南方科技大学 Southern University of Science and Technology | 统计与数据科学系 Department of Statistics and Data Science

Classification of Normal and Tumor Tissues in Colon Data Set

Author:

Deng Chunli 11711432

Feng Zhen 11711135

Liu Runqi 11711331

Course: MA439 Statistical Deep Learning

Professor: CHEN, Xin

Date: Nov 20, 2020

Abstract

Cancer is a disease that endangers a patient's life but is difficult to detect early when the symptoms are not obvious, so the key to reduce the rising death rate from cancer lies in how to detect cancer effectively and accurately as early as possible. With the development of Biogenetics, as well as more studies of microarray gene expression, classification of tissue samples can be a valuable diagnostic tool for diseases such as cancer. This report aims to classify 62 tissues as normal or tumor based on a high-dimension sparse data, Colon data set. In the given data set, the number of genetic features far outnumbers that of observations, and the sample size is limited for statistical analysis. Therefore, it is vital to select effective genetic variables and screen out effective classification methods to reduce the error rate. Several classification methods such as penalized logistic regression with LASSO, Classification and Regression Trees, Random forest and k-Nearest-Neighbor Techniques have been discussed in this report.

Keywords: high-dimension classification, dimension reduction, LASSO, CART, Random forest, KNN.

1. Introduction to the Colon data set

Alon et al. [1] used high-density oligonucleotide arrays to study gene expression patterns in tumor and normal colon tissues. The expression levels of p = 6500 genes were measures in n = 62 samples, 40 tumor and 22 normal colon tissue samples. In the publicly available data set, the 2000 genes with the highest minimal intensity across samples were kept for further study. There are three material we will use:

The matrix I2000 : the expression of the 2000 genes with highest minimal intensity across the 62 tissues, where genes are placed in order of descending minimal intensity.

The file ‘names’ : including the EST number and description of each of the 2000 genes, in an order that corresponds to the order in I2000.

The file tissues: the numbers correspond to patients, a positive sign to a normal tissue, and a negative sign to a tumor tissue.

Separation of data sets

(1) **Separation 1**: randomly assign two parts of the data to training set with first 40 observations and testing set with the later 22 observations.

(2) **Separation 2**: divide more randomly with randomly generated index to obtain the training set: accounting for 70% of the total sample size (43), and the left are in the test set (19).

All models were built based on training data and assessed by using the testing data.

2. Classical Discriminant Analysis

2.1 Logistic regression

The final aim of this project is to classify 62 tissues as normal or tumor given that they are labeled according to the values in "tissues", and the response only takes two values or in two categories. Based on that, we denote responses as Y = 0 and Y = 1, that means for each X, its response is random and follows the binomial distribution with probability

$$P(Y=1|X) = p(X,\tilde{\beta}) \quad \text{and} \quad P(Y=0|X) = 1 - p(X,\tilde{\beta})$$

and

$$Y \sim B(p(X,\tilde{\beta}))$$

Classification

For a new sample X, the estimated model can predict its p

$$\hat{p} = \frac{\exp(\hat{a} + \hat{\beta}^\top X)}{1 + \exp(\hat{a} + \hat{\beta}^\top X)}$$

The classification is made as follows

- if $\hat{p} \geq 0.5$, classify $Y = 1$
- if $\hat{p} < 0.5$, classify $Y = 0$

It is natural to consider building a logistic regression model to select significant variables and predict the response with the fitted model. However, in this case with massive biological genes (high dimension variables) and few patients(sample size), logistic regression does not work, because the sample size (62) is inadequate to determine p free parameters (2000). In other words, the sample covariance matrix is singular when the dimensionality is larger than the sample size. As a result, the fusion of dimension reduction methods needs to be considered.

2.2 Penalized Model with LASSO

LASSO (least absolute shrinkage and selection operator)

$$\min_{\beta} \Big\{ \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 \mathbf{x}_{i1} - ... - \beta_p \mathbf{x}_{ip})^2 + \lambda ||\beta||_1$$

and

$$\text{LASSO}: \quad \sum_{i=1}^{n} \{Y_i - \beta_1 \mathbf{x}_{i1} - \beta_2 \mathbf{x}_{i1}\}^2, \quad s.t. \quad \sum_{j=1}^{2} |\beta_j| \leq t$$

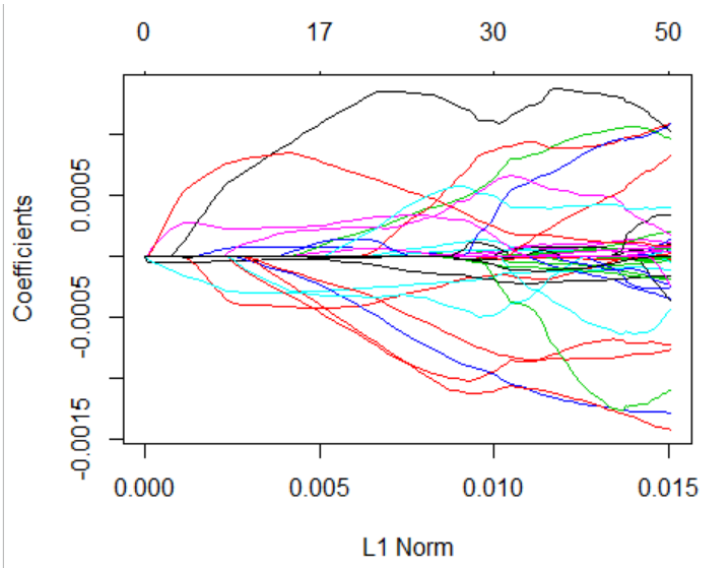
Because there are zeros values in the estimated β, the actual assumption behind the method is the "sparsity". This will help us for variable (model) selection.

LASSO is a helpful tool to discard variables that are irrelevant, the result shows that only a few variables(genes) are selected in the prediction, i.e. only those genes might

be the cause of the disease.

LASSO for the logistic regression

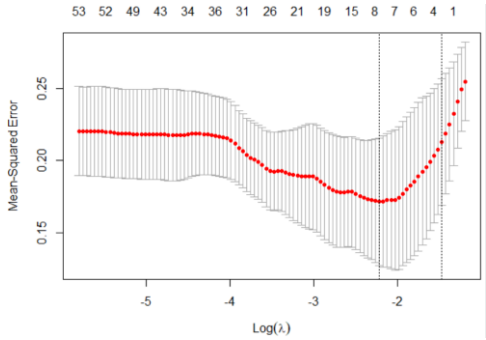
The coefficient paths for each class's variables are as follow:



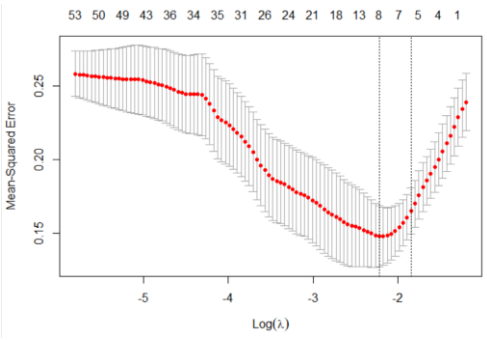
2.3 A comparison of different k-fold

Here, we use different values of fold to have an exploration of the best lambda. Considering the randomness of cross-validation, such problem can be settled by setting random seed. By testing, we find the seed (123) is ideal so other outputs of experiments are not listed here. The values of λ and its CV-values (number of folds=3,5,10):

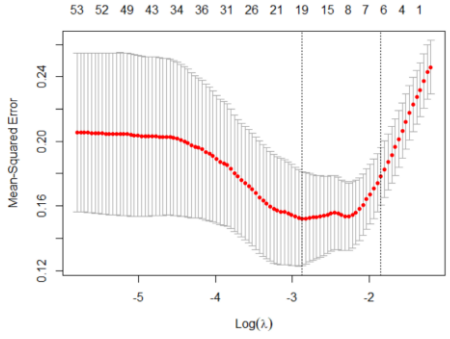
k=3



k=5



k=10



The best value of λ for them are: 0.1087579, 0.1087579, 0. 0.05670646. As the number of folds is increasing, the value of λ decreases. Because of the nature of the constraint, letting t sufficiently small or λ sufficiently large will cause some of the coefficients to be exactly zero. With the comparison of 3/5/10-fold, it reflects how the procedure of selecting the tuning parameter influences the variable selection.

	gene_index	coef
[1,]	1870	0.0042562983
[2,]	1772	0.0030585503
[3,]	1771	0.0015085261
[4,]	83	0.0002566751
[5,]	249	-0.0001281386
[6,]	377	-0.0012792053
[7,]	493	-0.0014458868

	gene_index	coef
[1,]	1870	0.0042562985
[2,]	1772	0.0030585505
[3,]	1771	0.0015085261
[4,]	83	0.0002566751
[5,]	249	-0.0001281386
[6,]	377	-0.0012792054
[7,]	493	-0.0014458869

	gene_index	coef
[1,]	1772	7.473256e-03
[2,]	1870	5.740775e-03
[3,]	1771	2.490805e-03
[4,]	1641	1.690854e-03
[5,]	1916	1.000838e-03
[6,]	974	3.693361e-04
[7,]	83	3.436727e-04
[8,]	698	1.303871e-04
[9,]	249	-7.292528e-05
[10,]	897	-1.516855e-04
[11,]	1058	-1.710876e-04
[12,]	554	-1.271120e-03
[13,]	1644	-1.904899e-03
[14,]	377	-1.948268e-03
[15,]	493	-1.960948e-03
[16,]	1482	-2.931024e-03
[17,]	523	-3.303752e-03

As it is shown that the choice of number of folds lead to different tolerance of variable variation, and their contribution to the final model. However, models fitted with different k-fold have several common explanatory variables, which are influential indications to the prediction with significant coefficients. (k=3/5/10)

```
> errorLasso
[1] 0.1098416
> classificationError
[1] 0.1052632
```

```
> errorLasso
[1] 0.1098416
> classificationError
[1] 0.1052632
```

```
> errorLasso
[1] 0.1252469
> classificationError
[1] 0.1578947
```

Since our training data set contains only about 40 samples, and the results (variable selection & classification error) above show that 3-fold is a reasonable choice.

2.4 Model assessment

The next model validation for the test data and model assessment are based on one random sample test with 5-fold. (best λ = 0.05358197) and the seed (123) case with 3-fold. (best λ = 0.1087579).

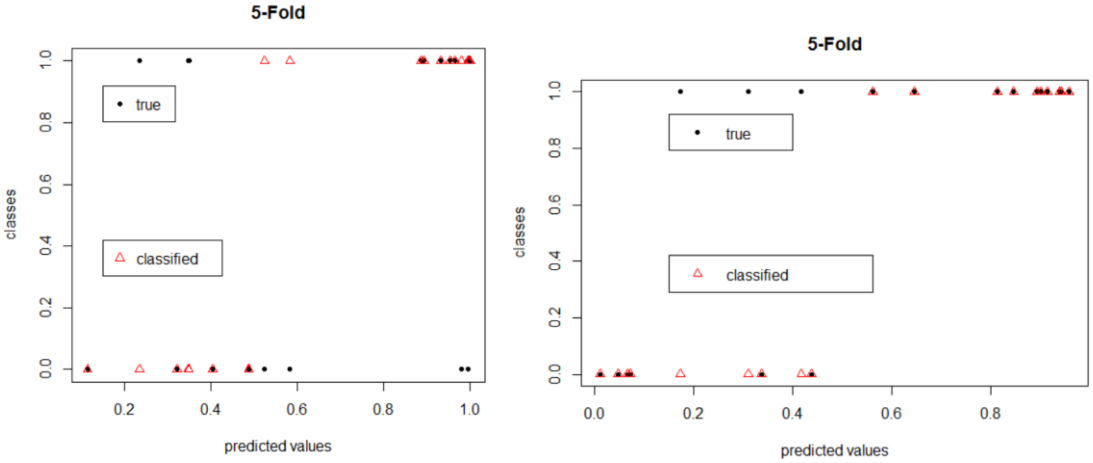
2.4.1 The first assessment is the results of two types of division of data sets.

(1)

Error of Lasso: 0.2184313, Classification Error: 0.3181818, Accuracy: 0.6818182

(2)

Error of Lasso: 0.1173961 Classification Error: 0.1578947, Accuracy: 0.8421053



Confusion matrix for the testing data is:

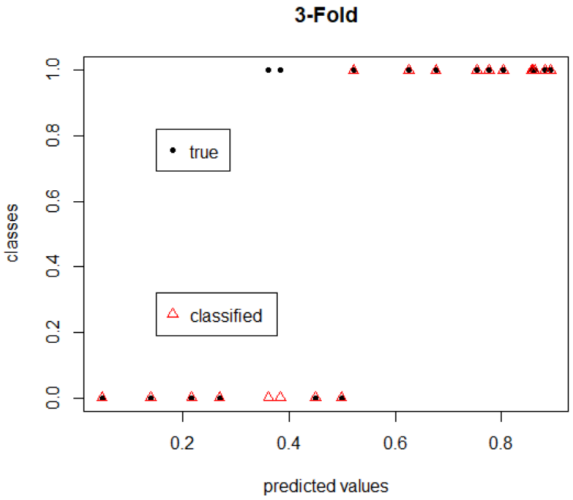
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  5  3
         1  4 10

               Accuracy : 0.6818
                 95% CI : (0.4513, 0.8614)
    No Information Rate : 0.5909
    P-Value [Acc > NIR] : 0.2608

                  Kappa : 0.3304

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.5556
            Specificity : 0.7692
         Pos Pred Value : 0.6250
         Neg Pred Value : 0.7143
             Prevalence : 0.4091
         Detection Rate : 0.2273
   Detection Prevalence : 0.3636
      Balanced Accuracy : 0.6624

       'Positive' Class : 0
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  6  3
         1  0 10

               Accuracy : 0.8421
                 95% CI : (0.6042, 0.9662)
    No Information Rate : 0.6842
    P-Value [Acc > NIR] : 0.1045

                  Kappa : 0.678

 Mcnemar's Test P-Value : 0.2482

            Sensitivity : 1.0000
            Specificity : 0.7692
         Pos Pred Value : 0.6667
         Neg Pred Value : 1.0000
             Prevalence : 0.3158
         Detection Rate : 0.3158
   Detection Prevalence : 0.4737
      Balanced Accuracy : 0.8846

       'Positive' Class : 0
```

Even though every test is conducted with the same number of folds, the output will be slightly different. But the total accuracy stays unchanged in one type of separation of data for random tests.

Here when we use Separation 1 and Separation 2 of data sets, the results indicate the second one is much better. The randomly generated training set ensures as much individual diversity as possible in both groups of cells (normal or cancerous) in a limited sample size. Therefore, the model fitted in this way will test the rest of the data more scientifically and accurately.

2.4.2 The second assessment is the results of the final model.




```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  6  2
         1  0 11

               Accuracy : 0.8947
                 95% CI : (0.6686, 0.987)
    No Information Rate : 0.6842
    P-Value [Acc > NIR] : 0.03414

                  Kappa : 0.7765

 Mcnemar's Test P-Value : 0.47950

            Sensitivity : 1.0000
            Specificity : 0.8462
         Pos Pred Value : 0.7500
         Neg Pred Value : 1.0000
             Prevalence : 0.3158
         Detection Rate : 0.3158
   Detection Prevalence : 0.4211
      Balanced Accuracy : 0.9231

       'Positive' Class : 0
```

After fitting the training data to the LASSO penalized model, we selected a few effective variables of the tumor class. In addition, this penalized model is more interpretive than other traditional models by keeping the most influential variables.
When testing the final model, it has a great classification efficiency: 0.8947368, and only 2 observations are misclassified.
As above experiments show, randomly selected and generated data sets are more appropriate, so we use **Separation 2** in the following models.

3. Classification and Regression Trees (CART) [2]

3.1 Decision Tree

Decision Tree is a common machine learning method. Taking dichotomy task as an example, we hope to learn a model from a given training data set to classify new examples. As the name implies, decision tree is based on tree structure to make decisions, which is just a natural mechanism for human beings to deal with decision problems. Our task of categorizing samples can be regarded as a response to the question "Are current samples normal?", therefore, Decision Tree can realize the exploration of data, describe the data outline, predict and classify, and know which variables are the most important. The regression tree partitions the space by binary recursive method: to decide on the splitting variables and split points (also called node).

3.2 The building of Decision Tree

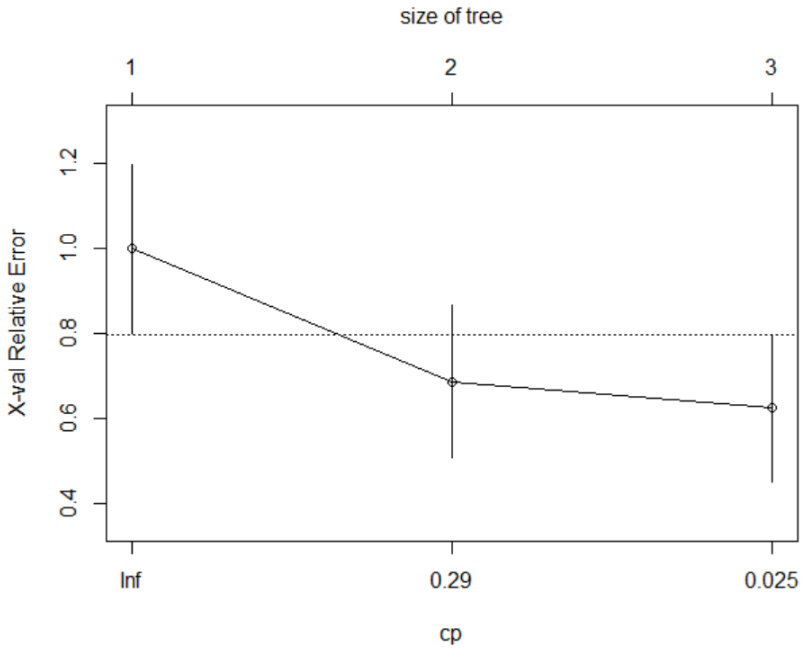
```
Classification tree:
rpart(formula = tissues ~ ., data = train, method = "class",
    parms = list(split = "gini"), control = tc)

Variables actually used in tree construction:
[1] X1419 X249

Root node error: 16/43 = 0.37209

n= 43

      CP nsplit rel error xerror    xstd
1 0.6875      0    1.0000 1.0000 0.19810
2 0.1250      1    0.3125 0.6875 0.17882
3 0.0050      2    0.1875 0.6250 0.17314
```

Cp is the penalty factor for complexity parameters that control the size of a tree. In short, the larger the CP, the smaller the nsplit. The output parameter (REL Error) indicates the average deviation ratio between the current classification model tree and the empty tree. Xerror is the cross-validation error, and XSTD is the standard deviation of the cross-validation error.

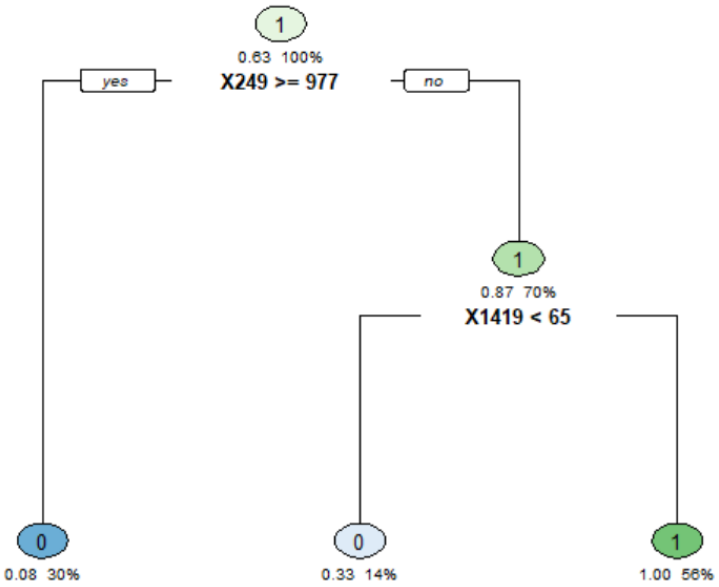
3.3 Check variables' importance

```
     X249      X245      X1423      X267      X765      X493
11.313536 10.443264  9.572992  9.572992  9.572992  8.702720
    X1419     X1165     X1346     X1772       X23       X31
4.266667 3.555556 3.555556 3.555556 2.844444 2.844444
```

As presented above, there are 16 most important variables listed, especially the first 6 ones.



Decision Tree



This diagram shows that the decision tree does not select variables based on the importance alone, because even with different genes they have some homogeneous expression. The model here uses the most important one X249, which has already determined the expression of most of the same genes and the second one X1419 that determines the expression of a few other types of expression genes.

3.4 Pruning Decision Tree

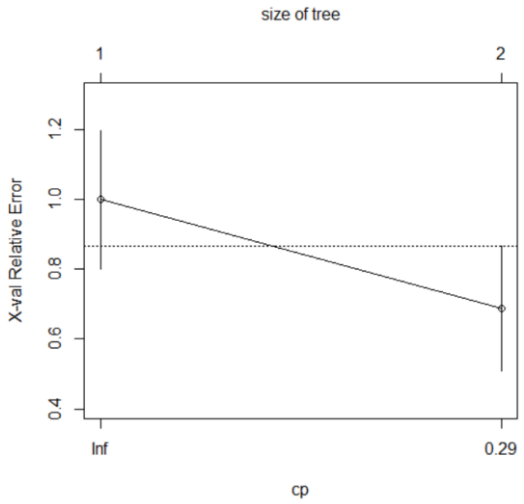
Pruning decision tree learning algorithm to deal with (pruning) is the principal means of "fitting", in the decision tree learning, as much as possible in order to correct classification the training sample, combined with the division process will be repeated, sometimes too much cause the decision tree branch, then might be due to the training sample is "too good" to learn, so much so that some characteristics of the training set itself as the general nature of the all the data has led to a fitting.
Therefore, the risk of overfitting can be reduced by actively removing some branches.
Check variables' importance:

```
      CP nsplit rel error xerror      xstd
1 0.6875      0    1.0000 1.0000 0.1981015
2 0.1250      1    0.3125 0.6875 0.1788204
```

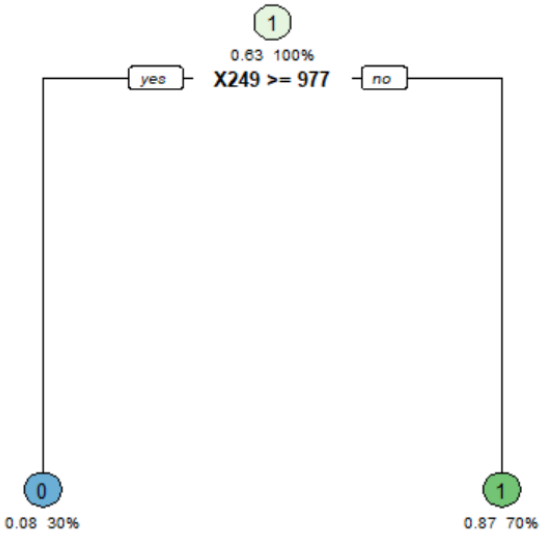
And the purpose of decision tree pruning is to get the tree with smaller xerror.

```
     X249      X245     X1423      X267      X765      X493
11.313536 10.443264  9.572992  9.572992  9.572992  8.702720
```

As can be seen, after the training, there are only 6 indicators are left that have significant importance, while many features unrelated to tumor are ignored.
The following plots present the X1419 is abandoned to simplify the model.



Decision Tree



The plot shows that after the pruning, the gene X1419 is abandoned to simplify the model, avoiding over-fitting. And the variable X249 is decisive for classification in this case.

3.5 Test & assess the model

Confusion matrix

actual	predicted 0	1
0	5	1
1	3	10

Error of Cart: 0.4701028, Classification Error: 0.2105263, Accuracy: 0.7894737

It shows that the classification efficiency of CART is lower than the LASSO penalized model: 0.7894737 < 0.8421053. It is worth noting that CART performances poor when we want to try to control the Type one error. And although it can produce the importance of different variables, the variables that play a decisive role in the

decision tree are not just the same as their importance, so it is hard to explain certain contribution of every variable. Because we only have two categories here and the response y is tagged with close labels, as well as limited sample size for machine learning, the prediction ability of the fitted model is not satisfactory.

4. Random forest

The Random Forest algorithm generates models by training multiple decision trees, and then makes comprehensive use of multiple decision trees for classification. It includes the randomness of Sample, Characteristics, Parameters and Model.
There are three main algorithms to construct decision tree:
A. ID3 selection criteria: characterized by information gain in the process of decision tree generation.

B. C4.5 Selection criteria: characterized by information gain ratio in the process of decision tree generation.

C. CART regression trees: the square error minimization criterion is used, and for classification trees, the Gini index minimization criterion is used for feature selection to generate binary trees.

Entropy: $$H = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$ Gini index: $$Gini = 1 - \sum_{i=1}^{n} p_i$$

The entropy and Gini value represent the complexity of the data. When the entropy or Gini value is too small, the purity of the data is relatively high. **If the entropy or Gini value is less than a certain number, the node will stop splitting**.

4.1 Building models

After generating T decision trees according to 1, for each new test sample, the classification results of multiple decision trees are integrated as the classification results of the random forest.
The target feature here is the category type: the minority is subordinate to the majority, and the category with the most classification results of a single tree is taken as the classification result of the whole random forest.

```
Call:
 randomForest(formula = tissues ~ ., data = train, importance = TRUE,
   na.action = na.roughfix)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 666

          Mean of squared residuals: 0.1607681
                    % Var explained: 31.19
```

Importance measured by mean decrease Gini:

	gene_index2	import			
[1,]	493	1.07629713	[26,]	897	0.05493701
[2,]	249	0.49344824	[27,]	682	0.04902640
[3,]	1582	0.46303861	[28,]	940	0.04840469
[4,]	1473	0.39496166	[29,]	1334	0.04558333
[5,]	1671	0.33615965	[30,]	652	0.04524490
[6,]	245	0.32977888	[31,]	579	0.04326108
[7,]	1771	0.31742483	[32,]	1843	0.04204225
[8,]	1042	0.21225761	[33,]	1996	0.04173801
[9,]	1423	0.17394956	[34,]	1659	0.04006547
[10,]	1772	0.15830116	[35,]	1608	0.03942726
[11,]	780	0.14561697	[36,]	1383	0.03877650
[12,]	1002	0.14555698	[37,]	576	0.03850951
[13,]	765	0.12076940	[38,]	1485	0.03725977
[14,]	1346	0.11242176	[39,]	1839	0.03673719
[15,]	391	0.10737504	[40,]	489	0.03673509
[16,]	1935	0.10683982	[41,]	1411	0.03544267
[17,]	1060	0.10073807	[42,]	914	0.03436675
[18,]	1058	0.09216809	[43,]	1998	0.03381669
[19,]	1870	0.08976465	[44,]	1983	0.03376904
[20,]	513	0.08882809	[45,]	31	0.03314499
[21,]	698	0.07800680	[46,]	1304	0.03143268
[22,]	267	0.07585334	[47,]	1419	0.03112723
[23,]	822	0.05993884	[48,]	143	0.03110503
[24,]	1466	0.05887480	[49,]	992	0.03095938
[25,]	1487	0.05530371	[50,]	1067	0.03063700

As presented, after the training, three indicators show more importance (import>0.4) than other variables, and from indicator gene 493, there is a sharp drop in variable importance.

4.2Assess the model with test data

Confusion matrix

actual \ predict	0	1
0	6	0
1	2	11

Error of Forest: 0.1204691, Classification Error: 0.1052632, Accuracy: 0.8947368
The classification efficiency of Random Tree is much better than Decision Tree:
0. 8947368> 0. 7894737.

5. k-Nearest-Neighbor Techniques (KNN)

5.1 Introduction of KNN

The nearest neighbor method (Fix and Hodges (1951), see also Cover and Hart(1967)) represents one of the simplest and most intuitive techniques in the field of statistical discrimination. It is a nonparametric method, where a new observation is placed into the class of the observation from the learning set that is closest to the new observation.
The key point: Determine the type of test sample according to the adjacent samples. Adjacent samples are the k samples closest to it, which are determined by calculating their distances from all known samples. There are many ways to calculate distance.[3]

The Euclidean distance is generally used in KNN, that is, the space distance between two points, which is the L2 norm of the difference between two points.

$$d(X_i, X_j) = ||X_i - X_j|| = \{\sum_{s=1}^{p}(x_{is} - x_{js})^2\}^{1/2}$$

where $X_i = (x_{i1}, ..., x_{ip})$ and $X_j = (x_{j1}, ..., x_{jp})$, or its absolute distance

$$d(X_i, X_j) = \sum_{s=1}^{p}|x_{is} - x_{js}|.$$

A general expression: (Minkowski distance)

$$d(X_i, X_j) = \{\sum_{s=1}^{p}|x_{is} - x_{js}|^q\}^{1/q} = \|X_i - X_j\|_q.$$

which is also called L_q-norm/distance or $q-$norm/distance.

Three elements of kNN algorithm: k value selection, distance measurement and classification decision rules all have important influence on classification results.

5.2 Selection of k

	errorKnn	classificationError
k=1	0.21052632	0.21052632
k=2	0.10526316	0.21052632
k=3	0.08187135	0.10526316
k=4	0.06907895	0.05263158
k=5	0.06736842	0.05263158
k=6	0.06286550	0.05263158
k=7	0.08915145	0.05263158
k=8	0.10032895	0.10526316
k=9	0.12020793	0.05263158
k=10	0.13105263	0.05263158

It shows the change of classification error is not linear with the selection of k, and one consideration to choose the “best” k:

The “best k” should be $k(x) = arg\min_k MSE_k(x)$

where

$$MSE_k(x) = E\{m(x) - \hat{m}_k(x)\}^2$$

If we want to get an optimal classification efficiency, when k=4 the result is satisfactory enough, so we have a comparison of k=4 & k=10.

Confusion matrix with KNN (k=4)

	predicted	
actual	0	1
0	6	0
1	1	12

Confusion matrix for KNN model (k=10)

```
          predicted
actual   0  1
     0   5  1
     1   0 13
```

Comparing their Confusion matrix, the main difference lies in their performance of Type 1 error and Type 2 error. If combined with the clinical background, we would think that it is more serious to misdiagnose a cancer person as being free of disease, so the model with k=10 is preferred.

5.3 Assessment of Model

Results of the model with k=10
Error of Forest: 0.1310526, Classification Error: 0.05263158, Accuracy: 0.9473684
The model fitted with KNN has a very satisfactory efficiency for classification.

6. Model Comparison

6.1 Comparison of Accuracy

Model Name	Accuracy
LASSO Penalized Logistic Regression	0. 8947368
CART	0. 7894737
Random forest	0. 8947368
KNN	0. 9473684

(1) LASSO penalized logistic regression

By shrinking the coefficients of most of 2000 genetic features to 0, LASSO Penalized Logistic Regression selects only a few important genes and become the most interpretive model so far. But when the data is non-linear and a more complex model is needed, it is not flexible and time-consuming.

(2) CART

CART is a model with low computational complexity and its results can be visualized. However, when it comes to accuracy and dealing with over-fitting problem, it is not satisfactory.

(3) Random forest

Due to the independence rule, Random forest is the fastest algorithm and perform the best in algorithm efficiency. In practice, this model has its own advantages in avoiding overfitting. Nonetheless, for small dataset or low dimensional data, it is hard to control the internal operation because of the randomness, though repeated tests of parameters and setting seeds may be helpful.

(4) KNN

Based on accuracy, we choose k-Nearest-Neighbor Techniques (KNN) as our best model, which is also not sensitive to outliers. Even so, its model performance is

greatly influenced by parameter selection, like k value or distance choice. Meanwhile, its practical application is limited by cumbersome calculation.

6.2 Variable selection and interpretation

We select those common significant genes in different models and explain them with the file 'names'. Models help us select those important genes that contribute to the canceration of normal cells and have detectable expression for cancer prediction.

(1) LASSO Penalized Logistic Regression: 1870, 1772, 1771, 493, 365

365	Hsa.821	X14958	gene	1			Human hmgI mRNA for high mobility group protein Y.
493	Hsa.37937	R87126	3' UTR	2a	197371	"MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus)	
1771	Hsa.601	J05032	gene	1		"Human aspartyl-tRNA synthetase alpha-2 subunit mRNA, complete cds.	
1771	Hsa.6814	H08393	3' UTR	2a	45395	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)	
1771	Hsa.1660	H55916	3' UTR	1	204131	"PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR (HUMAN);.	

(2) Influential variables selected with Decision Tree: 249, 1419

249	Hsa.8147	M63391	gene	1		"Human desmin gene, complete cds.
1419	Hsa.990	M31606	gene	1		"PHOSPHORYLASE B KINASE GAMMA CATALYTIC CHAIN, TESTIS (HUMAN);.

(3) Important variables selected with the Random Tree: 493, 249, 1582,1473

	gene_index2	import
[1,]	493	1.07629713
[2,]	249	0.49344824
[3,]	1582	0.46303861
[4,]	1473	0.39496166

249	Hsa.8147	M63391	gene	1		"Human desmin gene, complete cds.
493	Hsa.37937	R87126	3' UTR	2a	197371	"MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus)
1473	Hsa.1410	R54097	3' UTR	1	41511	TRANSLATIONAL INITIATION FACTOR 2 BETA SUBUNIT (HUMAN);.
1582	Hsa.2928	X63629	gene	1		H.sapiens mRNA for p cadherin.

The selection results above are consistent to the findings of numerous genetic studies. For example, the listed gene 365 encodes a chromatin-associated protein involved in the regulation of gene transcription, integration of retroviruses into chromosomes, and the metastatic progression of cancer cells.[4] It is indeed an influential factor for the prediction of tumor cell. Other explanations for variables can be found on specialized genomics websites. [5]

7. Conclusion

Gene expression data is very "hot" topic nowadays and more and more related analysis methods are applied to explain biomedical phenomena and help us predict and treat diseases. The goal is to learn which genes are associated with the various diseases or other states associated with the cell lines. This project just provides several common classification methods for processing genetic data, producing statistically significant results and model interpretation. However, the mechanisms by which genes act on human health or cytopathic disease are not explained by one or two

models. In practical application, the advantages of different models can be combined to help us analyze data. With the era of big data and the popularity of biological genetic engineering, more in-depth studies are worth exploring.

Reference

[1] http://genomics-pubs.princeton.edu/oncology/affydata/index.html
[2] https://www.cnblogs.com/karlpearson/p/6224148.html
[3] https://blog.csdn.net/albert201605/article/details/81040556
[4] https://www.cancer.net/navigating-cancer-care/cancer-basics/genetics/genetics-cancer
[5] https://genome.ucsc.edu/cgi-bin/hgGene?db=hg38&hgg_chrom=chr6&hgg_gene=ENST00000311487.9&hgg_start=34236872&hgg_type=knownGene