Statistics - Week 6

Jimmy Tsz Ming Yue*

University of Sydney

jyue6728@uni.sydney.edu.au

Semester 2 Statistics 2018

Contents

1 Training error Vs Test error

> **Definition. Test Error** is the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.

> **Definition. Training Error** is calculated by applying statistical learning to the observations used in its training

Often, training error rate and test error rates are quite different, in particular the training error can dramatically underestimate the test error rate.

1.1 Validation Set Approach

To minimise these errors, the best solution is to take a large designated test set. But this is not always possible, as such we need alternative methods such as BIC, which makes mathematical adjustments to the training error rate in order to estimate the test error rate, which is the subject of later discussion. However there are other methods,

> **Validation Set Approach** which comprises of considering a class of methods that estimate the test error by holding out a subset of the training observations from the fitted process, and

[*] 440159151

> then applying the statistical learning method to those held out observations. We first randomly divide the available set of samples into two parts: a training set and a validation or hold-out set. The model is then fit on the training set and the fitted model is used to predict the respones for the observations in the validation set. The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response and misclassification in the case of a qualitative (discrete) response.

1.2 Drawbacks of validation set approach

The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which obserations are included in the validation set. IN the validation approach, only a subset of the observations (the ones included in the training set) are used to fit the model. This suggests that the validation set error may tend to overestimate the test error for the model fit on the entire data set. As such the widely used approach for estimating test error is:

> **K - Fold cross validation** Estimates can be used to select best model and to give an idea of the test error of the final chosen model. Idea is to randomly divide the data into k equal-sized parts. Leave out part k, fit the model to the other $K-1$ parts (combined), and then obtain predictiosn for the left-out k-th part. This is done in turn for each part $k = 1, 2, \ldots K$. adn then the results are combined.
> We provide the formulation:
>
> Let the K parts be $C_1, C_2, \ldots, C_K$ where C_k denotes the indicies of the observations in part k. There are n_k observations in part k: if n is a multiple of K, then $n_k = n/K$. Then compute:
>
> $$\text{CV}_(K) = \sum_{k=1}^{K} \frac{n_k}{n} \text{MSE}_k \tag{1}$$
>
> where $\text{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$ and $\hat{y}_i$ is the fit for observation i, obtained from the data with part k removed. Setting $K = n$ yields n-fold or leave one out cross validation (LOOCV)

We can extend this for classification problems:

> **Cross - Validation for classification** We divide the data into K roughly equal-sized parts $C_1, \ldots, C_K$. C_k denotes the indicies of the observations in part k. There are n_k observations in part k. If n is a multiple of K, then $n_k = n/K$. Then let us compute:
>
> $$\text{CV}_(K) = \sum_{k=1}^{K} \frac{n_k}{n} \text{Err}_k \tag{2}$$
>
> where $\text{Err}_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i) / n_k$
> Then the estimated standard deviation of CV_k is:
>
> $$\hat{\text{SE}}(\text{CV}_K) = \sqrt{\sum_{k=1}^{K} (\text{Err} - \bar{\text{Err}})^2 / (K-1)} \tag{3}$$

1.3 Overall Classification accuracy rate

$$\text{ACC} = 1 - \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq \hat{y}_i) \tag{4}$$

There are several disadvantages:

1. Makes no distinctions about the type of errors being made. In spam filtering, the cost of erroneous deleting an important email is likely to be higher than incorrectly allowing a spam email past a filter.

2. Does not consider the natural frequencies of each class.

1.4 Confusion Matrix

1.5 Sensitivity and Specificity

Definition. Accuracy

$$\text{ACC} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{5}$$

Definition. Sensitvity

$$Sen = \frac{TP}{(TP + FN)} \tag{6}$$

Definition. Specificity

$$Spe = \frac{TN}{(TN + FP)} \tag{7}$$

Definition. F_1

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{8}$$

which is the harmonic mean

Definition. GM

$$GM = \sqrt{\frac{TP}{TP + FN}\frac{TP}{TP + FP}} \tag{9}$$

2 The Bootstrap

2.1 Boostrap

Definition. Boostrap The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertaintiy associated with a given estimator or statistical learning method.

For example it can provide an estimate of the standard error of a coeffficien, or a confidence interval for that coeffficient.

$$y_i = \beta_0 + \beta_1 x_i + e_i \tag{10}$$

2.2 A simple example

Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively where X and Y are random quantities. We will invest a fraction α of our money in X, and will invest the remaining $1-\alpha$ in Y. We wish to choose α to minimise the total risk or varaince, of our investtment. In other words, we want to minimise $\mathrm{Var}(\alpha X + (1-\alpha)Y)$. We can show that the value that minimises this risk is subsequently given by;

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}} \tag{11}$$

where $\sigma_X^2 = \mathrm{Var}(X)$, $\sigma_Y^2 = \mathrm{Var}(Y)$ and $\sigma_{XY} = \mathrm{Cov}(X, Y)$.

But the values of σ_X^2, σ_Y^2 and σ_{XY} are unknown. We can compute estimates for these quantites; $\hat{\sigma}_X^2, \hat{\sigma}_X^2$ and $\hat{\sigma}_{XY}$, using a data set that contains measurements for X and Y. We can then estimate the value of α that minimises the variance of our investment using:

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}} \tag{12}$$

To estimate the standard deviation of $\hat{\alpha}$, we repeated the process of simulating 100 paired observations of X and Y, and estimating α 1000 times. We thereby obtained 1000 estimates for α, which we can call $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$. The mean over all 1000 estimates for α is:

$$\bar{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996 \tag{13}$$

very close to $\alpha = 0.6$, and the standard deviation of the estimates is:

$$\sqrt{\frac{1}{1000-1} \sum_{r=1}^{1000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083 \tag{14}$$

This gives us a very good idea of the accuracy of $\hat{\alpha}$: $\mathrm{SE}(\hat{\alpha}) \approx 0.083$. So roughly speaking, for a random sample from the population, we would expect $\hat{\alpha}$ to differ from α by approximately 0.08, on average.

2.3 The real world sitatuon

The procedure outlined above cannot be applied, because for real data we cannot generate new samples from the original population.

However, the bootstrap approach allws us to use a computer to mimic the process of obtaining new data sets so that we can estimate the variability of our estimate without generating additional samples. Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set with replacement. Each of these boostrap data sets is created by samping with replacement, and is the same size as our original dataset. As a result some observations may appear more than once in a given boostrap data set and some not at all.

2.4 Generic Boostrap Procedure

Denoting the first boostrap data set by Z^{*^1} we use Z^{*^1} to produce a new boostrap estimate for α, which we call $\hat{\alpha}^{*^1}$. This procedure is repeated B times for some large value of B (say 100 or 1000), in order to produce B different boostrap data sets $Z^{*^1}, Z^{*^2}, \ldots, Z^{*^B}$ and B different corresponding α estimates; $\hat{\alpha^{*1}}\hat{\alpha^{*2}}\hat{\alpha^{*B}}$. We estimate the standard error of these boostrap estimates using the formula,

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\alpha^{*^r} - \bar{\hat{\alpha}}^{*})^2} \tag{15}$$

THis serves as an estimate of the stnadard error of $\hat{\alpha}$ estimated from the original data set.

3 Tutorial

The “breast.txt” dataset contains benign and malignant breast tumour samples. Each sample is measured by various factors including:

1. Clump Thickness
2. Uniformity of Cell Size
3. Uniformity of Cell Shape
4. Marginal Adhesion
5. Single Epithelial Cell Size
6. Bare Nuclei
7. Bland Chromatin
8. Normal Nucleoli
9. Mitoses

The last coclumn contains class lavels with ”1” being malignant and ”0” as benign:

1. Download the breast dataset file from the course webpage

Solution. We first import the dataset

```
> source("functions_w6.R")
> library(MASS)
> library(mlbench)
> library(class)
> breast <- read.table("breast.txt", sep ="\t", header = TRUE)
> df <- data.frame(breast)
> Column.names <- c("Clump_Thickness", "Uniformity_Size", "Uniformity_Shape",
+ "Marginal_Adhesion", "Single_Ep_Size", "Bare_Nuclei", "Bland_Chromatin",
+ "Normal_Nucleoli", "Mitoses", "Class")
> colnames(df) <- Column.names
> dim(df)
```

```
[1] 683  10
```

□

2. Design a 10-fold cross validation procedure to evaluate logistic regression and k-NN $k = 3$ classification accuracy.

Solution:

```
> library(caret)
> set.seed(1)
> fold <- createFolds(df$Class, k=10)
> knn.TP <- knn.TN <- knn.FP <- knn.FN <- c()
> logit.TP <- logit.TN <- logit.FP <- logit.FN <- c()
> df$Class <- as.factor(df$Class)
> for(i in 1:length(fold)){
+   truth <- df$Class[fold[[i]]]
+
+
+
+ logit.model <-train(Class~. ,data=df[-fold[[i]],], method = "glm")
+ preds <- predict(logit.model, newdata=df[fold[[i]],-10])
+ logit.TP <- c(logit.TP, sum((truth == preds)[truth == "0"]))
+ logit.TN <- c(logit.TN, sum((truth == preds)[truth == "1"]))
+ logit.FP <- c(logit.FP, sum((truth != preds)[truth == "1"]))
+ logit.FN <- c(logit.FN, sum((truth != preds)[truth == "0"]))
+
+
+
+
+
+ preds <- knn(df[-fold[[i]],-10], df[fold[[i]],-10], df$Class[-fold[[i]]], k=3)
+     knn.TP <- c(knn.TP, sum((truth == preds)[truth == "0"]))
+     knn.TN <- c(knn.TN, sum((truth == preds)[truth == "1"]))
+     knn.FP <- c(knn.FP, sum((truth != preds)[truth == "1"]))
+     knn.FN <- c(knn.FN, sum((truth != preds)[truth == "0"]))
+
+ }
> evaluate(knn.TN, knn.FP, knn.TP, knn.FN)

sen: 0.973 sep: 0.959 F1: 0.975 GM: 0.975

> evaluate(logit.TN, logit.FP, logit.TP, logit.FN)

sen: 0.977 sep: 0.95 F1: 0.975 GM: 0.975

>
>
```

□

3. Calcuate TP, TN, FP, FN and compute sensitivity and specificity for each classifier

> **Solution.** See above
>
> □

4. Compute F1 scare and compare the two classifiers.

> **Solution.** See above, we can see that the F1 score is the same above with only the sensitivity and sepicificity being fundamentally different. According to such metrics it is hard to compare as to wheter the classification is better in either scenarios, althouth we can see marginally that the knn method is better at looking at the rates of benign, whereas the logreg method is better looking at the malignant □