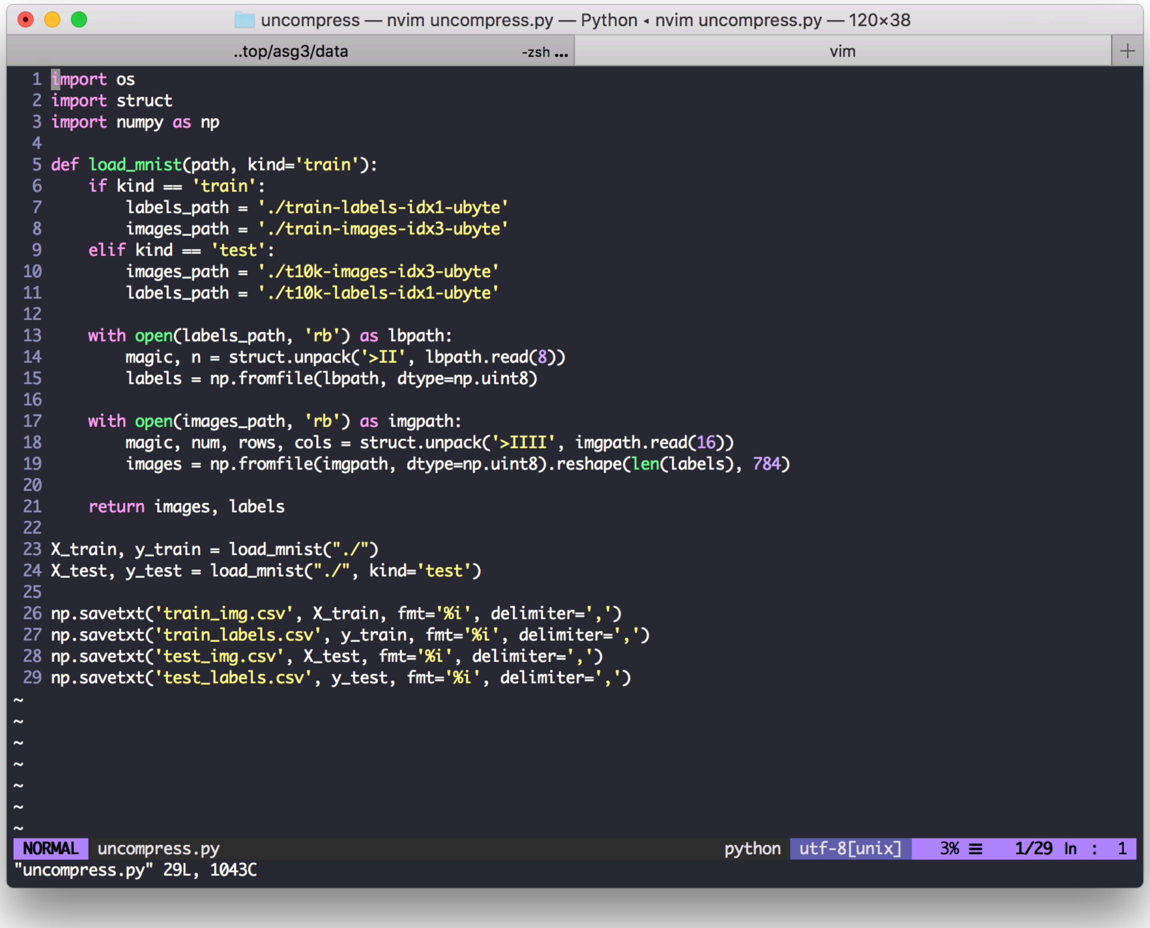
I declare that the assignment submitted on Elearning system is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/.

Signed (Student Cao Yuhang ) Date: 04/07/2018
Name Cao Yuhang SID 1155092180

# Uncompress:

Convert original data to “csv” format. Each row is a 784 dimension data, each column is a pixel feature.



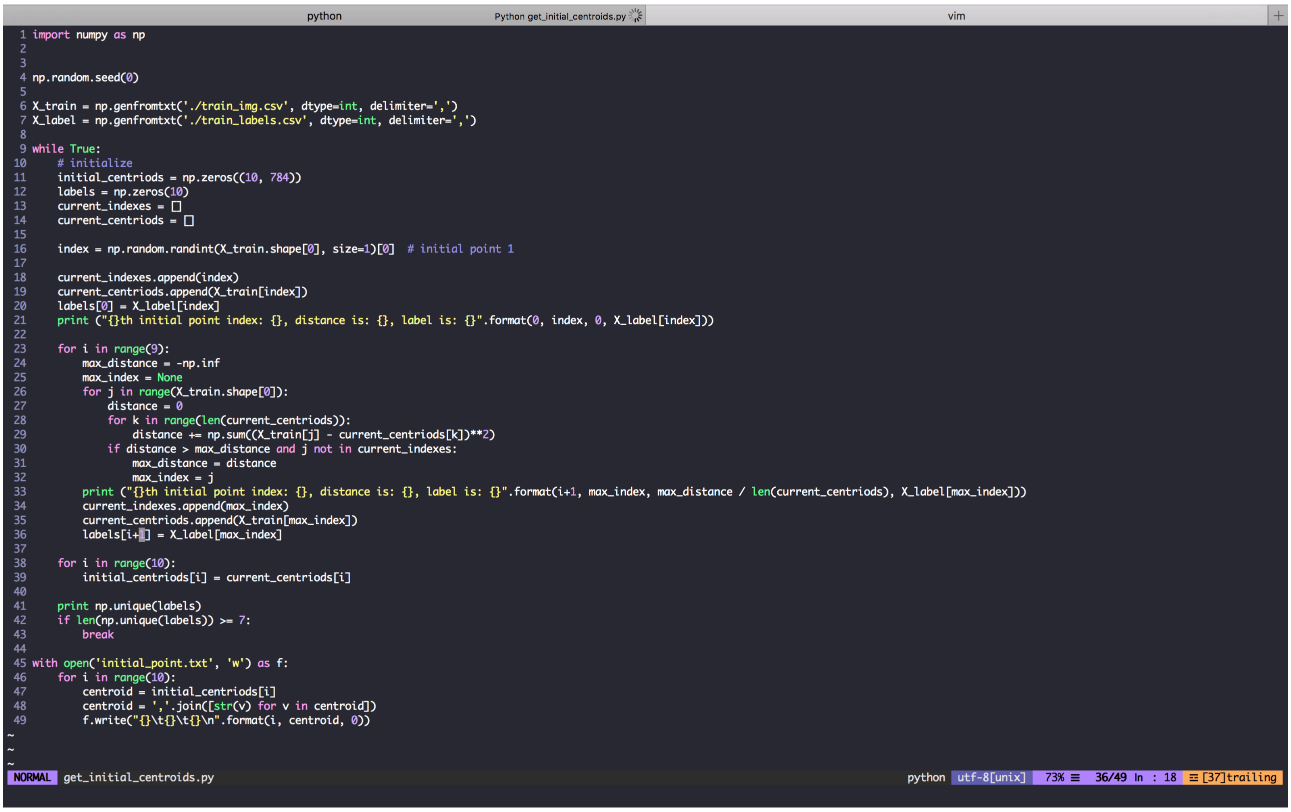
```python
import os
import struct
import numpy as np

def load_mnist(path, kind='train'):
    if kind == 'train':
        labels_path = './train-labels-idx1-ubyte'
        images_path = './train-images-idx3-ubyte'
    elif kind == 'test':
        images_path = './t10k-images-idx3-ubyte'
        labels_path = './t10k-labels-idx1-ubyte'

    with open(labels_path, 'rb') as lbpath:
        magic, n = struct.unpack('>II', lbpath.read(8))
        labels = np.fromfile(lbpath, dtype=np.uint8)

    with open(images_path, 'rb') as imgpath:
        magic, num, rows, cols = struct.unpack('>IIII', imgpath.read(16))
        images = np.fromfile(imgpath, dtype=np.uint8).reshape(len(labels), 784)

    return images, labels

X_train, y_train = load_mnist("./")
X_test, y_test = load_mnist("./", kind='test')

np.savetxt('train_img.csv', X_train, fmt='%i', delimiter=',')
np.savetxt('train_labels.csv', y_train, fmt='%i', delimiter=',')
np.savetxt('test_img.csv', X_test, fmt='%i', delimiter=',')
np.savetxt('test_labels.csv', y_test, fmt='%i', delimiter=',')
```

```
"uncompress.py" 29L, 1043C
```

# Get Initial Centroids:

1. Choose first centroid randomly.
2. Select next initial centroids in such a way that the Euclidean distance of that point is maximum from other selected initial centroids.
3. Repeat step 2 until we get k initial centroids.
4. The program will ensure initial points contain at least 7 different labels.

The result is stored in “cluster_id \t centroid \t cluster_number” format, cluster_number represents how many images in this cluster, at beginning, this is just a dump number, just for the consistent in later processing.



```python
import numpy as np


np.random.seed(0)

X_train = np.genfromtxt('./train_img.csv', dtype=int, delimiter=',')
X_label = np.genfromtxt('./train_labels.csv', dtype=int, delimiter=',')

while True:
    # initialize
    initial_centriods = np.zeros((10, 784))
    labels = np.zeros(10)
    current_indexes = []
    current_centriods = []

    index = np.random.randint(X_train.shape[0], size=1)[0]  # initial point 1

    current_indexes.append(index)
    current_centriods.append(X_train[index])
    labels[0] = X_label[index]
    print ("{}th initial point index: {}, distance is: {}, label is: {}".format(0, index, 0, X_label[index]))

    for i in range(9):
        max_distance = -np.inf
        max_index = None
        for j in range(X_train.shape[0]):
            distance = 0
            for k in range(len(current_centriods)):
                distance += np.sum((X_train[j] - current_centriods[k])**2)
            if distance > max_distance and j not in current_indexes:
                max_distance = distance
                max_index = j
        print ("{}th initial point index: {}, distance is: {}, label is: {}".format(i+1, max_index, max_distance / len(current_centriods), X_label[max_index]))
        current_indexes.append(max_index)
        current_centriods.append(X_train[max_index])
        labels[i+1] = X_label[max_index]

    for i in range(10):
        initial_centriods[i] = current_centriods[i]

    print np.unique(labels)
    if len(np.unique(labels)) >= 7:
        break

with open('initial_point.txt', 'w') as f:
    for i in range(10):
        centroid = initial_centriods[i]
        centroid = ','.join([str(v) for v in centroid])
        f.write("{}\t{}\t{}\n".format(i, centroid, 0))
~
~
~
```

# Q1 (a):

Mapper:
input: split of data
output: “cluster_id \t partial sum of this cluster \t image number of this cluster”

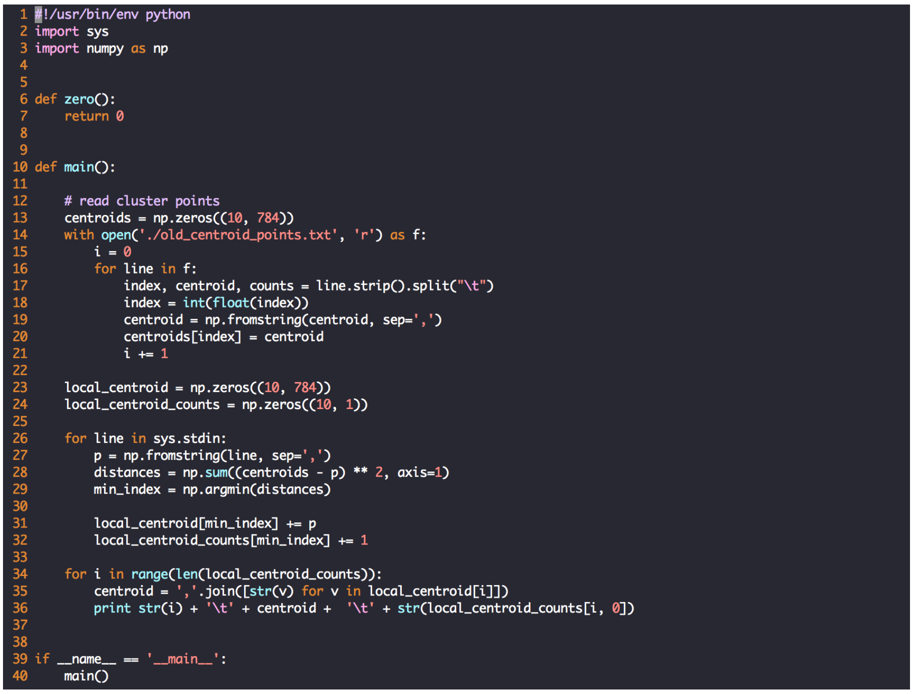
Reducer:
Merge partial sum and image number of same cluster_id, divided sum by number to get mean of this cluster_id

Code and Result:

- mapper1. py: mapper
- reducer1.py: reducer
- format_output.py: get desire output format
- cal_error.py: calculate error between consecutive iterations
- run.sh: run the program in an iterative manner
- error.txt: track the error changing
- ori_centroid_points.txt, old_centroid_points.txt, new_centroid_points.txt: tmp files
- res_a: result of part_a

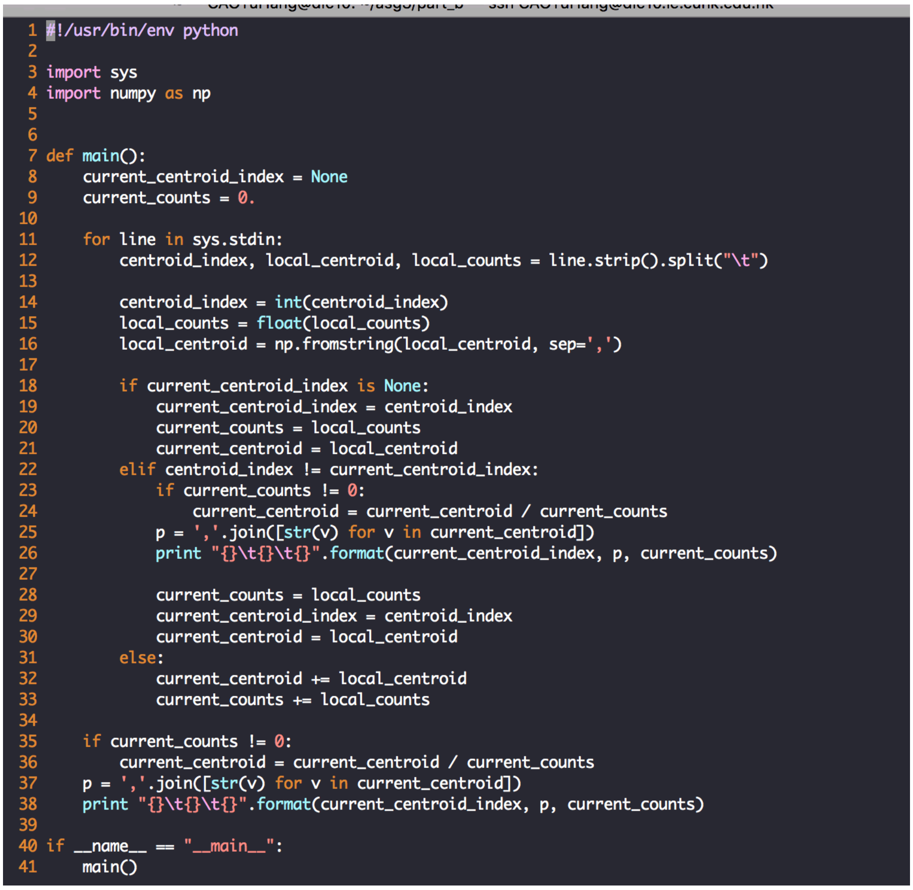
Code:

Mapper:



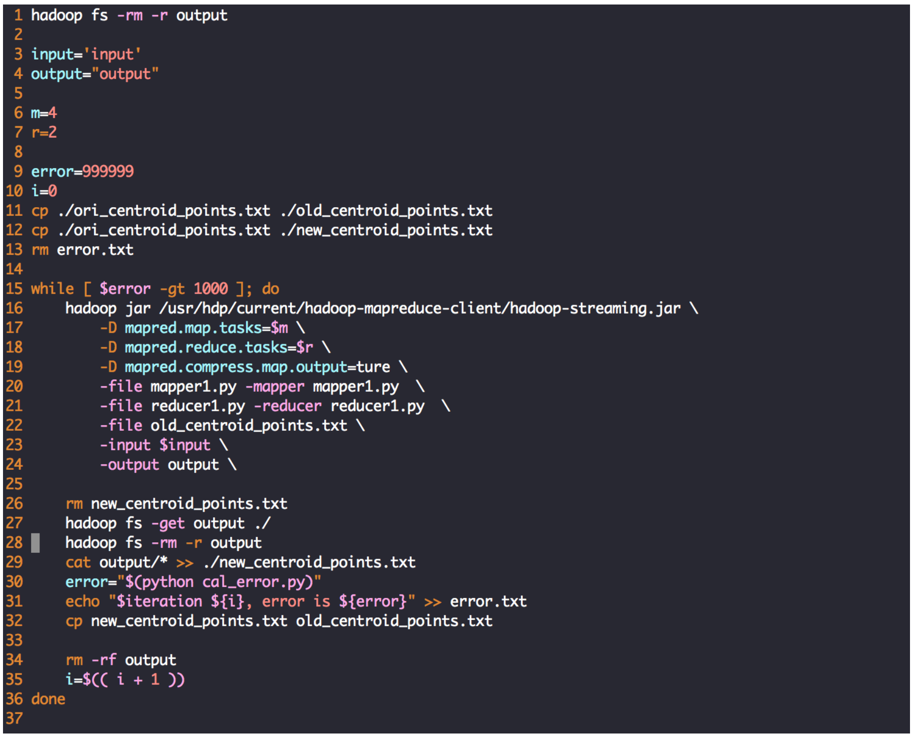
```python
#!/usr/bin/env python
import sys
import numpy as np


def zero():
    return 0


def main():

    # read cluster points
    centroids = np.zeros((10, 784))
    with open('./old_centroid_points.txt', 'r') as f:
        i = 0
        for line in f:
            index, centroid, counts = line.strip().split("\t")
            index = int(float(index))
            centroid = np.fromstring(centroid, sep=',')
            centroids[index] = centroid
            i += 1

    local_centroid = np.zeros((10, 784))
    local_centroid_counts = np.zeros((10, 1))

    for line in sys.stdin:
        p = np.fromstring(line, sep=',')
        distances = np.sum((centroids - p) ** 2, axis=1)
        min_index = np.argmin(distances)

        local_centroid[min_index] += p
        local_centroid_counts[min_index] += 1

    for i in range(len(local_centroid_counts)):
        centroid = ','.join([str(v) for v in local_centroid[i]])
        print str(i) + '\t' + centroid +  '\t' + str(local_centroid_counts[i, 0])


if __name__ == '__main__':
    main()
```

Reducer:



```python
#!/usr/bin/env python

import sys
import numpy as np


def main():
    current_centroid_index = None
    current_counts = 0.

    for line in sys.stdin:
        centroid_index, local_centroid, local_counts = line.strip().split("\t")

        centroid_index = int(centroid_index)
        local_counts = float(local_counts)
        local_centroid = np.fromstring(local_centroid, sep=',')

        if current_centroid_index is None:
            current_centroid_index = centroid_index
            current_counts = local_counts
            current_centroid = local_centroid
        elif centroid_index != current_centroid_index:
            if current_counts != 0:
                current_centroid = current_centroid / current_counts
            p = ','.join([str(v) for v in current_centroid])
            print "{}\t{}\t{}".format(current_centroid_index, p, current_counts)

            current_counts = local_counts
            current_centroid_index = centroid_index
            current_centroid = local_centroid
        else:
            current_centroid += local_centroid
            current_counts += local_counts

    if current_counts != 0:
        current_centroid = current_centroid / current_counts
    p = ','.join([str(v) for v in current_centroid])
    print "{}\t{}\t{}".format(current_centroid_index, p, current_counts)

if __name__ == "__main__":
    main()
```

run.sh:



```
hadoop fs -rm -r output

input='input'
output="output"

m=4
r=2

error=999999
i=0
cp ./ori_centroid_points.txt ./old_centroid_points.txt
cp ./ori_centroid_points.txt ./new_centroid_points.txt
rm error.txt

while [ $error -gt 1000 ]; do
    hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar \
        -D mapred.map.tasks=$m \
        -D mapred.reduce.tasks=$r \
        -D mapred.compress.map.output=ture \
        -file mapper1.py -mapper mapper1.py  \
        -file reducer1.py -reducer reducer1.py  \
        -file old_centroid_points.txt \
        -input $input \
        -output output \

    rm new_centroid_points.txt
    hadoop fs -get output ./
    hadoop fs -rm -r output
    cat output/* >> ./new_centroid_points.txt
    error="$(python cal_error.py)"
    echo "$iteration ${i}, error is ${error}" >> error.txt
    cp new_centroid_points.txt old_centroid_points.txt

    rm -rf output
    i=$(( i + 1 ))
done

```

Result:

Centroid 0:

[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.015,0.02,0.022,0.023,0.024,0.012,0.047,0.096,0.156,0.045,0.001,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.006,0.073,0.167,0.296,0.303,0.334,0.452,0.517,0.48,0.503,0.338,0.2,0.146,0.104,0.041,0.007,0.026,0.012,0.0,0.0,0.0,0.0,0.0,0.004,0.001,0.011,0.035,0.035,0.218,0.654,1.375,2.081,2.613,2.946,3.555,3.694,3.461,2.859,2.602,2.516,1.971,1.167,0.718,0.395,0.2,0.093,0.011,0.012,0.0,0.0,0.0,0.015,0.009,0.066,0.117,0.34,1.069,2.242,4.131,6.768,10.634,17.906,27.925,36.744,40.644,39.351,36.134,29.892,23.333,15.524,8.293,3.666,1.511,0.59,0.117,0.031,0.012,0.0,0.0,0.01,0.025,0.08,0.464,1.177,2.526,4.417,6.702,10.926,18.681,33.345,53.565,71.533,81.112,80.723,72.96,59.566,43.459,28.303,15.05,6.92,3.161,1.309,0.419,0.076,0.01,0.0,0.0,0.025,0.058,0.108,0.733,2.043,4.049,6.496,9.254,14.22,23.555,41.133,66.433,91.607,106.937,107.129,95.066,75.394,51.407,31.49,16.636,8.453,4.606,2.47,0.978,0.183,0.024,0.0,0.0,0.04,0.031,0.188,1.177,2.91,5.074,7.658,10.397,15.547,24.817,42.767,71.697,102.611,123.503,124.323,107.632,79.708,50.943,28.649,15.614,9.024,6.221,4.37,1.939,0.289,0.038,0.0,0.0,0.02,0.024,0.304,1.505,3.307,5.194,7.184,9.655,13.653,21.939,39.631,72.587,111.316,138.326,137.28,112.695,77.528,44.767,23.554,14.033,9.843,8.213,6.015,2.992,0.458,0.019,0.0,0.008,0.012,0.032,0.319,1.443,2.989,4.009,5.412,7.074,10.358,17.648,36.702,75.888,124.421,156.189,148.242,111.548,68.703,36.124,18.873,12.185,9.965,8.335,6.018,2.792,0.486,0.058,0.0,0.014,0.006,0.026,0.261,1.072,2.154,2.647,3.415,5.181,8.55,16.3,38.507,85.77,144.835,176.351,152.387,100.252,51.773,23.499,12.036,8.465,6.853,5.541,3.783,1.698,0.382,0.028,0.0,0.006,0.004,0.028,0.204,0.652,1.334,1.723,2.555,4.858,9.386,19.433,44.198,100.502,169.401,190.869,145.071,77.797,31.001,11.314,5.678,3.912,3.164,2.541,1.619,0.691,0.148,0.011,0.0,0.0,0.007,0.028,0.189,0.509,0.92,1.213,2.368,5.747,12.798,23.985,50.412,120.686,193.496,196.861,128.169,51.532,14.394,4.679,2.461,1.871,1.556,1.155,0.697,0.285,0.107,0.021,0.003,0.0,0.002,0.02,0.191,0.439,0.742,1.324,3.042,7.782,16.172,27.795,60.785,149.74,211.614,195.142,103.614,29.73,7.137,3.178,2.095,1.704,1.24

```
1,1.086,0.828,0.355,0.151,0.025,0.0,0.0,0.001,0.026,0.108,0.525,0.936,1.713,4.086,9.42,17.445,3
1.892,78.796,176.255,220.014,183.25,76.337,19.056,6.584,3.823,2.505,2.033,1.519,1.286,0.815,0
.343,0.134,0.0,0.0,0.0,0.01,0.047,0.133,0.891,1.549,2.477,5.041,10.001,18.778,40.874,102.565,19
0.523,218.151,160.129,57.204,16.531,7.433,4.353,3.16,2.645,2.043,1.533,0.984,0.424,0.199,0.02
6,0.0,0.004,0.015,0.052,0.289,1.376,2.866,4.071,6.253,11.065,24.096,57.787,123.895,192.8,203.2
4,132.09,49.523,18.363,8.74,5.067,3.795,3.104,2.417,1.877,1.386,0.862,0.297,0.013,0.0,0.013,0.0
04,0.046,0.424,2.395,5.226,6.651,8.672,15.82,35.871,77.725,136.007,185.538,177.294,109.008,4
6.66,20.179,9.735,5.699,4.422,3.781,3.18,2.79,2.156,1.229,0.331,0.013,0.0,0.002,0.005,0.04,0.54
5,3.336,7.663,10.03,13.117,25.459,52.884,94.828,140.645,171.911,152.434,95.562,45.98,20.655,
10.059,6.388,5.49,4.734,4.115,3.445,2.411,1.043,0.211,0.029,0.002,0.008,0.028,0.106,0.652,3.65
3,9.154,13.998,21.764,40.015,70.79,107.878,140.25,155.038,133.023,87.28,45.754,20.973,10.756
,7.073,5.792,4.851,3.896,3.067,2.028,0.755,0.108,0.016,0.0,0.0,0.014,0.112,0.652,3.334,8.994,16.
892,30.669,54.756,84.699,113.797,133.733,139.221,119.334,82.35,46.611,22.735,12.615,8.279,6.
282,4.748,3.46,2.388,1.64,0.587,0.113,0.005,0.003,0.001,0.029,0.079,0.586,2.843,8.463,18.535,3
6.381,61.622,88.599,111.177,122.922,124.05,107.38,77.662,46.517,23.275,12.624,8.09,5.697,4.0
36,2.604,1.674,0.977,0.345,0.101,0.0,0.003,0.001,0.023,0.048,0.512,2.616,7.84,18.617,37.307,60.
411,84.317,100.159,107.997,106.635,93.98,68.803,41.603,20.753,10.181,5.798,3.919,2.702,1.607
,0.912,0.361,0.12,0.026,0.0,0.0,0.0,0.0,0.054,0.591,2.269,6.464,15.178,29.707,47.722,64.133,74.2
78,77.983,77.182,68.904,50.679,29.692,13.566,5.657,2.977,1.957,1.245,0.821,0.482,0.177,0.046,
0.03,0.0,0.0,0.0,0.0,0.041,0.31,0.985,2.392,5.116,9.221,15.115,20.73,24.14,26.226,27.186,23.89,1
6.781,9.343,3.926,1.604,0.813,0.477,0.367,0.28,0.141,0.047,0.014,0.002,0.0,0.0,0.0,0.0,0.005,0.1
28,0.194,0.352,0.853,1.473,2.292,3.185,3.897,4.764,5.113,3.585,2.007,0.926,0.367,0.173,0.124,0.
109,0.082,0.057,0.024,0.004,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.003,0.037,0.069,0.082,0.104,0.182,0.
405,0.664,0.827,0.467,0.172,0.095,0.036,0.046,0.028,0.026,0.028,0.008,0.003,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.012,0.026,0.036,0.009,0.0,0.0,0.006,0.02,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0], 9781.0
Centroid 1:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.016,0.046,0.04,0.047,0.076,0.079,0.064,0.083
,0.253,0.438,0.435,0.365,0.235,0.173,0.052,0.04,0.005,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.002,0.014,0.00
3,0.066,0.204,0.576,1.508,3.451,6.886,13.326,20.995,28.784,33.922,34.204,30.496,24.046,16.898
,11.597,7.685,4.664,2.053,0.645,0.14,0.0,0.0,0.0,0.0,0.023,0.049,0.117,0.217,0.601,1.507,3.839,1
0.412,24.204,48.605,80.335,109.989,133.143,145.179,144.975,131.627,108.802,80.434,54.781,34
.894,20.901,10.145,3.138,0.628,0.01,0.0,0.001,0.029,0.051,0.104,0.189,0.668,1.788,5.638,15.387,
37.455,75.377,122.137,164.269,190.912,202.276,205.166,203.234,192.903,171.185,136.25,96.73
4,61.801,36.509,18.783,6.587,1.155,0.043,0.0,0.055,0.015,0.027,0.176,0.41,1.462,5.056,14.868,3
6.732,76.956,130.808,176.326,199.041,197.628,187.306,177.897,177.312,181.271,176.688,152.3
63,111.519,71.331,41.708,20.577,7.016,1.18,0.098,0.0,0.031,0.0,0.11,0.251,0.765,2.916,9.489,25.
724,59.431,110.091,159.274,182.479,170.228,140.423,116.505,107.542,117.084,140.632,155.178
,140.61,104.088,65.311,35.473,16.428,5.65,1.059,0.093,0.0,0.033,0.035,0.19,0.469,1.169,4.252,1
2.676,34.179,73.778,123.692,157.019,152.519,115.589,78.35,59.55,65.302,91.541,124.989,140.4
45,123.091,86.244,51.053,26.176,11.687,4.129,0.741,0.075,0.0,0.017,0.04,0.197,0.501,1.386,4.66
3,15.067,38.646,78.664,124.409,145.138,124.909,83.04,52.548,48.908,70.155,106.214,135.002,1
32.974,102.73,64.459,34.742,16.493,7.326,2.602,0.412,0.041,0.0,0.012,0.031,0.142,0.431,1.291,4
.345,14.563,38.426,79.145,121.659,140.097,123.331,89.038,69.548,78.644,107.957,136.38,140.7
98,114.766,75.567,40.716,19.444,8.303,3.388,1.055,0.199,0.005,0.0,0.002,0.004,0.092,0.238,0.98
6,3.524,13.056,36.425,75.402,119.1,145.876,146.814,129.27,118.93,131.896,151.517,151.664,12
3.528,82.052,45.721,20.928,8.972,3.715,1.461,0.436,0.131,0.031,0.0,0.0,0.0,0.042,0.154,0.777,2.
99,11.477,32.292,69.048,112.045,152.406,176.064,178.987,174.851,179.74,174.091,141.802,93.5
6,50.831,23.389,9.374,3.958,1.748,0.683,0.287,0.097,0.008,0.0,0.016,0.011,0.012,0.056,0.455,2.5
```

```
29,9.469,26.722,58.442,99.301,146.431,184.091,201.083,202.279,195.286,167.805,119.584,70.78
3,34.97,14.489,5.022,1.813,0.743,0.352,0.132,0.016,0.0,0.0,0.036,0.041,0.072,0.145,0.603,2.25,7.
822,21.406,47.785,86.035,131.22,170.485,187.159,188.147,177.147,149.517,110.046,69.582,34.9
59,13.057,3.612,0.805,0.244,0.12,0.041,0.062,0.0,0.0,0.017,0.168,0.407,0.72,1.442,3.227,8.893,2
2.679,47.453,83.287,121.633,145.807,150.337,148.697,147.085,139.361,118.827,81.564,40.88,13
.035,2.779,0.65,0.162,0.117,0.087,0.03,0.0,0.0,0.017,0.248,1.032,2.217,4.051,7.195,15.857,33.61
9,62.058,92.833,114.202,115.04,103.865,106.522,124.011,140.467,135.025,95.363,44.699,12.524
,2.238,0.695,0.272,0.152,0.077,0.024,0.0,0.0,0.003,0.594,2.462,5.539,9.989,17.075,30.521,54.142
,82.813,102.152,100.788,79.745,64.596,78.31,116.472,151.717,149.943,103.035,44.637,10.963,2.
147,0.793,0.404,0.192,0.111,0.027,0.0,0.0,0.0,1.146,4.298,9.754,19.028,32.355,53.38,79.129,100.
828,102.378,81.329,55.479,49.461,78.4,129.724,167.2,155.594,99.51,38.741,9.496,2.091,0.781,0.
386,0.163,0.039,0.024,0.0,0.0,0.029,1.319,5.772,13.116,27.193,49.767,79.276,106.305,118.718,1
08.956,82.784,66.725,76.052,115.556,161.673,177.769,145.361,82.405,29.337,7.772,1.93,0.807,0
.308,0.104,0.054,0.039,0.008,0.0,0.021,0.98,5.212,13.291,30.734,60.985,98.906,131.327,145.47,1
42.445,128.627,125.155,141.599,169.831,185.614,167.307,114.22,56.423,19.463,5.595,1.62,0.60
5,0.257,0.124,0.041,0.027,0.0,0.0,0.001,0.708,3.396,10.457,25.793,56.578,98.691,140.712,171.82
2,188.06,193.782,197.418,200.398,197.034,173.339,125.231,71.68,31.731,10.877,3.538,1.306,0.6
3,0.208,0.114,0.042,0.001,0.0,0.0,0.0,0.311,1.607,5.828,15.718,37.113,71.512,113.635,154.8,186.
533,203.929,204.694,188.51,155.536,112.133,67.839,34.159,14.217,4.796,1.635,0.833,0.364,0.10
9,0.043,0.012,0.0,0.0,0.0,0.0,0.105,0.547,2.079,5.443,13.448,28.227,50.216,75.613,97.626,109.75
2,108.094,92.459,67.947,41.719,21.763,9.923,3.676,1.222,0.439,0.265,0.232,0.105,0.003,0.0,0.0,
0.0,0.0,0.0,0.009,0.03,0.106,0.348,1.129,2.248,4.099,6.104,7.492,8.207,7.676,6.686,5.093,3.18,1.
466,0.528,0.199,0.064,0.008,0.0,0.011,0.038,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 6362.0
Centroid 2:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.004,0.045,0.015,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.009,0.022,0.027,0.023,0.025,0.046,0.068,
0.039,0.114,0.099,0.086,0.097,0.07,0.018,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.014,0.028,0.037,
0.018,0.043,0.087,0.216,0.37,0.505,0.963,1.02,0.979,0.861,1.054,1.39,1.897,2.095,1.71,0.734,0.3
46,0.114,0.038,0.007,0.0,0.0,0.0,0.0,0.0,0.022,0.065,0.159,0.418,0.862,1.251,2.523,3.867,6.279,8.
272,8.749,8.498,7.656,8.026,9.178,10.803,10.219,7.24,4.181,2.018,0.832,0.305,0.115,0.0,0.0,0.0,
0.0,0.022,0.092,0.769,1.925,3.706,5.646,8.96,13.963,21.261,31.49,42.325,51.727,58.423,61.171,5
8.111,50.02,39.339,27.706,16.874,9.381,4.955,2.556,1.468,0.765,0.315,0.052,0.007,0.054,0.347,1
.37,4.324,8.935,15.412,23.029,32.971,47.147,68.497,95.281,116.631,127.718,131.752,132.917,13
1.048,120.73,95.752,63.849,34.179,16.409,8.25,4.63,2.651,1.497,0.449,0.059,0.027,0.095,1.171,4
.142,9.714,19.034,30.403,44.91,64.028,93.047,131.548,163.99,174.957,166.339,155.4,153.358,16
1.306,164.657,141.878,97.318,53.271,24.755,10.826,5.064,2.518,1.178,0.307,0.037,0.004,0.101,1
.889,5.985,12.821,24.006,38.405,57.645,85.799,127.846,168.398,182.534,163.351,133.542,116.1
73,121.121,146.644,170.471,159.374,113.731,62.129,28.413,11.346,4.141,1.815,0.85,0.244,0.041
,0.0,0.116,1.889,5.599,11.799,21.26,35.215,56.654,93.33,139.196,164.804,147.342,105.419,73.13
9,63.108,80.105,124.753,166.822,161.811,116.01,61.537,26.452,9.635,2.611,0.817,0.342,0.073,0.
015,0.021,0.047,1.404,3.83,8.107,14.584,26.836,51.554,94.972,136.835,140.554,99.887,54.372,3
2.86,34.257,62.803,122.466,169.942,160.873,109.226,55.021,22.026,7.658,1.695,0.531,0.178,0.0
86,0.016,0.023,0.037,0.603,1.974,4.532,9.371,20.938,49.492,95.84,130.847,117.449,71.104,34.66
2,26.067,37.376,76.033,141.59,181.1,157.046,97.308,45.091,17.539,5.58,1.262,0.371,0.061,0.003
,0.0,0.0,0.025,0.207,0.932,2.7,6.946,18.663,48.173,93.136,122.417,106.516,69.475,48.579,47.771
,63.944,106.13,167.923,191.725,148.632,84.43,36.408,14.25,4.967,1.801,0.674,0.079,0.0,0.0,0.0,
0.003,0.066,0.434,1.826,5.901,17.485,44.56,84.608,111.126,107.262,89.175,81.556,83.209,95.87
3,135.717,190.102,196.875,138.579,71.816,30.649,12.842,5.872,2.575,0.956,0.295,0.0,0.0,0.001,
0.0,0.028,0.285,1.398,5.334,15.363,35.609,67.982,94.433,101.793,100.792,99.885,100.418,109.9
04,153.175,204.691,193.969,121.293,57.08,23.821,11.313,5.728,2.665,1.163,0.372,0.029,0.0,0.00
6,0.0,0.0,0.163,1.075,4.139,10.852,23.352,44.161,65.415,77.439,82.453,83.644,84.25,103.234,16
6.268,215.445,180.18,94.685,37.803,15.141,8.073,4.508,2.377,0.914,0.184,0.005,0.0,0.0,0.009,0.
004,0.057,0.549,2.235,6.041,12.063,21.516,32.553,41.112,45.75,47.954,53.869,96.017,182.694,2
19.468,155.11,63.43,19.949,7.792,4.14,2.367,1.313,0.555,0.073,0.0,0.0,0.0,0.006,0.0,0.079,0.277,
0.922,2.454,5.049,8.171,12.391,16.09,19.062,22.347,37.766,108.327,201.486,210.255,123.324,38
.205,9.115,3.426,1.848,1.141,0.677,0.353,0.051,0.0,0.0,0.0,0.014,0.0,0.05,0.14,0.309,0.751,1.686,
2.637,4.026,5.545,7.438,12.603,44.415,135.474,213.395,187.219,90.205,21.626,4.568,1.818,0.97
4,0.629,0.291,0.111,0.015,0.0,0.0,0.0,0.006,0.011,0.016,0.063,0.118,0.261,0.691,1.192,1.775,2.28
3,4.479,14.756,70.677,162.583,212.846,157.311,63.806,13.491,3.248,1.322,0.79,0.573,0.337,0.08
9,0.007,0.0,0.0,0.0,0.0,0.004,0.0,0.07,0.096,0.223,0.597,0.957,1.598,2.402,6.223,30.301,102.23,1
82.071,198.501,126.505,46.738,9.965,2.844,1.237,0.73,0.468,0.309,0.105,0.033,0.0,0.0,0.0,0.0,0.
0,0.0,0.016,0.071,0.298,0.703,1.212,1.833,3.603,13.695,55.598,129.316,188.957,175.259,102.772
,37.847,8.97,2.508,0.998,0.469,0.21,0.14,0.035,0.053,0.009,0.0,0.0,0.0,0.0,0.0,0.009,0.054,0.384,
0.764,1.544,2.918,7.363,27.971,81.041,145.473,179.816,148.958,84.744,32.605,8.136,2.238,0.79
4,0.327,0.147,0.073,0.039,0.053,0.009,0.0,0.0,0.0,0.0,0.008,0.042,0.1,0.304,0.799,1.917,4.431,13.
02,41.494,92.244,140.298,150.608,116.442,66.286,26.361,6.961,2.101,0.782,0.249,0.124,0.059,0.
052,0.035,0.0,0.0,0.0,0.0,0.005,0.0,0.03,0.072,0.205,0.497,1.427,4.092,14.162,40.669,79.333,105.
092,99.268,72.936,41.908,17.186,4.941,1.503,0.548,0.144,0.116,0.034,0.015,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.002,0.009,0.04,0.096,0.335,1.893,7.43,20.84,37.825,44.935,39.923,26.964,15.027,6.415,1.
985,0.48,0.058,0.006,0.025,0.009,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.015,0.096,0.431,1.23
5,2.043,3.504,4.184,4.324,2.988,1.396,0.448,0.121,0.01,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 7163.0
Centroid 3:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.002,0.046,0.046,0.002,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.006,0.056,0.095,0.034,0.073,0.148,0.132,0.15
4,0.087,0.013,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.01,0.098,0.274,0.588,1.174,1.
684,2.554,3.161,3.464,4.082,4.237,3.153,2.122,1.233,0.639,0.207,0.061,0.057,0.061,0.02,0.0,0.0,
0.0,0.0,0.0,0.002,0.004,0.031,0.262,1.026,2.403,5.864,11.099,17.94,26.972,35.643,44.06,49.034,4
9.833,44.9,35.656,24.452,15.581,8.118,3.527,1.192,0.2,0.007,0.0,0.0,0.0,0.0,0.0,0.014,0.03,0.23,1
.401,3.742,9.315,19.355,33.718,52.578,76.299,100.42,120.557,131.941,134.231,126.392,105.116,
78.979,54.176,31.392,15.412,5.646,1.091,0.135,0.012,0.0,0.0,0.0,0.0,0.025,0.038,0.768,3.392,9.7
34,21.155,39.311,62.871,90.81,120.383,145.356,164.691,177.198,181.628,174.729,157.967,126.7
5,91.749,59.315,31.883,13.253,3.19,0.635,0.089,0.0,0.0,0.0,0.0,0.014,0.074,1.504,6.058,15.604,3
2.794,55.38,81.605,108.509,132.717,148.211,158.546,165.052,171.465,173.935,168.838,150.844,
119.166,83.388,48.542,21.437,5.577,0.846,0.096,0.015,0.0,0.0,0.028,0.0,0.119,2.036,7.724,19.30
4,37.355,57.89,79.417,98.675,112.234,117.026,117.01,118.003,124.515,137.555,151.615,152.135
,131.917,97.986,60.788,28.573,7.634,0.884,0.07,0.011,0.0,0.0,0.0,0.0,0.17,2.02,7.852,19.498,33.6
19,48.956,62.172,73.12,77.247,75.969,71.119,69.876,77.919,100.853,131.494,148.001,136.831,1
05.119,65.537,31.431,8.115,0.849,0.051,0.0,0.0,0.0,0.0,0.032,0.186,1.906,6.671,15.626,25.429,34
.221,40.737,45.926,46.361,41.415,36.818,36.529,49.83,80.884,122.173,147.173,140.785,106.787,
64.78,30.081,7.781,0.72,0.006,0.0,0.0,0.0,0.0,0.019,0.133,1.228,4.691,10.636,15.546,20.011,23.4
12,25.689,24.306,20.604,19.414,24.746,44.003,81.559,126.531,151.74,141.909,102.734,58.9,25.1
11,6.388,0.591,0.1,0.052,0.0,0.0,0.0,0.0,0.165,0.988,3.153,5.851,8.123,10.366,12.492,13.389,12.8
76,12.999,17.227,28.424,53.926,94.446,137.351,156.236,138.575,94.083,48.489,18.715,4.512,0.6
45,0.123,0.01,0.0,0.0,0.0,0.0,0.171,0.804,2.047,3.168,4.787,7.014,8.78,10.577,13.046,17.991,28.3
34,45.329,74.195,112.783,149.305,157.46,128.372,79.389,36.886,12.925,3.104,0.859,0.273,0.01,
0.0,0.0,0.0,0.01,0.312,0.94,2.279,4.374,7.126,11.089,15.791,21.852,29.867,40.412,54.126,73.369,
100.118,134.626,158.95,150.818,111.824,63.108,26.6,8.682,3.003,1.643,0.577,0.055,0.0,0.0,0.01
3,0.005,0.503,1.697,4.897,10.386,18.818,29.332,41.456,54.315,67.577,81.134,95.487,112.295,13
5.369,156.169,161.77,138.597,94.553,50.285,20.935,7.808,4.347,2.749,1.058,0.141,0.011,0.0,0.0,
0.145,1.209,4.48,13.097,26.938,44.95,63.634,82.693,100.377,116.103,129.664,142.255,155.898,1
```

```
67.133,169.615,157.937,124.101,81.65,45.243,22.781,12.89,8.789,5.238,2.113,0.313,0.041,0.0,0.
0,0.385,2.688,11.569,29.574,54.583,80.365,103.969,123.542,136.452,148.288,159.872,172.072,1
80.869,183.795,175.101,151.538,117.103,79.898,51.547,33.517,23.429,16.962,10.245,3.946,0.44
1,0.002,0.0,0.0,0.504,5.97,22.608,50.801,84.822,113.294,132.478,142.333,147.264,155.235,166.2
21,178.909,186.732,185.255,172.594,149.944,121.277,92.578,69.282,51.708,38.791,27.711,17.03
7,7.386,0.979,0.0,0.0,0.0,0.582,9.925,34.328,71.752,110.511,137.518,149.751,152.743,155.02,15
9.717,169.867,181.135,185.341,181.623,169.929,152.775,133.752,112.381,92.671,75.2,58.009,40
.896,23.33,8.877,0.949,0.0,0.0,0.0,0.831,12.947,42.233,84.755,127.663,158.608,173.135,175.729,
175.811,177.599,181.775,182.565,177.027,167.203,155.657,145.099,136.259,123.989,108.627,90
.348,69.305,46.907,24.766,8.337,0.972,0.0,0.0,0.011,1.056,12.998,42.746,83.268,130.47,167.359,
190.991,198.021,194.716,188.278,178.583,163.683,146.481,130.192,120.493,116.962,117.42,112
.611,101.659,84.534,62.78,40.552,21.009,5.971,0.738,0.0,0.0,0.055,0.91,9.529,32.369,66.783,109
.285,147.712,173.846,183.449,176.28,159.658,139.13,115.785,94.915,80.509,74.822,76.176,80.0
39,80.87,74.463,61.029,43.978,27.903,13.621,3.406,0.288,0.0,0.0,0.0,0.475,4.75,16.546,38.188,6
6.816,95.28,115.031,121.101,113.76,96.792,76.261,56.556,42.76,35.076,34.333,37.233,42.038,44
.275,41.18,33.595,24.13,14.663,6.367,1.224,0.08,0.0,0.0,0.0,0.055,1.055,4.663,12.326,22.119,32.
715,39.196,39.54,34.969,26.999,18.885,12.709,9.223,7.958,9.037,10.705,12.445,13.744,13.542,1
1.65,8.07,4.224,1.425,0.189,0.018,0.0,0.0,0.0,0.023,0.159,0.407,1.363,2.316,3.116,3.009,2.419,1.
573,1.03,0.747,0.48,0.438,0.615,0.874,0.871,0.993,1.332,1.675,1.511,1.152,0.414,0.103,0.0,0.0,0.
0,0.0,0.0,0.0,0.012,0.026,0.082,0.154,0.191,0.272,0.196,0.158,0.128,0.051,0.011,0.0,0.0,0.055,0.0
55,0.028,0.033,0.115,0.081,0.054,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 4650.0
Centroid 4:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.02,0.064,0.053,
0.053,0.031,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.02,0.134,0.122,0.078,0.101,0.301,0.58,1.216,2.03
8,2.793,3.324,3.767,3.68,3.27,2.843,2.582,1.935,1.082,0.465,0.123,0.002,0.0,0.0,0.0,0.0,0.0,0.0,0.
014,0.029,0.125,0.339,0.653,1.745,3.918,7.455,14.169,25.33,39.497,52.234,62.615,66.398,61.919
,49.806,36.434,22.16,11.194,4.43,0.925,0.122,0.0,0.0,0.0,0.0,0.0,0.0,0.015,0.065,0.291,0.923,2.33
4,5.683,13.045,25.739,46.193,72.546,102.49,128.788,146.187,153.247,145.078,124.021,95.53,62.
7,34.649,14.919,4.018,0.543,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.238,0.801,2.136,5.67,14.357,30.917,57.0
23,91.766,127.872,159.628,181.64,194.042,197.243,194.422,181.301,152.213,113.881,70.529,34.
329,10.884,1.397,0.02,0.0,0.0,0.0,0.0,0.0,0.0,0.288,1.232,4.017,12.562,29.875,59.213,97.287,137.
088,168.355,186.092,193.896,194.122,193.314,195.142,195.52,184.669,155.138,110.225,62.258,
23.481,3.906,0.156,0.0,0.0,0.0,0.0,0.011,0.0,0.365,2.156,8.412,24.912,54.341,94.651,136.887,169
.71,183.351,180.709,169.761,158.206,152.313,155.332,168.313,182.658,179.312,145.182,92.528,
40.055,8.329,0.199,0.0,0.0,0.0,0.0,0.0,0.039,0.706,4.036,16.922,44.767,85.233,131.901,168.791,1
81.867,171.884,150.722,126.267,108.34,97.42,99.927,119.829,155.176,179.409,169.121,121.78,5
9.25,13.708,0.24,0.0,0.0,0.0,0.0,0.0,0.053,1.125,8.789,30.986,69.942,119.29,163.248,182.039,171
.023,140.769,107.683,79.986,60.481,51.176,53.655,74.95,118.923,166.426,180.721,145.418,79.2
74,20.567,0.309,0.0,0.0,0.0,0.0,0.0,0.036,2.259,16.535,49.786,100.955,152.13,183.029,175.493,1
41.02,99.74,66.497,43.364,30.705,25.881,29.209,46.502,91.225,150.728,184.239,160.257,96.144,
27.712,0.529,0.0,0.0,0.0,0.0,0.0,0.0,3.79,27.374,74.49,133.335,177.754,184.672,151.219,102.483,
60.265,34.55,21.46,16.22,14.85,17.641,33.16,75.741,139.179,183.079,167.014,106.331,33.808,0.
843,0.0,0.0,0.0,0.0,0.0,0.065,6.56,43.306,101.782,161.853,190.451,170.049,118.239,64.768,30.46
7,17.073,12.187,10.399,9.771,12.634,28.659,71.299,136.36,182.874,167.352,108.484,36.986,1.17
7,0.0,0.0,0.0,0.0,0.0,0.098,10.873,62.376,128.077,182.646,189.436,145.073,83.498,34.919,14.874
,9.758,8.028,7.359,7.468,12.206,32.328,79.634,142.745,182.544,161.748,103.049,36.572,1.387,0.
0,0.0,0.0,0.0,0.0,0.227,17.308,82.242,150.242,192.091,177.386,118.149,54.187,18.521,8.62,6.183
,5.734,6.03,7.582,17.349,46.074,98.555,155.33,180.324,150.089,91.066,31.336,1.204,0.0,0.0,0.0,
0.0,0.0,0.372,23.527,98.293,164.793,194.292,162.976,95.803,36.254,11.566,6.273,5.099,5.434,7.
481,14.173,33.935,74.073,127.374,169.908,171.561,130.225,73.005,23.039,0.918,0.0,0.0,0.0,0.0,
0.0,0.554,29.765,108.405,170.886,192.335,154.123,86.79,32.036,11.649,7.615,7.916,10.528,17.8
56,34.804,67.619,113.108,156.96,176.352,152.792,104.075,51.596,14.441,0.588,0.0,0.0,0.0,0.0,0.
009,0.836,32.604,107.464,168.743,190.984,158.04,96.189,45.487,23.134,18.323,21.152,29.926,4
7.58,75.941,114.248,152.567,175.425,163.802,122.526,74.263,32.032,7.813,0.287,0.0,0.0,0.0,0.0,
0.053,0.916,30.33,96.555,158.67,190.355,174.317,127.827,82.414,57.343,52.365,58.541,74.09,98
.77,129.508,159.159,176.17,166.359,132.181,87.248,45.314,16.169,2.965,0.137,0.0,0.0,0.0,0.0,0.
012,0.649,22.582,75.259,136.759,181.457,193.098,173.483,143.028,122.577,116.289,121.943,13
7.143,157.379,173.942,180.192,166.071,132.781,91.291,51.721,22.38,6.26,0.951,0.039,0.0,0.0,0.
0,0.0,0.0,0.461,12.886,48.699,100.837,154.24,191.354,204.984,198.905,188.786,184.023,184.593
,188.28,189.829,181.581,159.495,124.859,85.464,49.385,22.894,7.755,1.793,0.26,0.003,0.019,0.0
08,0.0,0.0,0.0,0.216,4.913,23.612,58.311,103.407,149.069,183.882,201.977,205.824,203.932,198.
48,186.928,166.418,137.032,101.816,67.386,38.824,18.095,6.931,1.952,0.31,0.069,0.0,0.0,0.015,
0.0,0.0,0.0,0.012,1.117,6.352,19.813,44.526,76.259,108.598,134.085,148.296,149.424,137.644,11
7.284,90.631,63.979,40.148,21.718,10.108,4.053,1.386,0.397,0.122,0.04,0.012,0.0,0.0,0.0,0.0,0.0,
0.0,0.248,0.74,2.679,6.805,13.684,22.685,31.629,37.527,38.639,35.359,28.251,20.226,12.705,7.2
19,3.381,1.536,0.717,0.381,0.175,0.067,0.027,0.015,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.019,0.119,0.3
16,0.496,0.704,0.884,0.937,0.854,0.695,0.456,0.292,0.219,0.176,0.155,0.113,0.106,0.04,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.01,0.026,0.108,0.139,0.099,0.104,0.107,0.025,0.0,0.021,
0.036,0.027,0.004,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 4777.0
Centroid 5:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.01,0.0,0.0,0.0,0.026,0.052,0.043,0.09,0.182,0.491,0.689,0.753,
0.786,0.6,0.209,0.057,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.003,0.049,0.027,0.023,0.052,0.032,0.093,0.
172,0.293,0.745,2.014,4.978,10.626,17.493,22.648,23.908,20.073,12.939,7.223,3.313,0.952,0.388
,0.106,0.0,0.0,0.0,0.0,0.0,0.034,0.052,0.109,0.197,0.548,0.77,1.481,1.959,2.984,5.343,10.384,21.3
42,39.7,57.914,69.373,66.744,52.45,33.928,18.914,9.005,3.63,1.568,0.296,0.01,0.0,0.0,0.0,0.004,
0.051,0.02,0.232,0.508,1.137,1.654,2.625,3.714,6.232,11.587,24.624,51.575,85.278,109.987,114.
575,101.121,74.319,46.24,26.934,14.283,7.113,3.403,0.873,0.06,0.0,0.0,0.0,0.064,0.11,0.092,0.33
9,0.73,1.309,1.846,2.718,3.908,7.426,17.702,45.168,88.661,129.016,146.366,134.547,107.186,74.
189,46.856,28.693,17.336,10.583,5.662,1.721,0.188,0.017,0.0,0.0,0.029,0.037,0.181,0.441,0.776,
1.261,1.863,2.605,4.01,10.01,31.567,78.152,130.245,159.615,154.727,127.065,94.262,64.571,42.
863,29.106,20.483,13.913,7.492,2.573,0.344,0.08,0.0,0.0,0.0,0.068,0.112,0.317,0.545,0.988,1.496
,2.267,4.873,17.744,59.385,123.027,168.69,169.421,140.272,105.127,77.289,57.244,41.729,30.62
6,22.915,16.629,8.872,2.887,0.457,0.054,0.0,0.0,0.008,0.029,0.048,0.232,0.488,0.844,1.297,2.522
,8.583,37.946,105.331,169.307,186.684,153.869,110.321,78.009,61.58,50.199,38.895,28.525,21.0
62,15.173,8.403,2.697,0.447,0.033,0.0,0.0,0.003,0.043,0.115,0.223,0.369,0.738,1.151,4.002,20.04
7,78.444,159.113,199.469,171.992,115.641,72.235,50.633,46.239,42.849,34.198,23.285,15.691,1
0.64,5.581,1.713,0.227,0.0,0.0,0.0,0.0,0.015,0.139,0.231,0.378,0.646,1.73,10.04,49.244,132.682,2
00.318,195.898,132.139,73.879,42.621,34.058,37.105,39.284,32.301,20.046,11.939,6.632,2.863,0
.92,0.156,0.0,0.0,0.0,0.0,0.02,0.119,0.19,0.331,0.851,4.82,27.551,95.975,182.287,212.293,163.45
1,90.341,48.563,35.696,36.584,43.235,46.758,37.883,21.862,11.025,5.012,1.956,0.618,0.206,0.0,
0.0,0.003,0.012,0.067,0.217,0.222,0.48,2.482,13.532,58.66,145.165,209.183,195.1,123.178,68.38
3,52.747,55.417,63.642,70.478,68.564,50.983,27.012,11.516,4.569,1.8,0.461,0.221,0.0,0.0,0.013,
0.035,0.044,0.135,0.293,1.015,6.748,31.597,97.171,178.846,208.861,161.186,97.126,75.875,82.3
89,94.445,106.242,112.01,99.944,68.768,33.584,12.583,4.512,1.681,0.457,0.198,0.0,0.0,0.0,0.0,0.
065,0.238,0.843,3.427,15.887,56.224,129.779,190.416,185.757,127.286,91.37,95.014,111.982,12
6.844,139.802,146.313,130.107,87.337,39.946,14.256,4.729,1.646,0.52,0.286,0.0,0.0,0.0,0.007,0.
2,0.655,2.427,9.005,30.368,80.155,146.229,182.25,155.13,107.726,94.798,106.473,116.818,127.7
92,147.142,161.755,148.532,97.93,43.128,14.696,4.475,1.491,0.744,0.34,0.023,0.0,0.019,0.084,0.
```

```
388,1.773,6.511,18.778,47.246,97.964,150.291,165.797,134.489,102.303,96.74,99.963,99.815,10
9.967,141.441,168.736,154.282,96.13,39.705,12.839,4.132,1.771,1.015,0.395,0.077,0.0,0.001,0.1
21,0.591,3.822,12.341,30.405,63.095,109.679,147.395,153.394,127.85,103.096,92.536,85.951,87.
203,111.807,154.206,177.101,145.714,79.361,29.729,9.235,3.49,2.022,1.074,0.348,0.043,0.0,0.0,
0.036,0.785,5.73,17.938,39.864,73.337,113.769,145.074,151.987,133.075,108.747,94.859,93.509,
111.881,149.085,180.96,172.295,115.332,54.179,18.905,6.584,3.26,2.072,1.039,0.272,0.061,0.0,0
.011,0.052,0.872,6.479,20.095,42.357,74.619,112.083,144.826,161.598,154.358,138.101,130.865,
140.334,164.377,188.266,182.844,135.858,73.165,29.311,9.918,4.27,2.414,1.649,0.974,0.326,0.0
51,0.0,0.027,0.116,0.946,5.345,17.066,37.44,66.614,102.673,141.175,171.926,185.88,186.835,18
6.642,192.09,195.649,180.341,135.804,79.504,35.177,12.816,4.981,2.484,1.636,1.031,0.528,0.12
7,0.0,0.0,0.0,0.138,0.695,3.363,11.211,25.487,46.9,76.631,113.674,151.78,180.481,194.159,195.1
35,183.527,156.909,114.904,68.562,32.624,13.282,5.093,1.981,0.983,0.721,0.451,0.189,0.064,0.0
,0.0,0.0,0.084,0.345,1.644,5.408,12.78,24.715,40.975,61.051,83.919,103.171,115.325,114.233,97.
84,70.5,43.472,22.395,9.497,3.675,1.305,0.524,0.325,0.302,0.15,0.073,0.031,0.0,0.0,0.0,0.017,0.0
83,0.548,1.937,4.435,9.126,16.219,23.836,29.757,33.672,33.439,29.454,21.295,13.748,8.477,4.79
5,2.209,1.098,0.492,0.317,0.128,0.017,0.047,0.002,0.0,0.0,0.0,0.0,0.0,0.0,0.068,0.258,0.644,1.235
,1.899,2.661,3.341,3.791,3.673,3.2,2.36,1.603,1.614,1.326,0.538,0.273,0.119,0.039,0.046,0.021,0.
001,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.033,0.05,0.049,0.043,0.114,0.218,0.218,0.144,0.096,0.079
,0.071,0.142,0.114,0.017,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.026,0.017,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 4909.0
Centroid 6:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.021,0.03,0.047,0.01,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.009,0.079,0.146,0.115,0.065,0.033,0.045,0.018,0.0,0
.0,0.0,0.0,0.0,0.03,0.034,0.032,0.034,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.008,0.102,0.237,0.25
2,0.357,0.499,0.449,0.345,0.175,0.102,0.249,0.441,0.365,0.327,0.43,0.452,0.329,0.193,0.141,0.05
,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.063,0.403,0.72,1.196,1.945,2.461,2.783,3.192,3.215,2.184,1.72,2.367
,3.848,5.034,5.465,4.846,4.059,3.276,1.893,0.937,0.256,0.026,0.002,0.0,0.0,0.0,0.0,0.044,0.472,1.
74,3.957,6.349,9.555,13.98,16.389,16.554,15.628,13.239,11.852,13.186,17.464,21.257,21.764,18.
077,14.459,9.922,6.286,3.136,1.25,0.23,0.061,0.0,0.0,0.0,0.096,0.461,1.635,4.628,9.453,16.786,2
5.782,39.006,49.822,54.193,57.25,57.351,55.587,54.622,54.275,53.25,48.899,38.619,27.536,18.6
9,11.826,6.659,2.798,0.642,0.113,0.0,0.0,0.045,0.446,1.541,4.432,10.605,20.153,35.499,57.569,8
2.954,100.696,106.102,104.517,100.185,95.106,93.794,93.162,90.762,80.026,61.246,43.275,27.4
39,16.533,8.645,3.703,0.95,0.198,0.0,0.0,0.347,1.801,4.462,9.857,19.515,35.589,62.514,98.515,1
30.993,141.865,130.174,112.345,100.188,94.65,99.626,111.26,118.082,109.756,85.218,56.914,34
.411,18.646,8.205,3.008,1.024,0.223,0.0,0.014,0.631,2.844,6.847,14.625,28.665,53.433,91.885,13
4.47,159.079,146.941,114.901,87.886,72.957,69.631,81.401,106.469,130.761,131.477,105.677,68
.849,38.112,18.392,6.575,2.084,0.688,0.081,0.0,0.034,0.87,2.962,7.44,17.45,36.772,69.732,115.4
55,154.313,157.208,122.054,80.164,54.685,43.36,43.66,61.586,97.174,134.213,144.012,120.175,
76.268,39.419,16.989,5.105,1.339,0.324,0.006,0.0,0.033,0.735,2.508,6.732,17.732,41.841,81.009,
130.176,157.243,138.407,89.927,49.848,31.25,25.548,30.366,51.535,92.949,138.052,153.521,129
.114,78.809,38.216,15.436,5.008,1.4,0.317,0.041,0.0,0.012,0.425,1.725,5.615,17.809,45.127,88.7
34,136.878,153.957,123.241,70.624,35.994,23.409,23.056,32.374,56.936,100.143,148.039,162.54
7,132.179,77.898,35.686,13.873,5.56,2.229,0.291,0.045,0.011,0.0,0.156,0.978,4.447,17.394,47.01
6,94.084,140.611,153.735,122.707,76.543,48.82,40.129,44.016,56.39,80.612,119.643,163.353,17
1.009,132.975,75.952,34.05,14.302,7.179,3.379,0.429,0.085,0.044,0.0,0.057,0.48,3.696,16.407,46
.321,93.923,138.818,157.408,138.975,107.688,87.625,83.398,89.343,100.474,117.523,147.085,18
0.772,180.082,134.817,76.67,36.001,17.475,10.069,4.429,0.773,0.113,0.005,0.0,0.001,0.216,3.14
2,13.596,40.893,83.65,126.687,152.102,153.121,142.66,133.605,132.307,135.416,138.878,144.25
6,164.666,193.581,188.753,135.418,76.397,37.205,19.782,10.816,4.993,1.211,0.189,0.0,0.0,0.0,0.
052,2.107,9.476,31.585,64.294,98.1,125.679,140.357,144.116,144.155,142.729,140.003,135.47,1
```

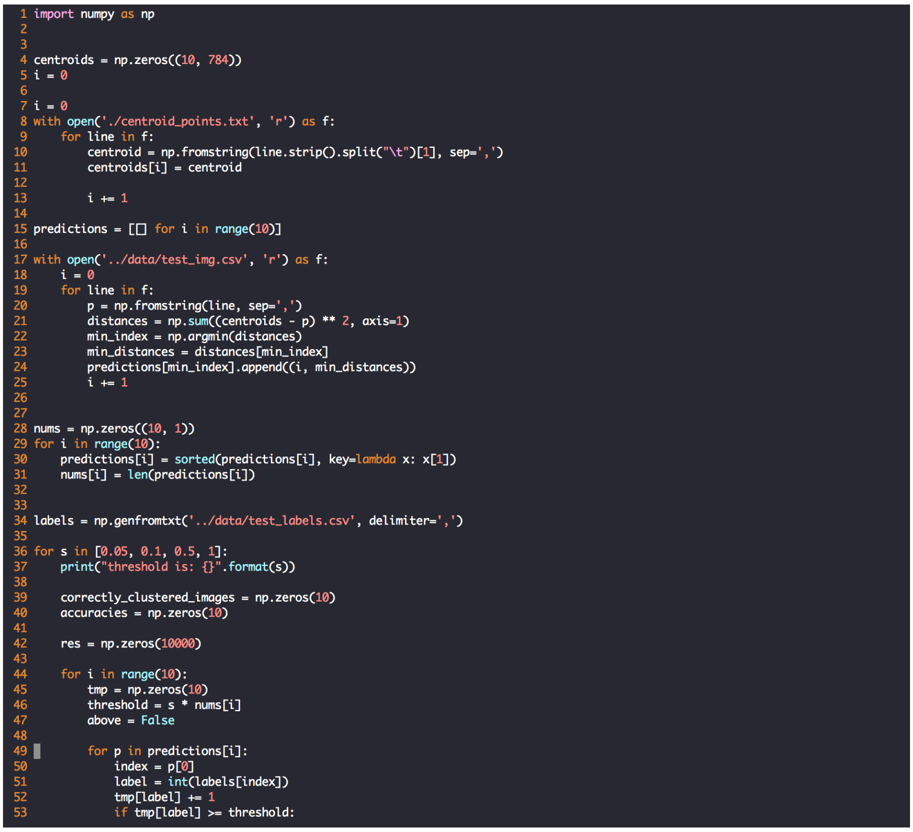
```
34.307,160.624,196.688,190.648,128.896,69.774,34.779,18.715,9.988,4.747,1.371,0.386,0.007,0.
0,0.0,0.061,1.121,6.011,20.015,39.956,61.73,81.167,95.625,103.014,104.651,101.874,96.21,89.23
5,95.413,141.035,190.243,181.246,114.914,59.424,28.888,15.194,7.827,3.838,1.074,0.244,0.019,
0.0,0.0,0.05,0.749,3.223,10.146,19.765,30.242,39.414,46.767,50.938,51.873,49.859,46.027,43.39
8,61.329,124.493,180.336,165.735,99.347,49.365,23.616,12.504,6.404,2.673,0.755,0.019,0.0,0.0,
0.0,0.04,0.428,1.367,4.221,7.541,10.891,14.488,17.086,18.152,18.351,17.732,17.697,21.283,49.1
41,118.529,171.179,151.163,89.833,45.1,21.915,11.414,5.515,2.005,0.605,0.047,0.0,0.0,0.0,0.023
,0.187,0.451,1.543,2.609,3.823,4.843,5.777,6.344,6.755,6.725,8.721,16.608,50.473,119.384,165.0
43,141.281,85.089,44.248,21.854,11.208,5.014,1.718,0.463,0.056,0.0,0.0,0.0,0.001,0.021,0.113,0.
69,1.095,1.923,2.652,3.524,4.446,4.748,5.78,9.719,21.12,58.443,121.267,158.556,132.905,82.63,
44.911,22.043,11.06,4.458,1.351,0.312,0.009,0.0,0.0,0.0,0.0,0.0,0.035,0.342,0.679,1.18,2.13,2.94
3,3.856,4.702,6.986,12.539,27.94,65.788,121.086,150.088,124.154,80.684,45.764,23.126,10.906,
4.531,1.233,0.208,0.0,0.0,0.0,0.0,0.0,0.0,0.003,0.177,0.647,1.042,1.82,2.738,3.768,5.077,8.245,15
.542,33.257,68.431,113.857,134.75,112.583,75.218,43.953,21.782,10.085,4.279,1.102,0.117,0.0,0
.0,0.0,0.0,0.0,0.0,0.021,0.135,0.423,0.973,1.51,2.374,3.529,5.043,8.4,16.149,30.796,58.148,89.78
4,103.659,87.103,58.133,33.256,16.462,7.341,3.036,0.671,0.098,0.018,0.0,0.0,0.0,0.0,0.0,0.058,0.
075,0.149,0.347,0.684,1.03,1.823,2.731,5.098,9.868,18.466,33.601,51.718,58.354,48.344,31.371,
16.907,8.273,3.562,1.332,0.289,0.039,0.019,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.009,0.042,0.082,0.15
6,0.243,0.945,3.004,6.783,12.901,18.792,19.85,15.895,10.337,5.759,2.502,0.731,0.124,0.005,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.007,0.042,0.039,0.007,0.0,0.006,0.204,0.646,1.73,2.547,3.458,3.1
63,2.001,0.959,0.499,0.211,0.165,0.022,0.0,0.0,0.0,0.0], 5483.0
Centroid 7:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.004,0.002,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.016,0.043,0.052,0
.002,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.013,0.004,0.0,0.0,0.0,0.0,0.0,0.015,0.081,0.069,0.
088,0.013,0.028,0.063,0.068,0.132,0.242,0.682,0.742,0.469,0.295,0.088,0.016,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.018,0.01,0.014,0.068,0.284,0.448,0.467,0.426,0.306,0.371,0.543,1.151,3.073,5.2
21,5.517,4.759,3.03,1.248,0.323,0.044,0.0,0.0,0.0,0.0,0.0,0.023,0.064,0.129,0.24,0.545,0.836,1.19
2,1.977,3.425,4.926,5.283,4.924,4.812,5.442,7.85,12.681,17.357,19.173,17.506,13.192,6.509,1.83
2,0.311,0.007,0.0,0.0,0.02,0.014,0.163,0.595,1.381,2.427,4.306,6.947,10.997,16.087,23.163,31.04
5,37.133,43.387,49.854,54.733,57.089,55.406,50.386,43.135,33.51,23.475,12.828,4.78,1.163,0.08
4,0.0,0.007,0.049,0.178,0.828,2.592,5.523,10.136,16.765,25.703,38.241,55.43,76.784,97.24,108.8
01,114.313,118.395,123.253,129.937,128.434,110.234,83.485,55.921,35.396,18.931,8.34,1.988,0.
035,0.008,0.041,0.074,0.733,2.151,5.248,10.998,20.351,32.617,51.089,79.395,115.141,148.06,16
2.893,159.29,148.22,142.482,148.642,166.273,172.097,149.921,110.38,69.511,39.397,20.59,8.91
5,1.769,0.066,0.029,0.046,0.129,1.011,3.244,7.674,15.224,27.796,46.744,78.155,121.558,163.35,
181.29,166.65,139.191,116.689,110.873,129.832,164.15,178.644,155.412,109.099,63.76,32.151,1
4.475,5.855,1.154,0.041,0.033,0.011,0.13,1.184,3.699,8.13,16.054,30.461,57.801,99.986,147.879,
171.507,153.773,112.629,78.387,63.66,72.04,111.961,161.142,174.043,141.989,90.297,45.786,19
.765,8.172,3.343,0.637,0.043,0.004,0.013,0.142,0.792,3.093,7.27,15.552,34.14,70.204,119.455,15
3.162,143.82,97.808,54.016,32.834,33.327,64.87,127.0,174.629,166.905,119.298,64.244,27.991,1
0.486,3.808,1.362,0.35,0.016,0.0,0.0,0.084,0.426,1.952,5.881,16.422,41.557,83.941,127.688,140.
024,106.684,58.749,31.402,26.762,44.805,100.891,166.598,186.046,148.552,87.749,39.318,15.42
6,5.332,1.781,0.534,0.138,0.0,0.0,0.0,0.041,0.22,1.488,6.083,20.469,49.676,91.193,123.275,121.6
14,88.299,58.354,48.293,55.065,92.135,157.565,198.439,178.101,117.071,58.525,24.244,9.382,3.
573,1.218,0.324,0.055,0.0,0.0,0.0,0.007,0.075,1.526,8.124,25.725,54.55,88.166,110.063,108.91,9
5.453,85.465,83.998,100.812,150.681,201.678,203.277,150.154,83.0,38.982,18.098,8.72,4.043,1.
422,0.304,0.02,0.0,0.0,0.0,0.0,0.183,1.956,10.59,28.046,51.621,75.879,91.894,98.436,100.42,100.
959,107.493,138.925,194.245,216.896,180.29,109.656,53.822,26.229,14.488,8.263,4.341,1.632,0.
417,0.038,0.001,0.0,0.0,0.012,0.318,2.178,10.732,24.635,40.605,55.76,66.346,74.898,79.74,87.06
1,110.418,168.703,216.837,201.993,133.051,64.669,29.22,15.206,9.107,5.473,2.706,1.253,0.286,
0.041,0.002,0.0,0.0,0.025,0.386,1.952,7.899,16.268,25.473,32.264,37.89,43.006,50.534,69.802,12
5.415,200.103,215.037,156.6,76.086,28.187,11.637,6.332,3.987,2.326,1.262,0.688,0.188,0.038,0.
002,0.017,0.0,0.105,0.359,1.772,5.244,10.262,14.625,16.783,18.902,23.464,40.042,84.189,166.85
,217.399,183.328,97.569,32.478,9.033,3.574,2.189,1.452,1.075,0.637,0.475,0.101,0.013,0.001,0.0
,0.015,0.1,0.387,1.74,4.918,9.838,12.695,13.22,15.507,27.065,59.783,129.594,204.222,204.277,1
30.45,47.882,11.327,3.042,1.397,1.134,0.885,0.715,0.426,0.269,0.102,0.0,0.0,0.0,0.0,0.043,0.404,
1.473,4.925,11.537,15.495,17.778,25.548,50.148,101.183,177.638,210.966,162.685,74.09,18.252,
4.373,1.651,1.017,0.836,0.789,0.582,0.397,0.175,0.092,0.0,0.0,0.0,0.0,0.017,0.309,1.123,4.4,11.1
03,18.022,26.755,46.181,84.951,147.168,201.08,182.832,106.843,32.532,7.184,2.497,1.36,0.843,
0.779,0.754,0.586,0.361,0.243,0.091,0.015,0.0,0.0,0.0,0.0,0.183,0.725,3.289,9.373,19.053,37.58,6
8.944,118.108,173.075,187.313,135.22,57.457,12.97,3.499,1.648,1.1,0.829,0.712,0.59,0.57,0.314,
0.174,0.05,0.002,0.0,0.0,0.0,0.0,0.117,0.881,2.989,9.321,23.837,48.42,85.979,134.39,169.353,150
.485,88.187,26.743,5.934,2.196,1.047,0.665,0.598,0.597,0.499,0.464,0.3,0.128,0.029,0.0,0.0,0.0,0
.0,0.0,0.05,0.875,3.27,10.778,27.217,54.03,90.929,129.355,139.2,104.676,47.812,12.053,2.994,1.
287,0.642,0.393,0.361,0.351,0.32,0.226,0.176,0.061,0.016,0.0,0.0,0.0,0.0,0.0,0.079,0.655,3.295,1
0.294,23.857,46.435,73.181,94.987,88.53,54.87,19.382,4.575,1.241,0.519,0.218,0.133,0.132,0.15
4,0.161,0.11,0.053,0.022,0.0,0.0,0.0,0.0,0.0,0.0,0.006,0.327,1.502,4.905,10.933,20.994,30.301,36.
359,31.518,16.023,4.378,0.887,0.207,0.076,0.042,0.012,0.0,0.0,0.026,0.039,0.031,0.0,0.009,0.0,0.
0,0.0,0.0,0.0,0.0,0.024,0.145,0.381,0.777,1.052,1.344,1.754,1.707,0.787,0.245,0.114,0.025,0.0,0.0
,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 6453.0
Centroid 8:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.037,0.081,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.005,0.018,0.141,0.395,0.708,1.137,2.146,2.821,3.126,3.182,3.115,2.5
51,2.092,1.589,0.713,0.132,0.091,0.082,0.104,0.068,0.0,0.0,0.0,0.0,0.0,0.0,0.02,0.0,0.053,0.048,0.
704,2.004,3.649,7.083,12.78,19.428,28.792,38.559,45.749,44.938,37.541,25.158,13.609,6.133,1.9
73,0.457,0.268,0.136,0.0,0.0,0.0,0.0,0.0,0.0,0.045,0.023,0.026,0.342,1.499,4.208,8.335,16.038,28.
162,43.672,61.836,82.279,98.662,94.882,73.797,46.471,25.408,12.247,4.714,1.476,0.764,0.177,0.
064,0.001,0.0,0.0,0.0,0.0,0.0,0.023,0.082,0.776,2.591,6.387,13.42,26.876,45.743,68.15,96.252,12
3.152,129.354,109.522,77.214,46.391,24.703,12.357,5.828,2.31,1.099,0.609,0.305,0.031,0.0,0.0,0
.0,0.0,0.0,0.0,0.164,1.27,3.579,9.16,20.634,39.492,64.587,95.426,132.03,148.703,131.237,93.498,
56.745,31.733,16.11,8.645,4.663,2.265,1.677,1.413,0.74,0.108,0.0,0.0,0.0,0.0,0.0,0.013,0.255,1.8
43,5.578,14.062,29.536,55.946,88.571,130.411,159.708,147.572,106.926,62.6,32.463,16.922,9.10
8,5.703,4.004,3.365,2.865,2.305,0.778,0.205,0.027,0.0,0.0,0.0,0.0,0.122,0.419,2.772,8.467,20.169
,41.886,75.289,117.912,162.849,166.722,125.562,73.684,35.65,18.623,11.098,7.525,5.438,4.678,
4.518,4.006,2.922,0.88,0.363,0.141,0.0,0.0,0.0,0.0,0.198,0.75,3.541,11.251,28.124,56.108,97.606,
148.609,178.964,149.038,92.07,43.628,20.885,13.755,11.105,10.881,10.231,8.052,6.714,5.394,3.
492,1.299,0.506,0.191,0.0,0.0,0.0,0.009,0.238,0.93,4.562,15.241,37.176,73.501,122.848,173.69,1
71.994,117.85,58.231,23.846,16.337,16.289,17.456,20.386,22.527,20.251,14.847,9.086,4.507,1.5
99,0.576,0.281,0.028,0.0,0.0,0.044,0.225,1.072,5.79,20.008,47.653,92.635,149.659,182.77,149.95
1,84.135,32.872,17.826,20.7,27.122,35.46,43.593,47.347,43.63,32.395,19.284,7.916,2.832,0.865,
0.402,0.065,0.0,0.0,0.0,0.274,0.991,6.842,25.432,58.792,113.517,171.609,176.911,122.803,55.34
6,23.592,25.765,38.614,56.572,73.041,85.243,88.774,79.91,60.869,37.0,17.895,5.875,1.314,0.193
,0.044,0.0,0.0,0.0,0.163,0.916,8.305,31.032,71.251,133.225,180.908,162.403,98.209,41.659,32.76
4,49.339,76.133,103.761,123.743,134.716,135.895,125.281,98.704,62.786,31.799,12.291,2.379,0.
197,0.048,0.0,0.0,0.0,0.099,0.798,9.529,35.809,84.451,148.897,181.288,147.018,82.022,46.443,5
7.245,88.519,123.277,147.346,157.895,159.414,159.849,153.593,129.321,88.024,48.097,20.881,4
.25,0.256,0.013,0.0,0.0,0.0,0.0,0.821,10.873,41.405,96.301,157.323,178.636,136.879,80.808,66.9
54,91.946,128.268,154.005,158.504,149.06,142.053,148.892,160.395,146.73,106.484,61.631,28.3
88,6.726,0.311,0.0,0.0,0.0,0.0,0.0,0.777,11.465,45.162,101.772,160.146,176.434,137.871,94.637,
93.097,120.68,147.071,150.659,130.327,108.357,103.788,124.443,156.576,156.085,118.077,69.1,
31.361,8.519,0.391,0.0,0.0,0.0,0.0,0.0,0.684,11.427,44.004,99.812,158.173,177.933,150.341,116.
371,114.866,132.426,141.082,124.873,95.175,76.116,83.064,120.199,160.639,159.388,119.244,6
```

```
6.677,29.761,8.45,0.343,0.0,0.0,0.0,0.0,0.02,0.684,10.068,37.847,88.632,148.967,182.708,174.79
9,149.348,135.513,136.852,132.063,110.322,86.49,81.277,104.941,147.536,174.398,156.125,105.
308,54.684,22.458,6.031,0.32,0.022,0.0,0.0,0.0,0.138,0.75,7.483,27.822,70.17,128.692,179.141,1
97.679,188.22,167.243,154.986,143.933,128.311,120.863,130.661,156.618,180.22,172.563,132.2
92,77.722,36.767,13.218,3.261,0.215,0.048,0.0,0.0,0.0,0.087,0.543,4.106,15.592,44.616,94.869,1
53.013,197.395,212.76,206.796,194.254,183.252,176.792,178.192,184.855,188.019,176.322,139.
856,90.077,45.675,19.14,6.05,1.271,0.153,0.0,0.0,0.0,0.0,0.057,0.296,1.295,6.015,19.905,49.861,
98.199,151.993,193.752,214.255,220.103,217.245,211.469,203.689,190.688,165.277,126.617,81.
937,43.557,18.881,7.344,2.218,0.647,0.121,0.0,0.0,0.0,0.0,0.0,0.146,0.393,1.595,5.652,16.128,37.
881,72.732,113.0,148.982,172.672,180.509,172.732,153.273,123.479,88.112,56.843,30.653,14.69
3,6.166,2.545,0.913,0.269,0.089,0.0,0.0,0.0,0.0,0.0,0.081,0.104,0.248,0.854,2.285,5.232,13.41,25.
196,39.051,51.708,59.465,58.032,48.63,36.197,23.909,14.116,7.96,4.136,2.038,1.091,0.519,0.171
,0.065,0.0,0.0,0.0,0.0,0.0,0.027,0.079,0.085,0.12,0.275,0.465,0.95,2.108,4.031,6.193,7.894,8.274,
7.243,5.412,3.663,2.357,1.794,1.087,0.733,0.428,0.361,0.123,0.018,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.013,0.099,0.196,0.436,0.813,1.156,1.161,0.954,0.83,0.497,0.338,0.58,0.543,0.402,0.233,0.0
83,0.01,0.001,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.027,0.081,0.047,0.083,0.054,0.00
6,0.0,0.0,0.076,0.197,0.11,0.029,0.005,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.047,0.032,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 3137.0
Centroid 9:
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.002,0.046,0.11,0.253,0.488,0.884,1.459,1.974,2.302,2.363,2.
384,2.093,1.676,1.341,0.998,0.697,0.462,0.374,0.14,0.0,0.0,0.0,0.0,0.0,0.0,0.005,0.011,0.019,0.28
7,1.402,3.504,7.582,14.196,23.576,36.052,48.436,58.897,65.345,65.378,58.736,46.881,33.471,22.
409,14.174,8.952,5.677,3.237,1.639,0.595,0.068,0.0,0.0,0.0,0.0,0.012,0.306,2.001,6.673,15.424,2
9.914,52.111,81.136,113.198,144.258,165.859,176.057,173.053,160.209,136.841,105.681,72.837,
46.936,28.733,17.168,9.414,4.909,2.13,0.473,0.068,0.0,0.0,0.001,0.027,1.174,5.505,14.56,31.653,
58.513,94.389,133.802,170.023,192.805,199.701,198.688,195.868,190.973,178.672,151.686,113.
206,73.326,43.333,24.23,12.885,6.79,2.903,0.85,0.137,0.0,0.0,0.004,0.101,2.002,8.573,20.683,43.
057,73.917,109.479,142.848,161.607,161.233,149.318,138.627,137.693,146.937,154.965,147.423
,119.747,80.534,44.947,22.968,11.068,5.275,2.447,0.721,0.081,0.0,0.0,0.0,0.101,2.428,9.483,21.0
23,41.973,68.797,96.502,115.259,116.519,100.433,79.506,69.292,75.63,96.928,119.517,126.86,1
10.222,75.401,40.012,18.771,8.241,3.492,1.556,0.493,0.006,0.0,0.0,0.0,0.032,1.744,7.135,16.815,
33.751,55.182,76.272,86.87,81.621,63.677,45.527,42.626,58.082,85.337,109.506,117.345,99.94,6
5.636,33.539,14.822,6.031,2.298,0.738,0.174,0.0,0.0,0.0,0.0,0.031,1.205,4.447,11.505,24.892,43.
256,62.326,74.273,72.797,62.31,54.827,62.054,84.477,108.723,121.617,114.334,87.929,54.24,26.
941,11.067,4.459,1.476,0.352,0.12,0.001,0.0,0.0,0.0,0.04,0.59,2.704,8.484,19.781,37.034,58.44,7
7.649,88.69,93.364,101.503,117.08,134.789,144.664,138.755,115.26,79.936,47.417,23.357,9.169,
3.148,1.217,0.35,0.102,0.0,0.0,0.0,0.0,0.007,0.352,1.687,6.213,16.817,34.523,61.079,91.153,118.
738,140.29,156.428,170.113,176.996,174.613,158.88,126.677,88.268,55.094,28.365,11.122,3.439
,1.277,0.316,0.074,0.0,0.0,0.0,0.0,0.0,0.165,1.241,4.559,13.926,31.313,60.487,97.239,133.051,15
9.01,176.034,182.552,182.295,176.114,163.794,142.374,113.666,78.654,44.626,19.621,6.698,2.2
43,0.608,0.123,0.004,0.01,0.016,0.0,0.0,0.09,0.831,3.485,10.074,24.857,50.261,83.284,113.261,1
33.451,144.361,146.397,145.459,144.422,145.566,145.56,136.333,109.581,70.01,34.708,13.55,4.
24,1.12,0.208,0.047,0.007,0.005,0.0,0.035,0.176,0.697,2.68,7.044,16.583,34.485,56.678,75.844,8
6.566,90.27,88.903,86.355,90.559,105.387,127.749,142.329,133.027,96.076,52.023,23.531,7.669,
1.829,0.332,0.097,0.012,0.0,0.0,0.093,0.457,1.379,3.041,6.075,11.838,23.189,36.884,46.048,49.7
24,47.945,42.658,38.891,43.807,61.043,97.458,135.871,144.288,115.315,68.192,32.615,12.2,3.05
1,0.457,0.125,0.024,0.0,0.018,0.225,1.066,2.906,5.394,8.802,14.004,22.454,31.348,36.159,35.168
,30.201,22.717,16.756,17.948,32.913,73.808,126.673,149.499,125.328,79.268,40.156,16.006,4.02
```

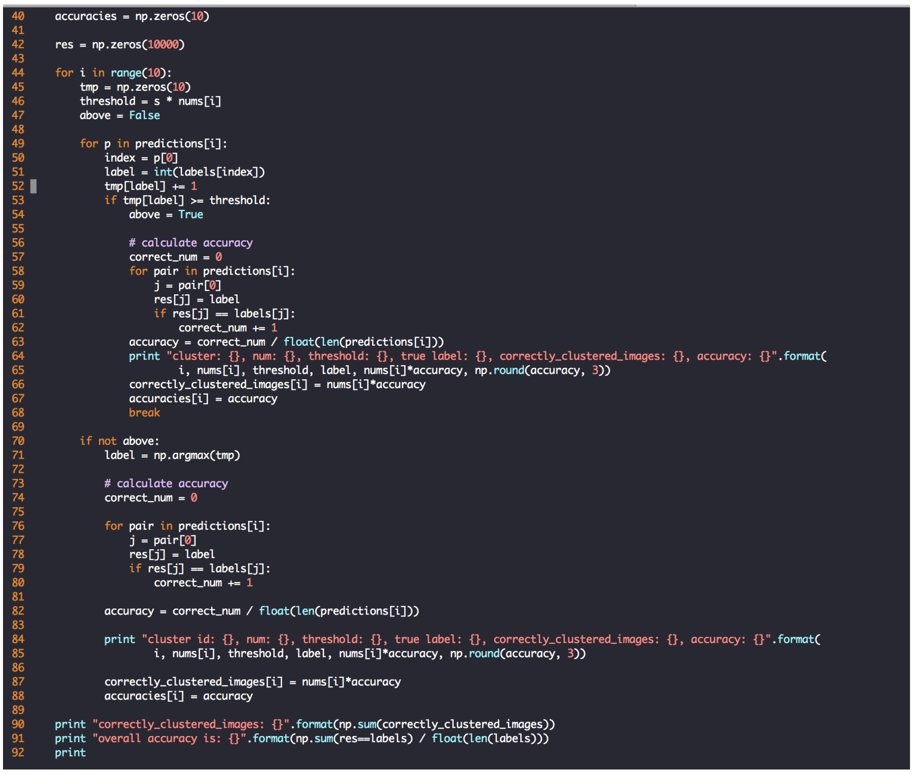
```
4,0.484,0.05,0.001,0.0,0.032,0.429,1.885,5.459,10.196,15.919,23.39,31.22,37.44,39.442,34.318,2
6.319,16.558,9.903,9.65,25.22,69.871,128.302,154.567,130.211,83.857,44.366,17.581,4.426,0.53
1,0.053,0.0,0.0,0.055,0.601,2.734,8.302,16.926,26.899,37.804,48.794,55.2,53.737,44.048,31.225,
19.204,13.821,19.024,43.741,93.43,145.928,161.911,129.605,82.342,42.363,15.732,3.79,0.578,0.
101,0.0,0.0,0.069,0.737,3.489,10.256,22.018,36.686,54.588,73.452,84.951,84.905,72.762,55.93,4
5.165,45.105,60.24,94.913,140.22,172.261,162.296,118.053,69.473,32.831,11.656,2.84,0.428,0.0
82,0.0,0.0,0.049,0.753,3.348,9.645,21.741,40.983,66.023,94.82,119.291,130.873,128.566,118.638
,112.971,118.401,138.099,165.344,185.768,178.309,140.865,90.407,47.774,20.3,6.825,1.709,0.39
,0.03,0.0,0.0,0.006,0.532,2.364,7.057,16.498,34.002,59.417,93.326,129.653,160.41,180.37,188.94
2,191.968,196.014,201.703,199.848,180.057,141.04,93.147,52.353,24.174,9.378,2.858,0.743,0.20
3,0.001,0.0,0.0,0.0,0.211,1.133,3.506,8.324,18.742,35.412,60.585,91.982,127.566,160.76,184.222
,194.745,191.716,175.985,147.66,109.869,71.412,40.833,19.35,7.815,2.782,0.814,0.176,0.055,0.0
11,0.0,0.0,0.0,0.025,0.32,0.946,2.482,5.829,10.878,19.3,31.737,46.614,63.296,77.063,83.884,79.9
79,67.613,50.238,32.54,18.302,8.833,3.623,1.361,0.427,0.154,0.074,0.031,0.0,0.0,0.0,0.0,0.0,0.03
5,0.118,0.245,0.553,0.973,1.577,2.441,3.335,4.011,4.7,5.084,4.82,4.137,3.072,1.929,1.03,0.448,0.
143,0.008,0.001,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.002,0.026,0.028,0.035,
0.026,0.004,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0], 7285.0
```

# Q1 (b):

code:



```python
import numpy as np


centroids = np.zeros((10, 784))
i = 0

i = 0
with open('./centroid_points.txt', 'r') as f:
    for line in f:
        centroid = np.fromstring(line.strip().split("\t")[1], sep=',')
        centroids[i] = centroid

        i += 1

predictions = [[] for i in range(10)]

with open('../data/test_img.csv', 'r') as f:
    i = 0
    for line in f:
        p = np.fromstring(line, sep=',')
        distances = np.sum((centroids - p) ** 2, axis=1)
        min_index = np.argmin(distances)
        min_distances = distances[min_index]
        predictions[min_index].append((i, min_distances))
        i += 1


nums = np.zeros((10, 1))
for i in range(10):
    predictions[i] = sorted(predictions[i], key=lambda x: x[1])
    nums[i] = len(predictions[i])


labels = np.genfromtxt('../data/test_labels.csv', delimiter=',')

for s in [0.05, 0.1, 0.5, 1]:
    print("threshold is: {}".format(s))

    correctly_clustered_images = np.zeros(10)
    accuracies = np.zeros(10)

    res = np.zeros(10000)

    for i in range(10):
        tmp = np.zeros(10)
        threshold = s * nums[i]
        above = False

        for p in predictions[i]:
            index = p[0]
            label = int(labels[index])
            tmp[label] += 1
            if tmp[label] >= threshold:
```



```python
    accuracies = np.zeros(10)

    res = np.zeros(10000)

    for i in range(10):
        tmp = np.zeros(10)
        threshold = s * nums[i]
        above = False

        for p in predictions[i]:
            index = p[0]
            label = int(labels[index])
            tmp[label] += 1
            if tmp[label] >= threshold:
                above = True

                # calculate accuracy
                correct_num = 0
                for pair in predictions[i]:
                    j = pair[0]
                    res[j] = label
                    if res[j] == labels[j]:
                        correct_num += 1
                accuracy = correct_num / float(len(predictions[i]))
                print "cluster: {}, num: {}, threshold: {}, true label: {}, correctly_clustered_images: {}, accuracy: {}".format(
                        i, nums[i], threshold, label, nums[i]*accuracy, np.round(accuracy, 3))
                correctly_clustered_images[i] = nums[i]*accuracy
                accuracies[i] = accuracy
                break

        if not above:
            label = np.argmax(tmp)

            # calculate accuracy
            correct_num = 0

            for pair in predictions[i]:
                j = pair[0]
                res[j] = label
                if res[j] == labels[j]:
                    correct_num += 1

            accuracy = correct_num / float(len(predictions[i]))

            print "cluster id: {}, num: {}, threshold: {}, true label: {}, correctly_clustered_images: {}, accuracy: {}".format(
                    i, nums[i], threshold, label, nums[i]*accuracy, np.round(accuracy, 3))

            correctly_clustered_images[i] = nums[i]*accuracy
            accuracies[i] = accuracy

    print "correctly_clustered_images: {}".format(np.sum(correctly_clustered_images))
    print "overall accuracy is: {}".format(np.sum(res==labels) / float(len(labels)))
    print
```

| **Table. 1.** | **The Accuracy** | **of Clustering** | **Performance** | **with** | **x = 5%** |
|---|---|---|---|---|---|
| Cluster Number | # images in the entire cluster | # of images considered (m) when determining the cluster label | Major Label of central images | # correctly clustered images | Classification Accuracy (%) |
| 0 | 1075 | 54 | 8 | 574 | 0.534 |
| 1 | 799 | 40 | 2 | 712 | 0.891 |
| 2 | 666 | 33 | 6 | 357 | 0.536 |
| 3 | 970 | 49 | 7 | 360 | 0.371 |
| 4 | 1285 | 64 | 3 | 668 | 0.52 |
| 5 | 1597 | 80 | 1 | 1109 | 0.694 |
| 6 | 1307 | 65 | 7 | 484 | 0.37 |
| 7 | 806 | 40 | 0 | 757 | 0.939 |
| 8 | 883 | 45 | 9 | 295 | 0.334 |
| 9 | 612 | 30 | 6 | 548 | 0.895 |
| Total Set | 10000 | 500 | NA | 5864 | 0.5864 |

| **Table. 2.** | **The Accuracy** | **of Clustering** | **Performance** | **with** | **x = 10%** |
|---|---|---|---|---|---|
| Cluster Number | # images in the entire cluster | # of images considered (m) when determining the cluster label | Major Label of central images | # correctly clustered images | Classification Accuracy (%) |
| 0 | 1075 | 108 | 8 | 574 | 0.534 |
| 1 | 799 | 80 | 2 | 712 | 0.891 |
| 2 | 666 | 66 | 6 | 357 | 0.536 |
| 3 | 970 | 98 | 7 | 360 | 0.371 |
| 4 | 1285 | 128 | 3 | 668 | 0.52 |
| 5 | 1597 | 160 | 1 | 1109 | 0.694 |
| 6 | 1307 | 130 | 9 | 458 | 0.35 |
| 7 | 806 | 80 | 0 | 757 | 0.939 |
| 8 | 883 | 90 | 9 | 295 | 0.334 |
| 9 | 612 | 60 | 6 | 548 | 0.895 |
| Total Set | 10000 | 500 | NA | 5838 | 0.5838 |

| Table. 3. | The Accuracy | of Clustering | Performance | with | x = 50% |
|---|---|---|---|---|---|
| Cluster Number | # images in the entire cluster | # of images considered (m) when determining the cluster label | Major Label of central images | # correctly clustered images | Classification Accuracy (%) |
| 0 | 1075 | 537 | 8 | 574 | 0.534 |
| 1 | 799 | 400 | 2 | 712 | 0.891 |
| 2 | 666 | 333 | 6 | 357 | 0.536 |
| 3 | 970 | 485 | 7 | 360 | 0.371 |
| 4 | 1285 | 643 | 3 | 668 | 0.52 |
| 5 | 1597 | 799 | 1 | 1109 | 0.694 |
| 6 | 1307 | 653 | 7 | 484 | 0.37 |
| 7 | 806 | 403 | 0 | 757 | 0.939 |
| 8 | 883 | 441 | 4 | 393 | 0.445 |
| 9 | 612 | 306 | 6 | 548 | 0.895 |
| Total Set | 10000 | 500 | NA | 5962 | 0.5962 |

| Table. 4. | The Accuracy | of Clustering | Performance | with | x = 100% |
|---|---|---|---|---|---|
| Cluster Number | # images in the entire cluster | # of images considered (m) when determining the cluster label | Major Label of central images | # correctly clustered images | Classification Accuracy (%) |
| 0 | 1075 | 537 | 8 | 574 | 0.534 |
| 1 | 799 | 400 | 2 | 712 | 0.891 |
| 2 | 666 | 333 | 6 | 357 | 0.536 |
| 3 | 970 | 485 | 7 | 360 | 0.371 |
| 4 | 1285 | 643 | 3 | 668 | 0.52 |
| 5 | 1597 | 799 | 1 | 1109 | 0.694 |
| 6 | 1307 | 653 | 7 | 484 | 0.37 |
| 7 | 806 | 403 | 0 | 757 | 0.939 |
| 8 | 883 | 441 | 4 | 393 | 0.445 |
| 9 | 612 | 306 | 6 | 548 | 0.895 |
| Total Set | 10000 | 500 | NA | 5962 | 0.5962 |

Best x: 50% and 100%

Explain:

1. Firstly observe that x=5% is better than x=10%, we rely more on the points that has smaller distance, in some sense we can reduce the noise (points that far away from centroid) influence. Let’s look cluster 6, when x=5%, it choose 7 as its label; when x=10%, it choose 9 as its label; however, in cluster 6, 7 is of 37% and 9 is of 35%, although the difference is not so big, but it really shows that sometimes choose nearby points has higher accuracy.

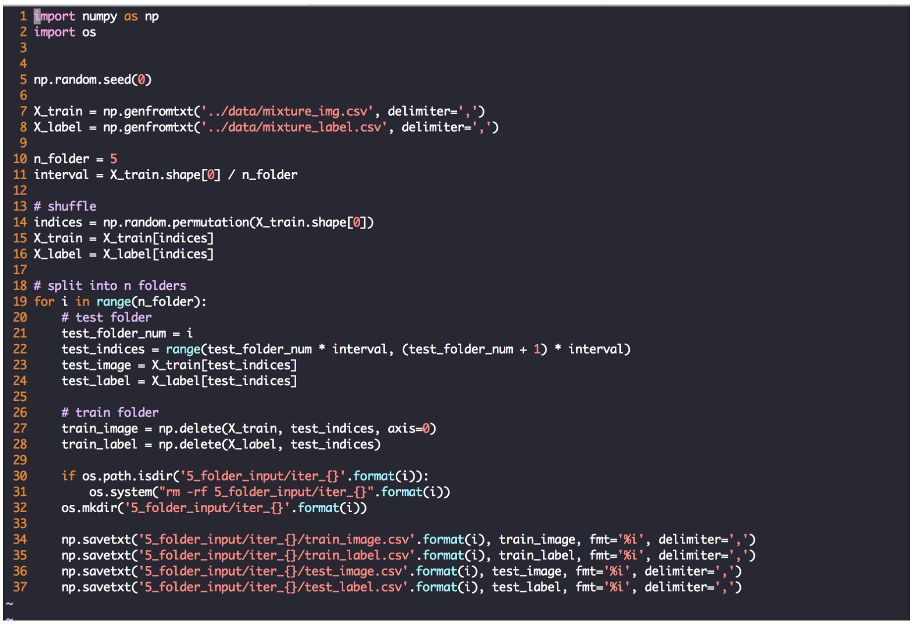
2. Secondly we observe x=50% and x=100% is better than x=5% and x=10%. In fact this is the phenomenon of “curse of dimensionality”, as dimension increase, the data points will concentrated more and more in a then-shell near the surface.
   Let’s look at cluster 8, in cluster 5% and cluster 10%, we can find the true label is 9; in cluster 50% and cluster 100%, we can find the true label is 4. While 9 is only of 0.334% and 4 is of 0.445%, although label 4 is more similar to centroid, but in the cluster it is minor; although label 9 is far away from centroid, but it is majority in the cluster

## Q1 (c):

code: (5_folder and 10_folder are almost same, just modify 5 to 10)

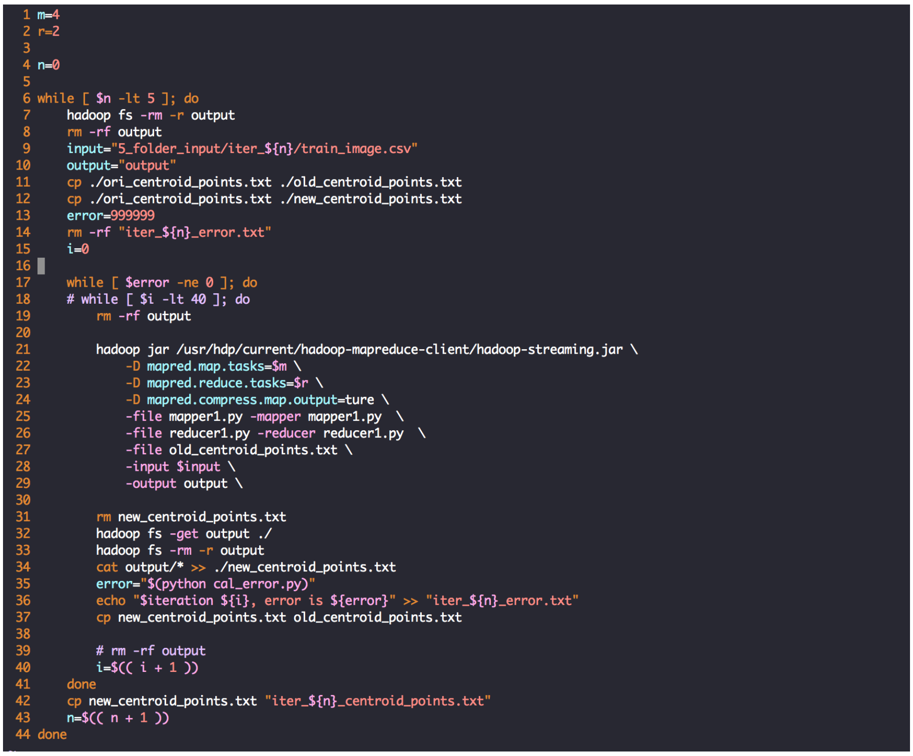
mapper and reducer are same with Q1 (a)

get_data.py: split total data (mixture with training and testing data) into 5 folder, loop over 5 folder, every time choose 1 folder as test set and 4 for train set



```python
import numpy as np
import os


np.random.seed(0)

X_train = np.genfromtxt('../data/mixture_img.csv', delimiter=',')
X_label = np.genfromtxt('../data/mixture_label.csv', delimiter=',')

n_folder = 5
interval = X_train.shape[0] / n_folder

# shuffle
indices = np.random.permutation(X_train.shape[0])
X_train = X_train[indices]
X_label = X_label[indices]

# split into n folders
for i in range(n_folder):
    # test folder
    test_folder_num = i
    test_indices = range(test_folder_num * interval, (test_folder_num + 1) * interval)
    test_image = X_train[test_indices]
    test_label = X_label[test_indices]

    # train folder
    train_image = np.delete(X_train, test_indices, axis=0)
    train_label = np.delete(X_label, test_indices)

    if os.path.isdir('5_folder_input/iter_{}'.format(i)):
        os.system("rm -rf 5_folder_input/iter_{}".format(i))
    os.mkdir('5_folder_input/iter_{}'.format(i))

    np.savetxt('5_folder_input/iter_{}/train_image.csv'.format(i), train_image, fmt='%i', delimiter=',')
    np.savetxt('5_folder_input/iter_{}/train_label.csv'.format(i), train_label, fmt='%i', delimiter=',')
    np.savetxt('5_folder_input/iter_{}/test_image.csv'.format(i), test_image, fmt='%i', delimiter=',')
    np.savetxt('5_folder_input/iter_{}/test_label.csv'.format(i), test_label, fmt='%i', delimiter=',')
~
```

run.sh: run 5 experiments for the corresponding 5 dataset (different test and train folder combination)



```bash
m=4
r=2

n=0

while [ $n -lt 5 ]; do
    hadoop fs -rm -r output
    rm -rf output
    input="5_folder_input/iter_${n}/train_image.csv"
    output="output"
    cp ./ori_centroid_points.txt ./old_centroid_points.txt
    cp ./ori_centroid_points.txt ./new_centroid_points.txt
    error=999999
    rm -rf "iter_${n}_error.txt"
    i=0

    while [ $error -ne 0 ]; do
    # while [ $i -lt 40 ]; do
        rm -rf output

        hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar \
            -D mapred.map.tasks=$m \
            -D mapred.reduce.tasks=$r \
            -D mapred.compress.map.output=ture \
            -file mapper1.py -mapper mapper1.py  \
            -file reducer1.py -reducer reducer1.py  \
            -file old_centroid_points.txt \
            -input $input \
            -output output \

        rm new_centroid_points.txt
        hadoop fs -get output ./
        hadoop fs -rm -r output
        cat output/* >> ./new_centroid_points.txt
        error="$(python cal_error.py)"
        echo "$iteration ${i}, error is ${error}" >> "iter_${n}_error.txt"
        cp new_centroid_points.txt old_centroid_points.txt

        # rm -rf output
        i=$(( i + 1 ))
    done
    cp new_centroid_points.txt "iter_${n}_centroid_points.txt"
    n=$(( n + 1 ))
done
```

The result is tested on x=100%, which is best x in part (b)

| Testing set | Classification Accuracy |
|---|---|
| Part 1 | 0.5901 |
| Part 2 | 0.5849 |
| Part 3 | 0.5959 |
| Part 4 | 0.5889 |
| Part 5 | 0.5916 |
| Average | 0.59028 |

| Testing set | Classification Accuracy |
|---|---|
| Part 1 | 0.5946 |
| Part 2 | 0.5844 |
| Part 3 | 0.5861 |
| Part 4 | 0.5874 |
| Part 5 | 0.5907 |
| Part 6 | 0.6014 |
| Part 7 | 0.5984 |
| Part 8 | 0.5819 |
| Part 9 | 0.5873 |
| Part 10 | 0.5986 |
| Average | 0.59108 |

# Q2 (a):

Input: $X$ : $N \times D$ data matrix

1. Initialize the parameters: $q_k, \pi_k$ ($Q$ is $K \times D$ matrix, $\pi$ is $K \times 1$ matrix)
   (can randomly initialize or using the centroids from k-means)

2. E Step: For each data point $x_n$, determine its assignment score to each Bernoulli k

$$\gamma\left(z_{nk}\right)=\pi_k p\left(x_n \mid q_k\right) / \left( \sum_{j=1}^{K} \pi_j p\left(x_n \mid q_j\right) \right)$$

$$p\left(x_n \mid q_k\right)=\prod_{i=1}^{D} q_{ki}{}^{x_{ni}} \left(1-q_{ki}\right)^{\left(1-x_{ni}\right)}$$

3. M Step: For each Bernoulli k, update parameters using new $\gamma\left(z_{nk}\right)$

$$q_k = \frac{\sum_{n=1}^{N} \gamma\left(z_{nk}\right) x_n}{\sum_{n=1}^{N} \gamma\left(z_{nk}\right)} \qquad \pi_k = \frac{\sum_{n=1}^{N} \gamma\left(z_{nk}\right)}{N}$$

4. Evaluate log likelihood. If likelihood or parameters converge, stop. Else go to step 2 (E step).

$$P\left(X\right)=\prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k p\left(x_n \mid q_k\right)$$

Perform probabilistic clustering:

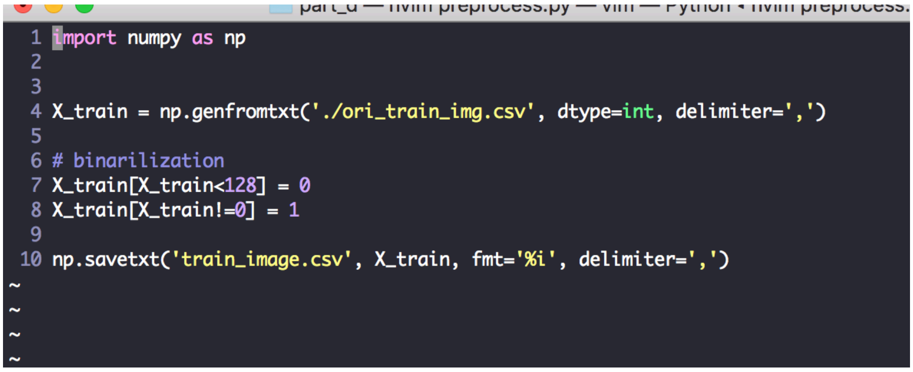
1. For each data point $x_n$, determine $p\left(x_n \mid q_k\right)$
2. Calculate $p\left(x_n, q_k\right)=p\left(x_n \mid q_k\right) \times \pi\left(k\right)$
3. Assign $x_n$ to cluster $k$ with maximum $p\left(x_n, q_k\right)$
   (Can also choose $k$ as probability distribution)

# Q2 (b):

Binarization:

Just set pixel < 128 to be 0, otherwise to be 1

code: preprocess.py

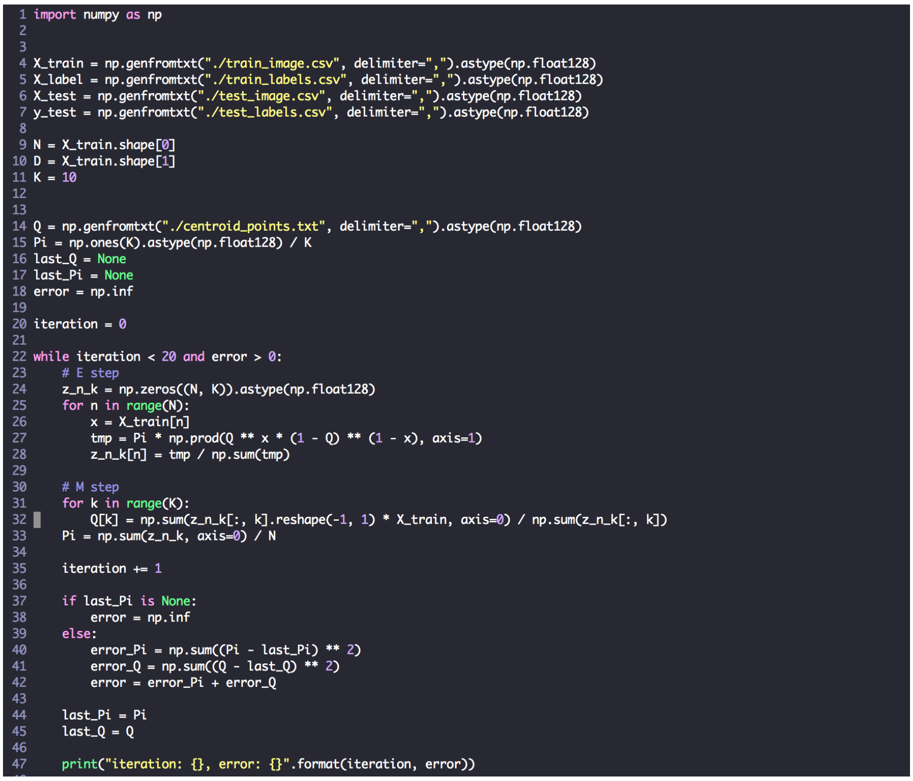

part_d — nvim preprocess.py — vim — Python ◂ nvim preprocess.

```python
import numpy as np


X_train = np.genfromtxt('./ori_train_img.csv', dtype=int, delimiter=',')

# binarilization
X_train[X_train<128] = 0
X_train[X_train!=0] = 1

np.savetxt('train_image.csv', X_train, fmt='%i', delimiter=',')
~
~
~
~
```

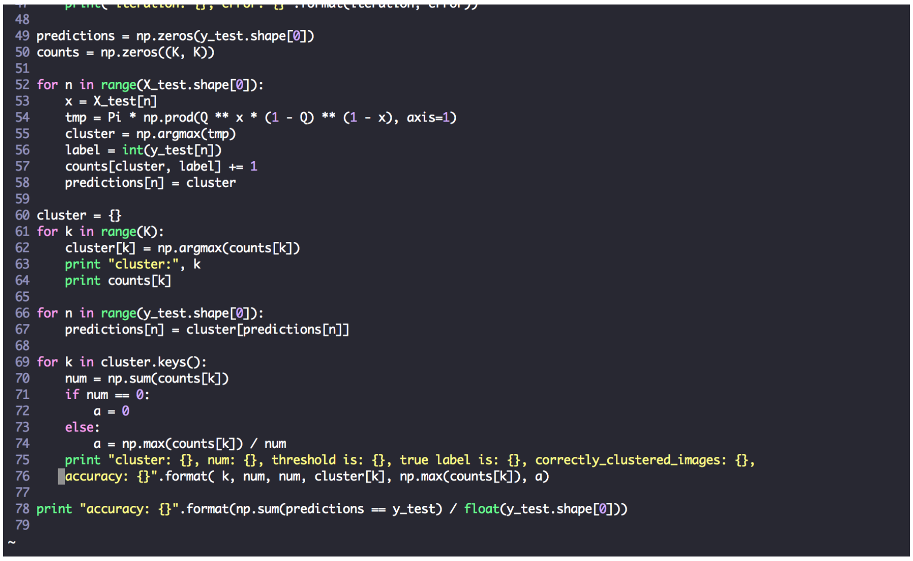
BMM and EM-algorithm:

I also use majority point method to choose true label, but just choose the label with largest number of points in a cluster to be its label.

code: BMM.py



```python
import numpy as np


X_train = np.genfromtxt("./train_image.csv", delimiter=",").astype(np.float128)
X_label = np.genfromtxt("./train_labels.csv", delimiter=",").astype(np.float128)
X_test = np.genfromtxt("./test_image.csv", delimiter=",").astype(np.float128)
y_test = np.genfromtxt("./test_labels.csv", delimiter=",").astype(np.float128)

N = X_train.shape[0]
D = X_train.shape[1]
K = 10


Q = np.genfromtxt("./centroid_points.txt", delimiter=",").astype(np.float128)
Pi = np.ones(K).astype(np.float128) / K
last_Q = None
last_Pi = None
error = np.inf

iteration = 0

while iteration < 20 and error > 0:
    # E step
    z_n_k = np.zeros((N, K)).astype(np.float128)
    for n in range(N):
        x = X_train[n]
        tmp = Pi * np.prod(Q ** x * (1 - Q) ** (1 - x), axis=1)
        z_n_k[n] = tmp / np.sum(tmp)

    # M step
    for k in range(K):
        Q[k] = np.sum(z_n_k[:, k].reshape(-1, 1) * X_train, axis=0) / np.sum(z_n_k[:, k])
    Pi = np.sum(z_n_k, axis=0) / N

    iteration += 1

    if last_Pi is None:
        error = np.inf
    else:
        error_Pi = np.sum((Pi - last_Pi) ** 2)
        error_Q = np.sum((Q - last_Q) ** 2)
        error = error_Pi + error_Q

    last_Pi = Pi
    last_Q = Q

    print("iteration: {}, error: {}".format(iteration, error))
```



```
    print("iteration: {}, error: {}".format(iteration, error))

predictions = np.zeros(y_test.shape[0])
counts = np.zeros((K, K))

for n in range(X_test.shape[0]):
    x = X_test[n]
    tmp = Pi * np.prod(Q ** x * (1 - Q) ** (1 - x), axis=1)
    cluster = np.argmax(tmp)
    label = int(y_test[n])
    counts[cluster, label] += 1
    predictions[n] = cluster

cluster = {}
for k in range(K):
    cluster[k] = np.argmax(counts[k])
    print "cluster:", k
    print counts[k]

for n in range(y_test.shape[0]):
    predictions[n] = cluster[predictions[n]]

for k in cluster.keys():
    num = np.sum(counts[k])
    if num == 0:
        a = 0
    else:
        a = np.max(counts[k]) / num
    print "cluster: {}, num: {}, threshold is: {}, true label is: {}, correctly_clustered_images: {},
    accuracy: {}".format( k, num, num, cluster[k], np.max(counts[k]), a)

print "accuracy: {}".format(np.sum(predictions == y_test) / float(y_test.shape[0]))

~
```

Results:

- I use the centroids from part a result.
- Assign cluster label that has largest number in this cluster.
- Iterations 20 times

| **Table. 1.** | **The Accuracy** | **of Clustering** | **Performance** | **with** | **x = 100%** |
|---|---|---|---|---|---|
| Cluster Number | # images in the entire cluster | # of images considered (m) when determining the cluster label | Major Label of central images | # correctly clustered images | Classification Accuracy (%) |
| 0 | 1200 | 1200 | 1 | 1042 | 0.868 |
| 1 | 1742 | 1742 | 6 | 848 | 0.487 |
| 2 | 0 | 0 | NA | NA | NA |
| 3 | 1131 | 1131 | 7 | 644 | 0.569 |
| 4 | 1606 | 1606 | 8 | 417 | 0.260 |
| 5 | 0 | 0 | NA | NA | NA |
| 6 | 0 | 0 | NA | NA | NA |
| 7 | 1825 | 1825 | 3 | 767 | 0.420 |
| 8 | 1635 | 1635 | 4 | 634 | 0.388 |
| 9 | 861 | 861 | 0 | 778 | 0.904 |
| Total Set | 10000 | 10000 | NA | 5130 | 0.5137 |

# Q3:

(a)

$$U=\begin{bmatrix} -0.31 & -0.28 \\ -0.27 & 0.22 \\ -0.23 & 0.47 \\ -0.36 & -0.47 \\ -0.70 & -0.23 \\ -0.40 & 0.62 \end{bmatrix}$$

$$S=\begin{bmatrix} 24.61 & 0 \\ 0 & 11.45 \end{bmatrix}$$

$$V^T=\begin{bmatrix} -0.27 & -0.51 & -0.37 & -0.22 & -0.43 & -0.35 & -0.41 \\ 0.69 & -0.45 & 0.42 & -0.37 & -0.03 & -0.07 & 0.01 \end{bmatrix}$$

Result is found by using “numpy package” of python

(b)

axis 1: $\begin{bmatrix} -0.27 & -0.51 & -0.37 & -0.22 & -0.43 & -0.35 & -0.41 \end{bmatrix}^T$

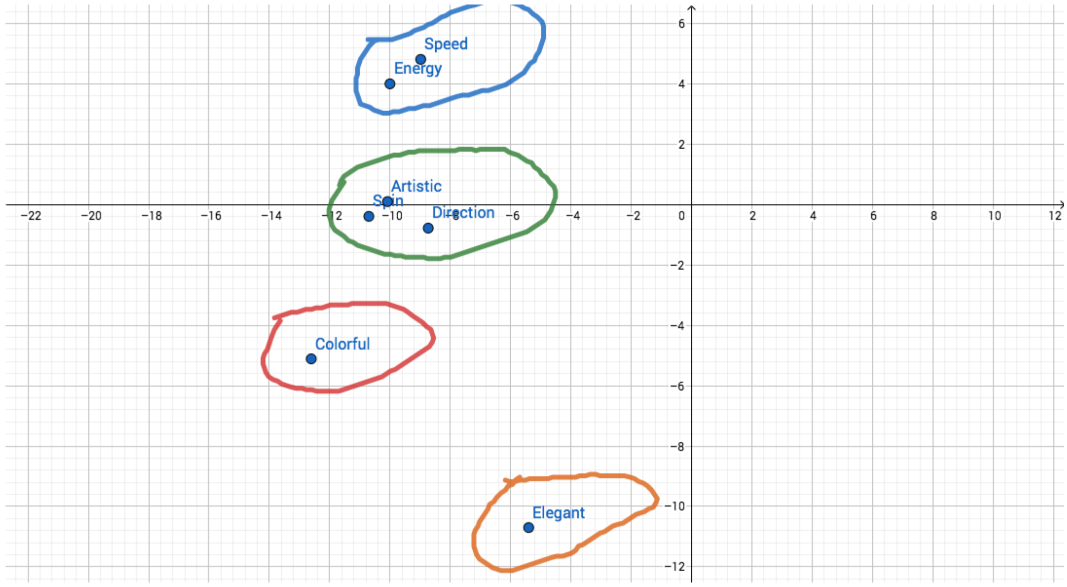
axis 2: $\begin{bmatrix} 0.69 & -0.45 & 0.42 & -0.37 & -0.03 & -0.07 & 0.01 \end{bmatrix}^T$

Coordinates of Doc under new 2-D system:

$$\begin{bmatrix} -0.27 & -0.51 & -0.37 & -0.22 & -0.43 & -0.35 & -0.41 \\ 0.69 & -0.45 & 0.42 & -0.37 & -0.03 & -0.07 & 0.01 \end{bmatrix}\begin{bmatrix} 5 \\ 0 \\ 5 \\ 0 \\ 4 \\ 1 \\ 1 \end{bmatrix}=\begin{bmatrix} -5.68 \\ 5.37 \end{bmatrix}$$

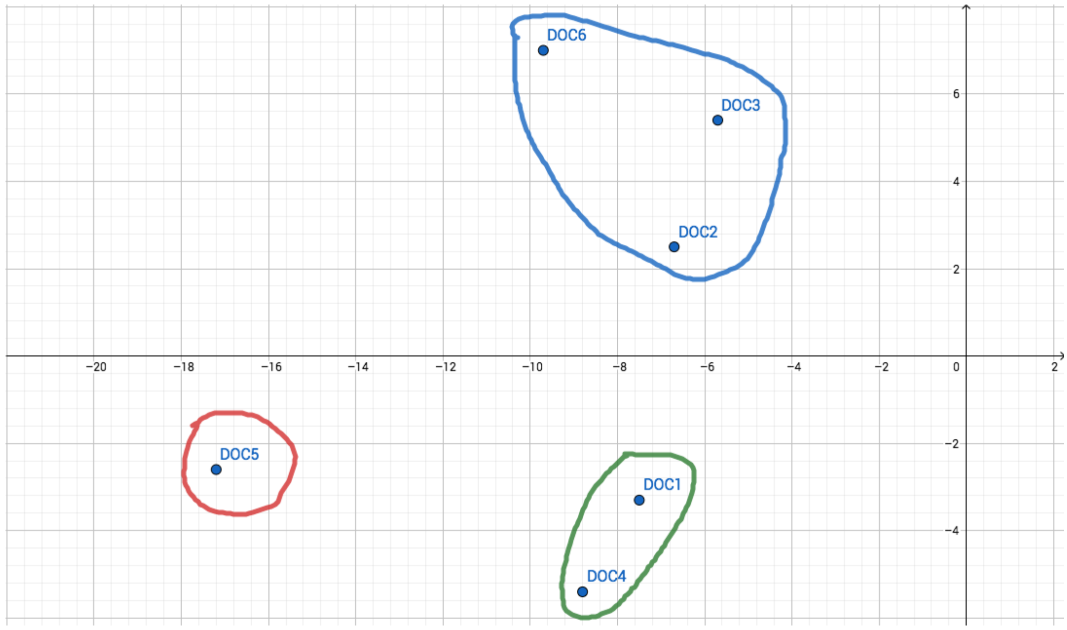
Coordinates of words:

| Energy | Colorful | Speed | Elegant | Spin | Direction | Artistic |
|---|---|---|---|---|---|---|
| [-6.74, 7.98] | [-12.61, -5.11] | [-8.98, 4.81] | [-5.4, -4.25] | [-10.7, -0.39] | [-8.73, -0.78] | [-10.08, 0.1] |



Coordinates of DOC:

| DOC1 | DOC2 | DOC3 | DOC4 | DOC5 | DOC6 |
|---|---|---|---|---|---|
| [-7.51, -3.25] | [-6.69, 2.46] | [-5.68, 5.37] | [-8.75, -5.39] | [-17.24, -2.64] | [-9.73, 7.02] |


DOC6
DOC3
DOC2
DOC5
DOC1
DOC4