

**MSc Electronic and Computer Engineering**

**Data Mining and Machine Learning (2019)**

**Lab 2 – Clustering and PCA**



Group 12

Miaoyu Niu (1893824)

Kai Chen (1948361)

2019-2-21

UNIVERSITY OF BIRMINGHAM

## PART 1: TF-IDF BASED TEXT RETRIEVAL

### Step 1: Compile two C programs agglom.c and k-means.c

Command:

```
cl agglom.c
```

```
cl k-means.c
```

Agglomerative clustering begins by assuming that each data point belongs to its own, unique, one point cluster, and each point is a centroid. Clusters are then combined until the required number of centroids is obtained.

About k-means Clustering, suppose we have determined the number of centroids we need and use  $k$  to represent this number. At the same time, it is assumed that a preliminary estimate of the position of the centroid is made. Then k-means clustering is an iterative process for moving these centroids to reduce distortion.

### Step 2: agglom lab2Data centFile numCent

The first step is to run aggregating clusters on the data in the dataFile until the number of centroids is numCent. The centroid coordinates will be written to the centFile file. The coordinates of the centroid are the fifth column of the zeroth row.

### k-means lab2Data centFile opFile numIter

numIter: the number of iteration =10

(1) numCent = 1

agglom lab2Data centFile 1

k-means lab2Data centFile opFile 10

```
x86 Native Tools Command Prompt for VS 2017
numClus=6
numClus=5
numClus=4
numClus=3
numClus=2

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=1, numCols=5

Results
=====
distortion[0]= 1080967.000000
distortion[1]= 732439.562500
distortion[2]= 732439.562500
distortion[3]= 732439.562500
distortion[4]= 732439.562500
distortion[5]= 732439.562500
distortion[6]= 732439.562500
distortion[7]= 732439.562500
distortion[8]= 732439.562500
distortion[9]= 732439.562500
k-means complete
F:\lab2-2019>
```

Final number is 732439.562500

The list of 10 numbers are the distortion after each iteration.

**(2) numCent =2**

**agglom lab2Data centFile 2**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=7
numClus=6
numClus=5
numClus=4
numClus=3

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=2, numCols=5

Results
=====
distortion[0]= 321272.281250
distortion[1]= 317542.625000
distortion[2]= 317282.718750
distortion[3]= 317257.187500
distortion[4]= 317257.187500
distortion[5]= 317257.187500
distortion[6]= 317257.187500
distortion[7]= 317257.187500
distortion[8]= 317257.187500
distortion[9]= 317257.187500
k-means complete
F:\lab2-2019>
```

Final number is 317257.187500

**(3) numCent =3**

**agglom lab2Data centFile 3**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=8
numClus=7
numClus=6
numClus=5
numClus=4

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=3, numCols=5

Results
=====
distortion[0]= 83273.203125
distortion[1]= 78126.726563
distortion[2]= 78126.726563
distortion[3]= 78126.726563
distortion[4]= 78126.726563
distortion[5]= 78126.726563
distortion[6]= 78126.726563
distortion[7]= 78126.726563
distortion[8]= 78126.726563
distortion[9]= 78126.726563
k-means complete
F:\lab2-2019>
```

Final number is 78126.726563

**(4) numCent =4**

**agglom lab2Data centFile 4**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=9
numClus=8
numClus=7
numClus=6
numClus=5

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=4, numCols=5

Results
=====
distortion[0]= 32769.117188
distortion[1]= 32335.769531
distortion[2]= 32335.769531
distortion[3]= 32335.769531
distortion[4]= 32335.769531
distortion[5]= 32335.769531
distortion[6]= 32335.769531
distortion[7]= 32335.769531
distortion[8]= 32335.769531
distortion[9]= 32335.769531
k-means complete
F:\lab2-2019>
```

Final number is 32335.769531

**(5) numCent =5**

**agglom lab2Data centFile 5**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=10
numClus=9
numClus=8
numClus=7
numClus=6

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=5, numCols=5

Results
=====
distortion[0]= 14179.541016
distortion[1]= 13455.114258
distortion[2]= 13455.114258
distortion[3]= 13455.114258
distortion[4]= 13455.114258
distortion[5]= 13455.114258
distortion[6]= 13455.114258
distortion[7]= 13455.114258
distortion[8]= 13455.114258
distortion[9]= 13455.114258
k-means complete
F:\lab2-2019>
```

Final number is 13455.114258

**(6) numCent =6**

**agglom lab2Data centFile 6**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=11
numClus=10
numClus=9
numClus=8
numClus=7

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=6, numCols=5

Results
=====
distortion[0]= 5795.420410
distortion[1]= 5021.820313
distortion[2]= 5021.820313
distortion[3]= 5021.820313
distortion[4]= 5021.820313
distortion[5]= 5021.820313
distortion[6]= 5021.820313
distortion[7]= 5021.820313
distortion[8]= 5021.820313
distortion[9]= 5021.820313
k-means complete
F:\lab2-2019>
```

Final number is 5021.820313

**(7) numCent =7:**

**agglom lab2Data centFile 7**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=12
numClus=11
numClus=10
numClus=9
numClus=8

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=7, numCols=5

Results
=====
distortion[0]= 6107.503418
distortion[1]= 4581.819824
distortion[2]= 4581.746094
distortion[3]= 4581.746094
distortion[4]= 4581.746094
distortion[5]= 4581.746094
distortion[6]= 4581.746094
distortion[7]= 4581.746094
distortion[8]= 4581.746094
distortion[9]= 4581.746094
k-means complete
F:\lab2-2019>
```

Final number is 4581.746094

**(8) numCent=8:**

**agglom lab2Data centFile 8**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=13
numClus=12
numClus=11
numClus=10
numClus=9

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=8, numCols=5

Results
=====
distortion[0]= 5892.135742
distortion[1]= 3989.890625
distortion[2]= 3987.923340
distortion[3]= 3987.835449
distortion[4]= 3987.835449
distortion[5]= 3987.835449
distortion[6]= 3987.835449
distortion[7]= 3987.835449
distortion[8]= 3987.835449
distortion[9]= 3987.835449
k-means complete
F:\lab2-2019>
```

Final number is 3987.835449

**(9) numCent=9:**

**agglom lab2Data centFile 9**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=14
numClus=13
numClus=12
numClus=11
numClus=10

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=9, numCols=5

Results
=====
distortion[0]= 5663.681152
distortion[1]= 3575.082275
distortion[2]= 3563.111572
distortion[3]= 3558.017334
distortion[4]= 3553.934570
distortion[5]= 3542.091064
distortion[6]= 3530.739746
distortion[7]= 3529.189209
distortion[8]= 3529.085693
distortion[9]= 3528.995850
k-means complete
F:\lab2-2019>
```

Final number is 3528.995850

**(10) numCent=10:**

**agglom lab2Data centFile 10**

**k-means lab2Data centFile opFile 10**

```
x86 Native Tools Command Prompt for VS 2017
numClus=15
numClus=14
numClus=13
numClus=12
numClus=11

F:\lab2-2019>k-means lab2Data centFile opFile 10
open input data file
open input centroid file
open output centroid file
Number of iterations = 10
Data file: numRows=1000 numCols=5
Info: number of data points = 1000
numCentroid=10, numCols=5

Results
=====
distortion[0]= 5470.070313
distortion[1]= 3178.527344
distortion[2]= 3148.433838
distortion[3]= 3138.711914
distortion[4]= 3132.962402
distortion[5]= 3119.700439
distortion[6]= 3107.398926
distortion[7]= 3105.075684
distortion[8]= 3104.753174
distortion[9]= 3104.663086
k-means complete
F:\lab2-2019>
```

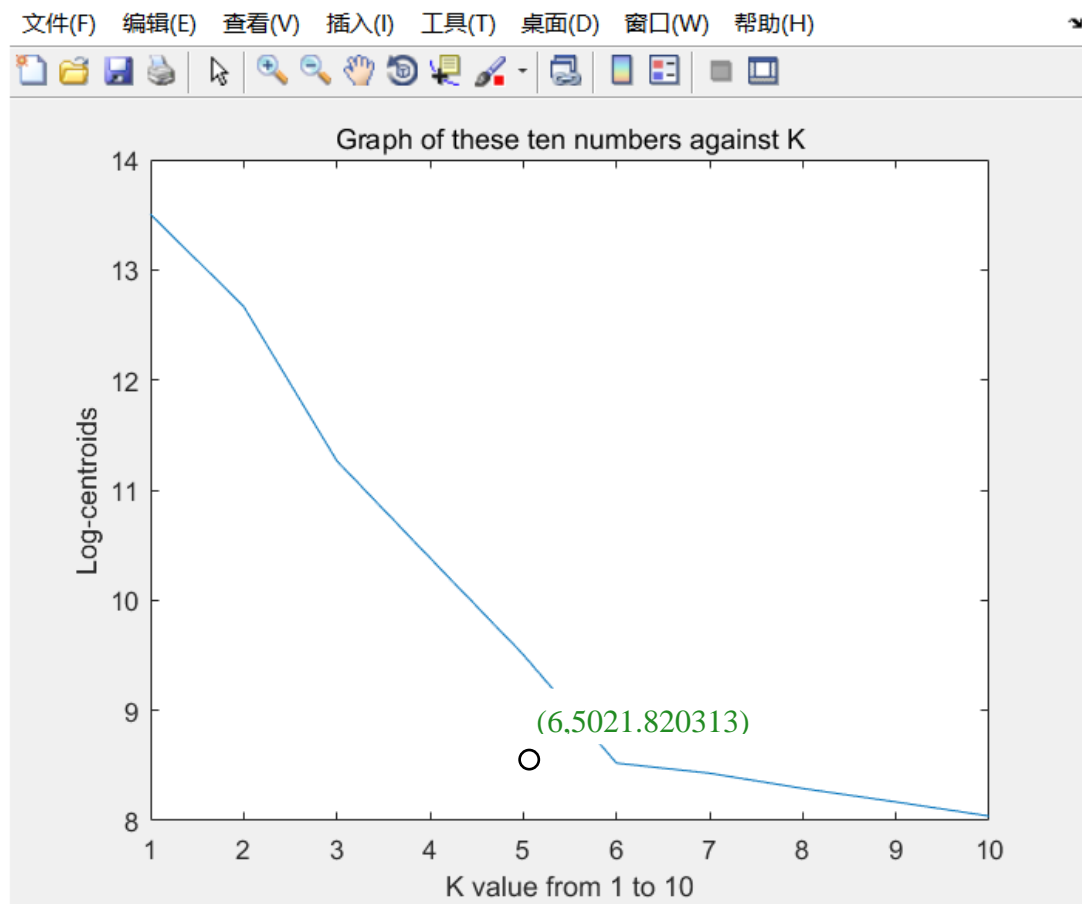
Final number is 3104.663086

```

1  x=1:1:10;
2  a=[732439, 565200, 317257, 187500, 78126, 726563, 32335, 769531, 13455, 114258, 5021, 820313, 4581, 746094, 3987, 835449, 3528];
3  A=log(a);
4  plot(x,A);
5  title('graph of these ten numbers against K')
6  xlabel('K value from 1 to 10')
7  ylabel('log-centroids')

```

*Figure 1.1 Code of Graph of numbers against K*



*Figure 1.2 Graph of these ten numbers against K*

### Conclusion to Part 1:

Distortion is the sum of the distance between each data point and its nearest centroid. The task of clustering is to find a set of centroids that minimize distortion. On the screen, the list of 10 numbers



are the distortion after each iteration. In the above figure, the “elbow” point is (6,5021.820313). The distortion starts to level off when  $k=6$ .

The Elbow method is a method of interpretation and validation of consistency within cluster analysis designed to help finding the appropriate number of clusters in a dataset. According to the elbow method, we can get the best result when the number of clusters  $k$  is 6.

## Part2: Principle Components Analysis (PCA)

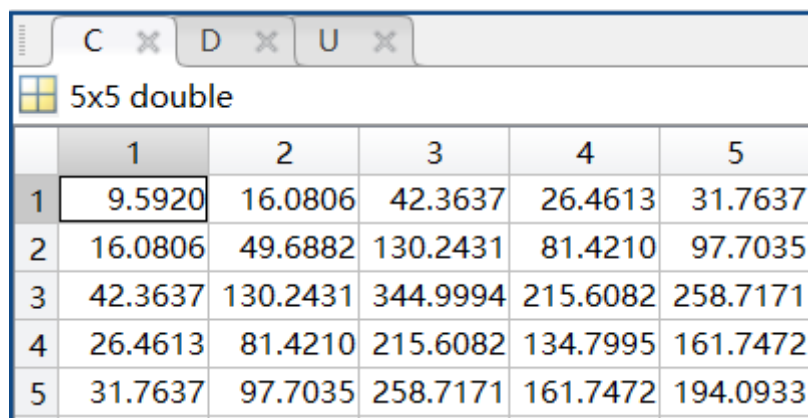
**Step 1:** Load the data (lab2Data-matlab) into matrix X in MATLAB.

**Step 2:** Compute the covariance matrix of the data by using the MATLAB 'cov' function:

```
>> C = cov(X)
```

**Step 3:** Apply eigenvector/eigenvalue decomposition to the covariance matrix:

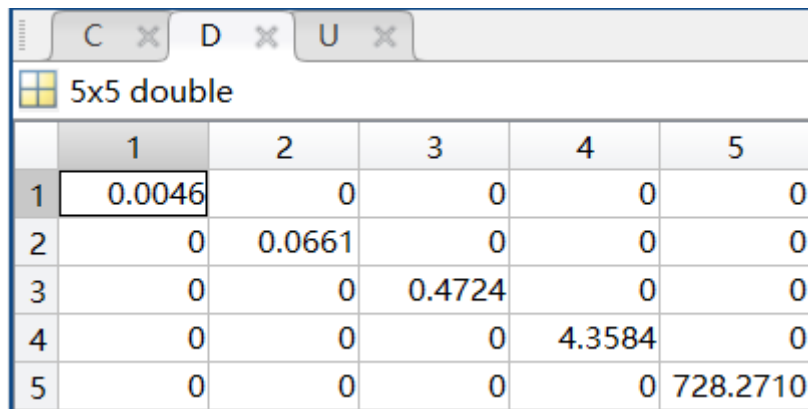
```
>> [U,D] = eig(C)
```



A screenshot of a MATLAB window titled 'C' showing a 5x5 double matrix. The matrix is displayed in a grid with columns labeled 1 to 5 and rows labeled 1 to 5. The values are symmetric across the diagonal.

	1	2	3	4	5
1	9.5920	16.0806	42.3637	26.4613	31.7637
2	16.0806	49.6882	130.2431	81.4210	97.7035
3	42.3637	130.2431	344.9994	215.6082	258.7171
4	26.4613	81.4210	215.6082	134.7995	161.7472
5	31.7637	97.7035	258.7171	161.7472	194.0933

Figure 2.1 Graph of covariance matrix C



A screenshot of a MATLAB window titled 'D' showing a 5x5 double matrix. The matrix is diagonal, with eigenvalues on the diagonal and zeros elsewhere. The values are 0.0046, 0.0661, 0.4724, 4.3584, and 728.2710.

	1	2	3	4	5
1	0.0046	0	0	0	0
2	0	0.0661	0	0	0
3	0	0	0.4724	0	0
4	0	0	0	4.3584	0
5	0	0	0	0	728.2710

Figure 2.2 Graph of eigenvalues matrix D

	1	2	3	4	5
1	0.0017	0.0040	0.0191	-0.9962	0.0850
2	0.0027	-0.0292	-0.9651	0.0036	0.2600
3	-0.0111	-0.6928	0.2066	0.0599	0.6882
4	0.7746	0.4497	0.1046	0.0419	0.4302
5	-0.6324	0.5629	0.1205	0.0476	0.5162

Figure 2.3 Graph of eigenvalues matrix  $U$

$U$  is a unitary matrix. Each column of  $U$  is a principal vector. The corresponding eigenvalues indicate the variance of the data along that dimension. Among them, Large eigenvalues indicate significant components of the data. Small eigenvalues indicate that the variation along the corresponding eigenvectors, such as noise.  $D$  is a diagonal matrix and all elements of  $D$  will be real and non-negative. PCA (Principal Component Analysis), the principal component analysis method, is one of the most widely used data dimensionality reduction algorithms. The main idea of PCA is to map  $n$ -dimensional features to  $k$ -dimension. This  $k$ -dimensional is a new orthogonal feature, also called principal component, which is a  $k$ -dimensional feature reconstructed based on the original  $n$ -dimensional features. In this lab,  $k$  is 5.

### Conclusion to Part 2:

The sum of the squared distances between the projected point and the origin for Principal Component (PC)1 is also its eigenvalue, and the value is 0.0046. The sum of the squared distances for PC2 is also its eigenvalue, and the value is 0.0661. The sum of the squared distances for PC3 is also its eigenvalue, and the value is 0.4724. The sum of the squared distances for PC4 is also its eigenvalue, and the value is 4.3584. The sum of the squared distances for PC5 is also its eigenvalue, and the value is 728.2710.

The sum of squared distances of SS (distance for PC1)/5000-1 is also its variation, and the value is 0.00000092. The sum of squared distances of SS (distance for PC2)/5000-1 is also its variation, and the value is 0.00001322. The sum of squared distances of SS (distance for PC3)/5000-1 is also its variation, and the value is 0.0000945. The sum of squared distances of SS (distance for PC4)/5000-1 is also its variation, and the value is 0.00087185. The sum of squared distances of SS (distance for

PC5)/5000-1 is also its variation, and the value is 0.14568334.

The total variation of all the PCs is  $0.00000092 + 0.00001322 + 0.0000945 + 0.00087185 + 0.14568334 = 0.14666383$ . According to that, PC1 accounts for  $0.00000092/0.14666383 = 0.0006272848595\%$  of the total variation around the PCs. PC2 accounts for  $0.00001322/0.14666383 = 0.009013810699\%$ . PC3 accounts for  $0.0000945/0.14666383 = 0.06443306438\%$ . PC4 accounts for  $0.00087185 / 0.14666383 = 0.5944546791\%$ . PC5 accounts for  $0.14568334/0.14666383 = 99.33147116\%$ .

### **Finally: The summary**

K-means represents all  $n$  data vectors by a small number of cluster centroids, and they are represented as a linear combination of a few cluster centroid vectors, where the linear combination weights must all be zero, except for the single 1. The role is to minimize the mean square reconstruction error. In contrast, the essence of PCA is dimension reduction processing, which represents  $n$  data vectors as a linear combination of a small number of feature vectors, and can also be used to minimize mean square reconstruction errors. PCA is used for dimensionality reduction / feature selection / representation learning e.g. when the feature space contains too many irrelevant or redundant features. The aim is to find the intrinsic dimensionality of the data. K-means and PCA are usually thought of as two very different problems: one as an algorithm for data clustering, and the other as a framework for data dimension reduction. They are, however, intimately related when you look at both through their parent problem of matrix factorization. K-means is a sparse version of PCA. In other words, k-means can be seen as a super-sparse PCA.