible but too good to be true. At this stage of the analysis, I believe that the issue may be that the data is imbalanced. If we look at Table II we can see that class BRCA is dominant. To challenge our model, I decided to remove the instances of dominant class, which would put it in the average category with the other four classes.

Table II. Class Value Count

|  |  |
| --- | --- |
| Class | Count |
| BRCA | 300 |
| KIRC | 146 |
| LUAD | 141 |
| PRAD | 136 |
| COAD | 78 |

Figure II displays a pie chart after the dataset has been balanced. Here we can see more reasonable percentages. At this point the dataset is ready to be classified.

Figure II Class Percentage (Balanced)

Chart, pie chart

Description automatically generated

# Methodolody

All the analysis were performed in Google Collaboratory. It is an open-source software that is easy to use and has most of the necessary packages preinstalled.

## Support Vector Classifier (SVC/SVM)

Support Vector Machines is a linear model for classification and regression problems. It can be used in linear and non-linear problems. The idea of SVM is straight forward, the algorithm creates a line or a hyperplane which separates the data into classes. At the first attempt, SVM finds a separating hyperplane between the data of two classes, then it takes the data as an input and outputs a line that separates those classes further, if possible. SVM then finds the points of each class that are closest to the hyperplane, these points are called support vectors. Now, it computes the distance between the line and the support vector, this distance is called a margin. The goal of this algorithm is to maximize the margin. The larger the margin distance the more optimal is the hyperplane.

Diagram, schematic

Description automatically generatedFigure III. SVM [3]

La

## K-Neighbors Classifier(KNN)

K-Nearest Neighbors is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test date by calculating the distance between the test date and all the training points. The algorithm selects the number of points which is closest to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of ‘K’. The training data and class that holds the highest probability will be selected.

## Random Forest

Random Forest is a supervised machine learning algorithm, put into simple words it builds multiple decision trees and merges them together for a more accurate and stable prediction. A big advantage of random forest is that it can be used for both classification and regression problems. Random Forest has nearly the same hyperparameters as a decision tree, it adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

# Results

Across the board all the analysis performed well, some had better scores than others but in general each one of these analyses are acceptable to be used on similar datasets.

## Support Vector Classifier (SVC/SVM)

Table III displays the results of SVC analysis, based on these scores the SVC has done an excellent job with the dataset at hand. In this case the biggest mistake that SVC made was in COAD where it misclassified one instance out of 37. The accuracy score for SVC came out to 0.9969. Table IV lays out a confusion matrix.

# Table III. SVC Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1 | Support |
| BRCA | 1.00 | 1.00 | 1.00 | 84 |
| COAD | 1.00 | 0.97 | 0.99 | 37 |
| KIRC | 1.00 | 1.00 | 1.00 | 66 |
| LUAD | 0.98 | 1.00 | 0.99 | 63 |
| PRAD | 1.00 | 1.00 | 1.00 | 76 |
| Accuracy |  |  | 1.00 | 326 |
| Macro avg | 1.00 | 0.99 | 1.00 | 326 |
| Weighted avg | 1.00 | 1.00 | 1.00 | 326 |

Table IV. SVC Confusion Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 78 | 0 | 0 | 0 | 0 |
| 0 | 43 | 0 | 0 | 0 |
| 0 | 0 | 75 | 0 | 0 |
| 1 | 0 | 0 | 64 | 0 |
| 0 | 0 | 0 | 0 | 65 |

*B. K-Neighbors Classifier (KNN)*

The accuracy score for KNN came out to 0.9877 and table V displays the result per class. Table VI shows the confusion matrix.

Table V. KNN Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1 | Support |
| BRCA | 0.97 | 0.97 | 0.97 | 68 |
| COAD | 1.00 | 1.00 | 1..00 | 39 |
| KIRC | 1.00 | 1.00 | 1.00 | 77 |
| LUAD | 0.97 | 0.99 | 0.98 | 75 |
| PRAD | 1.00 | 0.99 | 0.99 | 67 |
| Accuracy |  |  | 0.99 | 326 |
| Macro avg | 0.99 | 0.99 | 0.99 | 326 |
| Weighted avg | 0.99 | 0.99 | 0.99 | 326 |

Table VI. KNN Confusion Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 66 | 0 | 0 | 2 | 0 |
| 0 | 39 | 0 | 0 | 0 |
| 0 | 0 | 77 | 0 | 0 |
| 1 | 0 | 0 | 74 | 0 |
| 1 | 0 | 0 | 0 | 66 |

*C. Random Forest*

Table VII displays the results for Random Forest analysis with the accuracy score of 0.9846. Table VIII sets forth the confusion matrix.

Table VII. Random Forest Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1 | Support |
| BRCA | 0.99 | 0.97 | 0.98 | 74 |
| COAD | 1.00 | 0.95 | 097 | 37 |
| KIRC | 0.99 | 1.00 | 0.99 | 70 |
| LUAD | 0.96 | 0.99 | 0.97 | 76 |
| PRAD | 1.00 | 1.00 | 1300 | 69 |
| Accuracy |  |  | 0.98 | 326 |
| Macro avg | 0.99 | 0.98 | 0.98 | 326 |
| Weighted avg | 0.98 | 0.98 | 0.98 | 326 |

Table VIII. RF Confusion Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 72 | 0 | 1 | 1 | 0 |
| 0 | 35 | 0 | 2 | 0 |
| 0 | 0 | 70 | 0 | 0 |
| 1 | 0 | 0 | 75 | 0 |
| 0 | 0 | 0 | 0 | 69 |

To get a better visibility on the accuracy score results I have put together a table of accuracy scores for each model as shown in table IX.

Table IX. Models Accuracy Scores

|  |  |
| --- | --- |
| Model | Accuracy Score |
| SVC | 0.9969 |
| KNN | 0.9877 |
| RF | 0.9846 |

VI. Discussions

Based on the accuracy scores in the Result section all of algorithms performed well. One thing that I found interesting was the Radom Forest results. Based on my research Random Forest usually outperforms most of the classification algorithms. In this case Random Forest received an accuracy score of 0.9846 while SVC got 0.9969. I understand that the difference was very small, but it was still a bit surprising to me that SVC score was higher. Another thing that I have found to be very interesting is that all the classifiers had one thing in common. Each of the classifiers have misclassified one instance of class LUAD. I did look over the data with more detail. I have compared the overall averages of genes for each class but there seemed to be no pattern. Therefore, I could not identify anything out of ordinary that might cause the models to misclassify this specific class.

1. S. Karthik, M. Sudha “A Survey on Machine Learning Approaches in Gene Expression Classification in Modelling Computational Diagnostic System for Complex Diseases” International Journal of Engineering and Advanced Technology (IJEAT), December, 2018. ( Link: https://www.researchgate.net/profile/Karthik-Sekaran/publication/331638460\_A\_survey\_on\_machine\_learning\_approaches\_in\_gene\_expression\_classification\_in\_modelling\_computational\_diagnostic\_system\_for\_complex\_diseases/links/5cb212b2299bf1209764015e/A-survey-on-machine-learning-approaches-in-gene-expression-classification-in-modelling-computational-diagnostic-system-for-complex-diseases.pdf)
2. Heping Zhang, Chang-Yug-Yu, Burton Singer “Cell and tumor classification using gene expression data: Construction of forests” PNAS. March 17, 2003. (Link: <https://www.pnas.org/doi/10.1073/pnas.0230559100#sec-2>)
3. Ihttps://www.javatpoint.com/machine-learning-support-vector-machine-algorithmK. Elissa, “Title of paper if known,” unpublished.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**