Introduction

Classification and clustering are two machine learning methods to find the patterns in dataset. The main difference between the two methods is whether we need the response variable. Classification is a supervised machine learning which requires the access to the response variables and group the observations based on the predefined characteristics in the training set and test on the test set. On the other hand, clustering go beyond the predetermined characteristics and identify the similarities between observations. The reason clustering is unsupervised machine learning is we use algorithm to group the observations based on the similar characteristics which doesn't require a response variable as output to define the accuracy of the grouping.

Data Description

Caravan dataset was provided by Dutch data mining company Sentient Machine Research in 2000. The dataset includes 85 predictors to measure the demographic characteristics of 5822 individuals. The response variable is Purchase, which indicates whether a given individual purchases a caravan insurance policy.

Clustering Method

Clustering is one of the most comment methods for finding the cluster(subgroups) in a dataset. When we try to cluster the observations of a dataset, we focusing on seeking the partition them into different groups. The goal of clustering is to create homogeneous group such that the observations within each cluster are similar while the observations in different clusters are dissimilar to each other. In this section, we will focus on two of the most popular use methods: The K-means Clustering method and Hierarchical Clustering method.

1. Definitions

Before we hit the methods, let's firstly identify some definitions:

Similarity and dissimilarity:

1. *By distance*: If x, y are two points in R_p, a dissimilarity d has the following properties: $d(x,y) \geq 0, d(x,x) = 0 \; and \; d(x,y) = d(y,x)$
2. *Simple correlations*: $d(x_i, y_i) = 1 - r_{ij}$ where $R = (r_{ij})$ is the sample correlation matrix

Distance: In this report, we use Euclidean distance to define the distance.

$$\boldsymbol{d(x,y) = \sqrt{(x-y)^T(x-y)} = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \cdots + (x_p-y_p)^2}}$$

Here x, y are p-dimensional observation points.

Centroid: the vector of averages of the p features in the cluster.

ESS: The sum of the squared Euclidean distances of each observation to the centroid in their cluster. Formular: $ess = \sum_{i=1}^{k} \sum_{j \in J_i} (x_j - \overline{x_\iota})^T (x_j - \overline{x_\iota})$ here j is all observations in the $i \; th$ cluster.

2. K-means clustering

K-means clustering is a straightforward approach for partitioning a dataset into number of K distinct, non-overlapping clusters. K-means clustering procedure can also be treated as

the mathematical problem as: In each cluster, there are k sets write as $C_1, \ldots, C_k$, those sets containing the indices of the observations. These sets satisfy the properties below:

1. $C_1 \cup C_2 \cup \ldots \cup C_k = \{1, \ldots . n\}$ which means all the observation belongs to at least one of the K clusters.
2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$, which means the clusters are non-overlapping: no observation belongs to more than one cluster.

Recall the goal of clustering is to create homogeneous group. In K-means clustering, a good clustering is one for the within-cluster variation is as small as possible. Here, we define the within-cluster variation for cluster C_k is measuring the amount of the observations within a cluster different from each other ($W(C_K = \frac{1}{|C_K|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2$). Therefore, the problem can be writing as: $minimize\{from\ C_1, \ldots, C_K\}\{\sum_{k=1}^{K} W(C_k)\}$ also known as: how to partition the observations into K clusters make that the total within-cluster variation, summed over all K clusters as small as possible? The answer is the algorithm shows below.

2.1 Algorithm

1. Firstly, specify the desired number of clusters(K).
2. Secondly, initialize the clusters by either:
   a) Form an initial random assignment of the items into K clusters.
   b) Choose K "seeds" $x_i,\ i = 1,2 \ldots ., k$ and form k initial clusters by assigning each item to a cluster associated with the closest seed.
3. Iterate until without any observation changing between different groups:
   a) For each of the K clusters, computer the centroids $\overline{x_i}, i = 1,2, \ldots, k$
   b) For each observation, compute the distance of the item to each centroid and assign the item to the cluster associated with the closest centroid. (Here closest is defined using Euclidean distance).

2.2 K-means Clustering in R

The key of applying K-means clustering in R is use `kmeans()` function.

Performs K-means clustering by using `kmeans()` function

There are three inputs was required for `knn()`:

1. The dataset which have already removed the response variable
2. Number of K
3. Set `nstart` argument

a) If `nstart` set to a value greater than one is used, then K-means clustering will be performed by using multiple random assignments in initialize cluster step. Besides, the `kmeans()` function will output only the best results.

```
kmeans.re <- kmeans(Caravan_1, centers = 3, nstart = 20)
#Here we set the K=3, the output would have three clusters.
kmeans.re$tot.withinss
#Total within-cluster sum of squares, which we trying to minimize by performing K-means clustering
kmeans.re
#List the different clusters' size, means, clustering vector, within sum of squares by cluster and available components
#Note: From the output of kmeans.re (appendix table4), we could say that: By assigning the samples to 3 clusters rather than 5822 (number of samples) clusters achieved a reduction in sums of squares of 58.4 %.
```

3. Hierarchical clustering

Hierarchical clustering is one of the clustering methods that attempt to find "good" clusters. Different from K-means clustering, hierarchical clustering produces dendrograms which start at the bottom and go up. Since each observation is treated as an own cluster, at the bottom of the dendrograms, each node has a single observation.

3.1 Define the notion of Linkages:

(Evaluate dissimilarity between clusters)

1. *Single linkage method (Nearest neighbour):* defined as the minimum dissimilarity between points (x and y) in clusters (A and B): $D(A,B) = \min\{d(x,y): x \in A, y \in B\}$
2. *Complete linkage method (Furthest neighbour):* defined as the maximum dissimilarity between points (x and y) in clusters (A and B): $D(A,B) = \max\{d(x,y): x \in A, y \in B\}$
3. *Average linkage method (Group average):* defined as the average of the n_A and n_B dissimilarities between the points $\{x_i\}_{i=1}^{n_A}$ in A and the points $\{y_i\}_{i=1}^{n_B}$ in B:

$$D(A,B) = \frac{1}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} d(x_i, y_i)$$

3.2 Algorism

1. Firstly, choose a dissimilarity and method for computing the distance between clusters.
   a) In this report, we use Euclidean distance to measure of all $(n\ choose\ 2) = n(n-1)/2$ pairwise dissimilarities.
2. Secondly, choose an initial set of clusters, treat each observation as a single group.
3. While the number of clusters is larger than one:
   a) In the current set of clusters, compute the distance between every pair of clusters.
   b) Merging the two clusters that are most similar (that is, with the smallest distance).

c) Update the number of clusters by reduce one.

3.3 Hierarchical clustering in R

The key of applying K-means clustering in R is use `hclust()` function.

Performs Hierarchical clustering by using `hclust()` function:

We need to input the data and set the linkages method

```
#Don't forget to standardize the data before we do clustering
data.dist=dist(standardized.X)
hc.complete<-hclust(data.dist)                          #Complete linkage
hc.average<-hclust(data.dist,method="average") #Average linkage
hc.single<-hclust(data.dist,method="single")    #Single linkage
```

Visualize the plots: (output show in appendix figure5)

```
par(mfrow=c(1,3))                #Create a 3*3 matrix
#dendrograms obtained using plot() function
plot(hc.complete,main="Complete Linkage ",xlab="", sub="",ylab="")
plot(hc.average , main="average Linkage ", xlab="", sub="",ylab="")
plot(hc.single , main="single Linkage ", xlab="", sub="",ylab="")
```

From the output of the Hierarchical clustering, we can observe that compare three different methods, the complete and average linkage provide the clusters look more balanced, attractive.

Some other useful functions:

`#cutree()` function used to determine the cluster labels for each observation associated with a given cut of the dendrogram

```
cutree(hc.complete,2)
```

Classification Method

Classification is a method which used for predicting categorical responses variable. The process of predicting the response variable could be referred to "classifying" the observation due to the process involves assigning the observation to a class. We can also think as response variable takes on values from K's possible classes.

Recall that in regression, there is a continuous distribution of response variables across the universe with model: $Y = g(X) + \delta$, in this regression model, δ is the irreducible error. In classification, there is a discrete distribution across the K possible response variables at point $\vec{X} = \overrightarrow{x_0}$. We can also represent the classification process by:

$$p_1(x_0) = P(Y = 1|X = x_0) \cdots\cdots P_K(x_0) = P(Y = K|X = x_0)$$

The class that has the highest probability was cheated as the "true class" at any $X = x_0$ point in the universe. There are many different classification classifiers, such as Artificial neural networks, Boosting, and K-nearest neighbor, etc. The same objective of the classifier is finding a function f(X) that returns the most likely class y. In this section, we will focus on one of the most popular use methods: The K-Nearest Neighbour method.

1. K-Nearest Neighbour (KNN)

The K-nearest neighbour algorithm is also known as KNN. KNN is a machine learning algorithm designed for classification and regression classifiers. Contrary to the other 2 clustering algorithms, KNN is a supervised classifier that uses the average of the local neighbors to classify and predict the value. The model is based on a simple assumption that "similar points can be found near one another" which means after we figure out the neighbour of one observation, we should get an estimated value and characteristics of that observation.

1.1 Conditional probability

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

Here: N_0 is the set of K observations that are the closest to x_0
$I(y_i = j)$ is an indicator variable: If a given observation (x_i, y_i) in N_0 is a member of class j, then indicator variable evaluates to 1, otherwise evaluates to 0.

1.2 Algorithm

1. The first thing is to choose K, which also means the size of the "neighborhood". In terms of the number of points which will get to "vote" on the class.
2. The next step is select the point x_0 as the point that we are trying to predict in the p-dimensional space of X.
3. Then identify the K closest points to x_0 the neighborhood and write as $N_m(x_0)$.
4. Count the number of points of each class inside $N_m(x_0)$.
5. $Y(x_0)$ is the class with the highest count.

1.3 Properties

a. The choice of K

Since K is a tuning parameter, the choice of K is crucial and connecting with bias-variance trade-off. Firstly, let's consider two situations:

1. If K =1, the prediction is the observation y that has the closest x to x_0, which means the model are extreme interpolation. In this situation, our model is overfitting. To conclude, a smaller K means the models more flexibility, which leads high variance and low bias.
2. If K=sample size(n), the prediction is the average point of the whole sample, which is $\bar{y}$. In this situation, our model is underfitting since every prediction is equal to $\bar{y}$. To conclude, a larger K means the model less flexibility (extreme smoothing), which leads high bias and low variance.

To conclude above, the optimal K is the one that minimizes the MSE to find the best bias-variance tradeoff.

b. The definition of distance

In this report, we use Euclidean distance to define the distance. Which means the closest distance between x_i and x_0 is:

$$|x_0 - x_i| = \sqrt{(x_{0,1} - x_{i,1})^2 + \cdots + (x_{0,q} - x_{i,q})^2}$$

Here x_i is of dimension q

1.4 KNN in R

There are four main steps when doing KNN in R:

Step1: Standardize data

When two variables change in different rate, we could use `scale()` function to standardize the data to make all variables on a comparable scale. After using `scale()`, all the variables are given zero mean and a standard deviation of one.

```
standardized.X = scale(Caravan[,-86])
#Normalize the observations except the response variable (in column 86)
```

Step2: Split the observations into test set and training set.

Step3: Fit the KNN model by using `knn()` function

We will perform KNN method by using function `knn()` from `class` library. There are four inputs was required for `knn()`:

```
test=1:1000
train.X= standardized.X[-test ,]
test.X= standardized.X[test ,]
train.Y=Purchase [-test]
test.Y=Purchase [test]
#Here we split the first 1000 observations as the test set, and the remind as the training
```

4. A matrix containing the predictors associated with the training data
5. A matrix containing the predictors associated with the data for which we want to predict
6. A vector containing the class labels for the training observations
7. Value of K, the number of nearest neighbors to be used by the classifier

```
knn.pred=knn(train.X,test.X,train.Y,k=2)
mean(test.Y!=knn.pred)
mean(test.Y!="No")
table(knn.pred ,test.Y)
#Fit KNN model on the training set using K=2
```

Step4: Evaluate the performance on the data set

We will compare the performances in different K value by checking the table. Here is one of the outputs when we set K=2:

```
>knn.pred=knn(train.X,test.X,train.Y,k=2)
>table(knn.pred ,test.Y)
          test.Y
knn.pred  No  Yes
   No     886 52
   Yes    55  7
```

#893 out of 1000 test data are successfully predicted.

#Here the misclassification rate is: $\frac{52+55}{1000} = 0.107$ (10.7%)

NOTE: there are more tables (Table 1-3) shows in the appendix.

Comparison

The different between Clustering methods and Classification methods:

1. Clustering is unsupervised learning
   a) Clustering is a descriptive method, there is no response variable Y and no error rate perform, which means we cannot say "right" or "wrong" to the model.
2. Classification is supervised learning
   a) Response variable Y is a nominal variable

The different between Hierarchical clustering and K-means clustering:

1. K-means clustering requires choose the K (number of clusters); Hierarchical clustering does not require that.

Conclusion

In conclusion, the two methods are designed for different scenarios. Classification is designed for classifying the observations belonging to which group based on the output. Clustering is grouping the observations with the same characteristic into the same cluster. Many trending companies adapting these methods to provide better services. For example, Netflix use clustering methods to cluster customer based on their taste in the shows to create audience-oriented Netflix original series. Many financial firm use classification methods to identify financial fraud based on their past behaviors. This report introduces three simple and fundamental algorithms and I believe in the future, there will be more companies and people who realize the importance of using classification and clustering machine learning algorithms.