University of Glasgow | School of Computing Science

CSC3005 Laboratory/Tutorial 6: Clustering Analysis and Anomaly Detection

1. K Means Clustering

The k-means clustering algorithm represents each cluster by its corresponding cluster centroid. The algorithm would partition the input data into k disjoint clusters by iteratively applying the following two steps:

a) Form k clusters by assigning each instance to its nearest centroid.
b) Recompute the centroid of each cluster.

We will use sample movie ratings dataset to perform k-means clustering.

Step 1: Generate sample movie rating as follows using pandas dataframe :

	user	Avenger	X-Men	The Ring	Train to Busan
0	Forest	5	5	2	1
1	Jeannie	4	5	3	2
2	Malcom	4	4	4	3
3	Sye Loong	2	2	4	5
4	CaoQi	1	2	3	4
5	Ryan	2	1	5	5

Step 2: Remove useless feature "user" and perform clustering with k=2

a) Import cluster from sklearn
b) use drop method to drop feature user
c) configure the number of clusters to 2, maximum iteration to 50 and fit the movie data in step 1
d) show the clustering result with the label generated by K means

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.fit

What is your observation? Are you able to know how the means is being calculated and how the two clusters are being formed?

Step 3: Show the centroid of the two clusters.

a) Use the attributes cluster_centers from k_means library.
b) Show the centroid for the two clusters

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.fit

What has you observed? Can you derived the centroid values as shown?

Step 4 : Test with new data to check the two cluster formed is robust by the generating following new movie rating data

	user	Avenger	X-Men	The Ring	Train to Busan	Cluster ID
0	Lawrence	4	5	1	2	0
1	Alicia	3	2	4	4	1
2	Jacob	2	3	4	1	0
3	Thiru	3	2	3	3	1
4	Ifa	5	4	1	4	0

a) Using predict method on the new data to predict the new labels for the new data.

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.predict

b) Show the result using pandas dataframe.

Step 5: Determine the optimal choice of k that minimize the error

a) Repeat step 2 by applying k-means with varying number of clusters from 1 to 6 and compute their corresponding sum-of-squared errors (SSE) using k_means.inertia_

b) Plot number of clsuters versus SSE and determine what is the best choice for k value.

2. Hierarchical Clustering

We will explore Exploration of hierarchical clustering algorithms on

a) single link (MIN),
b) complete link (MAX),
c) group average.

Step 1: Import the dataset on mammal classification , T6_vertebate.csv

Step 2: **Single Link**: Import cluster.hierachy libary from scipy

a) Import hierarchy library from scipy.cluster package

   https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage

b) Drop the unwanted feature “Name” and class label
c) Fit the data with “single” link using linkage method
d) Show the dendrogram using

   https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html#scipy.cluster.hierarchy.dendrogram

What is your observation? Can you observe clustering is done by minimum distance between features?

Step 3: Complete link: Fit the data and activate complete (max distance between two clusters)

Repeat step 2 but fit the data with “complete” in the linkage method.

https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage

https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html#scipy.cluster.hierarchy.dendrogram

What is your observation? Can you observe clustering is done by maxmium distance between features?

What is your observation? Can you observe clustering is done by minimum distance between features?

Step 4: Average link: Fit the data and activate complete (max distance between two clusters)

Repeat step 2 but fit the data with “average” in the linkage method.

https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage

https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html#scipy.cluster.hierarchy.dendrogram

What is your observation? Can you observe clustering is done by average distance between all features?

3. Density Based Clustering

Density-based clustering identifies the individual clusters as high-density regions that are separated by regions of low density. DBScan is one of the most popular density based clustering algorithms. In DBScan, data points are classified into 3 types

1)core points,

2)border points,
3)noise points

based on the density of their local neighborhood. The local neighborhood density is defined according to 2 parameters:

1)radius of neighborhood size (eps)
2)minimum number of points in the neighborhood (min_samples).

For this approach, we will use a noisy, 2-dimensional dataset for evaluating DBSCAN algorithm.

Step 1: Import the dataset T6_Q3.data

a) Use pandas dataframe.read.csv to read the T6_Q3.data
b) Scatter plot the data

Step 2: Import the DBSCAN from sklearn.cluster package and fit the data

a) Import DBSCAN from sklearn.cluster package

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

b) Set eps =15.5 and min_samples =5 and fit the data

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN.fit

c) Obtain the cluster labels and use pandas dataframe to scatter plot both the original data and the data cluster labels.

What is your observation? Has DBSCAN clusters the data correctly?

Step 3: Repeat the same process of step 2 using K means clustering

a) Set k= 8, maximum of iteration to 50 and fit the data similarly to Question 1 approach.

What is your observation? Has K means clustering clusters the data correctly?

4. Anomaly Detection: Statistical Approach

Anomaly detection is the task of

1) identifying instances whose characteristics differ significantly from the rest of the data.

2) creating labels for first time unseen data so that to serve as training data for classification analysis.

Statistical approach assumes that the majority of the data instances are governed by some well-known probability distribution, e.g., Binomial or Gaussian distribution. Anomalies can then detected by seeking for observations that do not fit the overall distribution of the data.

We will be detecting anomalous changes in the daily closing prices of various stocks. The input data stocks.csv contains the historical closing prices of stocks for 3 large corporations of IT, Car manufacturer and bank (Microsoft, Ford Motor, and Bank of America).

Step 1: Import T6_Q4_stocks.csv

a) Use pandas dataframe.read.csv to read the T6_Q4_stocks.csv
b) Drop the feature "date"

Step 2: Create rule to detect anomaly.

We can compute the percentage of changes in the daily closing price of each stock as follows:

$$\Delta(\mathrm{t}) = 100 \times \frac{x_t - x_{t-1}}{x_{t-1}}$$

where x_t denotes the price of a stock on day t and x_{t-1} denotes the price on its previous day $t-1$

a) Use pandas dataframe to create the data $\Delta(\mathrm{t})$ for all the three companies

Step 3: Plot distribution of the three companies based on the $\Delta(t)$

a) Use matplotlib to plot the 3D scatter plot of the three companies based on $\Delta(t)$

Step 4: Find the data distribution parameters (mean and variance) of the 3 companies features

Assuming the data follows a multivariate Gaussian distribution, we can compute the mean and covariance matrix of the 3-dimensional data where

$$f(\mathbf{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^N |\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})^T}{2}}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance vector of the 3 companies, $\Delta(t)$

a) Use mean() and cov() on the $\Delta(t)$

Can you understand the covariance between pairs such as MSFT and Ford? Which one are the variances, σ^2 of the three company?

Step 5: Detect Anomaly

To determine the anomalous trading days, we can compute the Mahalanobis distance between the percentage of price change on each day against the mean percentage of price change.

$$\text{Mahalanobis} = (\mathbf{x} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})^T$$

where $\mathbf{x}$ is assumed to be a row vector.

a) Import numpy.matmul() method from numpy library for matrix multiplication

   https://numpy.org/doc/stable/reference/generated/numpy.matmul.html

b) Use numpy.apply_along_axis() for matmul to peform matrix multiplication for row vector of $\Delta(t)$ to obtain anomaly score

   https://numpy.org/doc/stable/reference/generated/numpy.apply_along_axis.html

c) Scatterplot the $\Delta(t)$ of the three companies and the anomaly score that mapped to each $\Delta(t)$

Can you observe the top 2 anomaly data of $\Delta(t)$?

Step 6: Examine the data associated with the top-5 highest anomaly scores

Using pandas dataframe, list the top 5 anomaly score of $\Delta(t)$

What is your observation?

5. Anomaly Detection: Proximity Approach

This is a model-free anomaly detection approach as it does not require constructing an explicit model of the normal class to determine the anomaly score of data instances. k-nearest neighbor approach is to calculate anomaly score. Specifically, a normal instance is expected to have a small distance to its k^{th} nearest neighbor whereas an anomaly is likely to have a large distance to its k^{th} nearest neighbor. Distance-based approach with k=4 to identify the anomalous trading days from the stock market data as described in Question 4.

Step 1: Import the NearestNeighbours from sklearn.neighbours

a) Import NearestNeighbors from sklearn.neighbors
b) Import Numpy
c) Import distance from scipy.spatial
d) fit the $\Delta(t)$ into NearestNeighbors and set knn=4, metric= distance.euclidean

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html

e) set the anomaly score using distance( ) with the 4^{th} neighbor

f) Scatterplot the $\Delta(t)$ of the three companies and the anomaly score that mapped to each $\Delta(t)$

Can you observe the top 5 anomaly data of $\Delta(t)$?

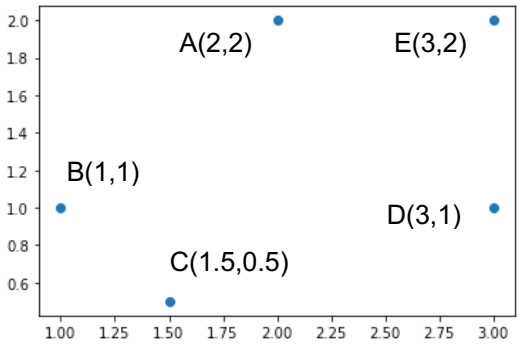
Step 2: Examine the data associated with the top-5 highest anomaly scores

Using pandas dataframe, list the top 5 anomaly score of $\Delta(t)$

What is your observation?

6. Theory on K-Means Clustering

Given the following five points in 2D plot (x, y). Cluster it into two clusters using Euclidean distance



Let the initial centroid/cluster mean be point A and B. Go through 2 iterations to find the two clusters.

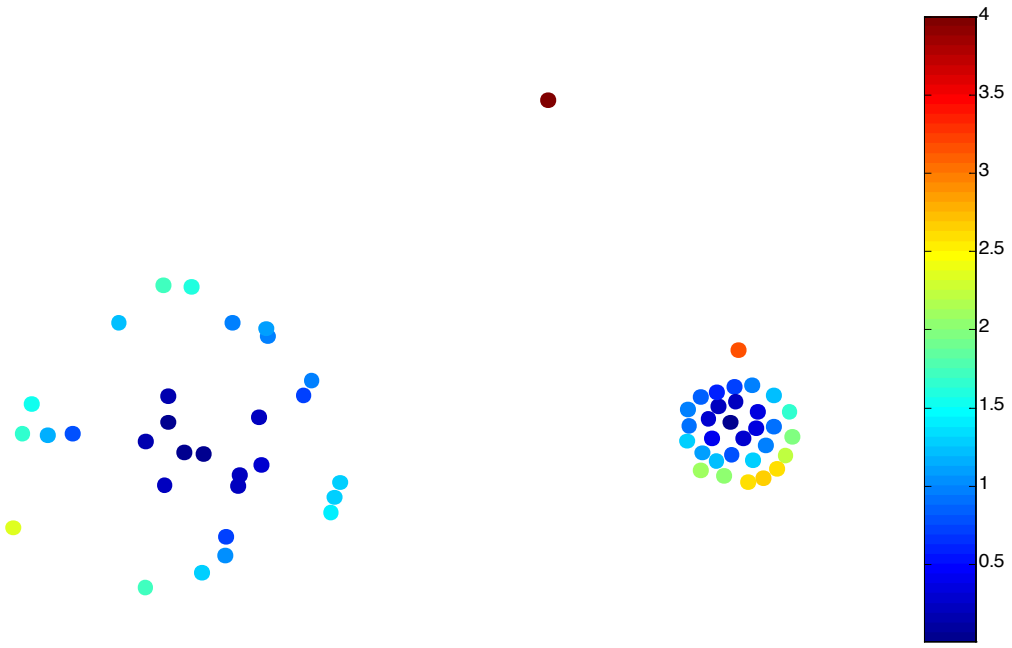
7. Theory on Anomaly Detection using Statistical Test

Many statistical tests for outliers were developed in an environment in which a few hundred observations was a large data set. We explore the limitations of such approaches.

a) For a set of 1,000,000 values, how likely are we to have outliers according to the test that says a value is an outlier if it is more than three standard deviations from the average? (Assume a normal distribution.)

(b) Does the approach that states an outlier is an object of unusually low probability need to be adjusted when dealing with large data sets? If so, how?

8. Theory on Anomaly Detection using K means clustering



Consider the (relative distance) K-means scheme for outlier detection as shown above

a) The points at the bottom of the compact cluster have a somewhat higher outlier score than those points at the top of the compact cluster. Why?

b) Suppose that we choose the number of clusters to be much larger, e.g.,10. Would the proposed technique still be effective in finding the most extreme outlier at the top of the figure? Why or why not?

c) The use of relative distance adjusts for differences in density. Give an example of where such an approach might lead to the wrong conclusion.