Lab EXE 1: Clustering (deadline week 7)

1. Introduction

Suppose we have N n-vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. Each of them is a feature vector of an object. The goal of clustering is to group or partition N objects into k groups or clusters. The objects in a cluster should have similar feature. For example, we can use clustering algorithms to partition patients.

- Suppose $\boldsymbol{x}_i$'s are feature vectors associated with N patients admitted to a hospital, a clustering algorithm clusters the patients into k groups of similar patients.

In the lecture notes, we present the knowledge discovery of using clustering. In this experiment, we focus on data compression using clustering. We will use Matlab in our exercise. Please make sure that you have install MatLab or you know how to use online MatLab.

Algorithm 1: k-means Algorithm

Repeat until convergence

0. Set the initial code vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$

1. Partition the vectors into k groups. For each vector $i = 1, .., N$, assign to the group j if $\|\boldsymbol{x}_i - \boldsymbol{c}_j\| < \|\boldsymbol{x}_i - \boldsymbol{c}_{j'}\|$, for all j' not equal to j. That is $\boldsymbol{c}_j$ is the nearest code vector.

2. Update representatives. For each group $j = 1, .., k$, set $\boldsymbol{c}_j$ to be the mean of the vectors in group j.

$$\boldsymbol{c}_j = \frac{1}{\text{number of vectors in Group j}} \sum_{\boldsymbol{x}_i \text{ in Group j}} \boldsymbol{x}_i$$

3. Repeat 1 and 2 until converges.

2. K-means

Suppose we N n-vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. The goal of clustering is to group or partition the vectors (if possible) into k groups or clusters. Each group is represented a representative vector, or saying code vector, $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$

$$\boldsymbol{c}_j = \frac{1}{\text{number of vectors in Group j}} \sum_{\boldsymbol{x}_i \text{ in Group j}} \boldsymbol{x}_i \quad (1)$$

The k-means algorithm, shown in Algorithm 1, is the most popular clustering algorithm. It uses the current codevectors to partition the data vectors into k groups. Afterwards, the codevectors are updated based on the partitioning results.

After the clustering process, we use the mean square error (MSE) as the performance indicator, given by

$$MSE = \frac{1}{N}\sum_{j=1}^{k}\sum_{\boldsymbol{x}_i \, belongs \, to \, \boldsymbol{c}_j}\left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\|^2 . \quad (2)$$

We use the following example to illustrate the k-mean. Suppose that we have 4 vectors:

$\boldsymbol{x}_1 = \begin{bmatrix}1\\1\end{bmatrix}, \boldsymbol{x}_2 = \begin{bmatrix}1\\2\end{bmatrix}, \boldsymbol{x}_3 = \begin{bmatrix}4\\2\end{bmatrix}, \boldsymbol{x}_4 = \begin{bmatrix}3\\2\end{bmatrix}$, and the initial codevectors: $\boldsymbol{c}_1 = \begin{bmatrix}1\\1\end{bmatrix}, \boldsymbol{c}_2 = \begin{bmatrix}1\\2\end{bmatrix}$.

First round: After the first round partition, we have the partitioning result shown in Table 1. Based on the partitioning result shown in Table 1, the new cluster centers are:

$$\boldsymbol{c}_1 = \begin{bmatrix}1\\1\end{bmatrix}, \boldsymbol{c}_2 = \frac{1}{3}\left(\begin{bmatrix}1\\2\end{bmatrix} + \begin{bmatrix}4\\2\end{bmatrix} + \begin{bmatrix}3\\2\end{bmatrix}\right) = \begin{bmatrix}\frac{8}{3}\\2\end{bmatrix}. \quad (3)$$

Second round: With the new codevectors, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, we obtain the new partitioning shown in Table 2. From the table, the new cluster centers are:

$$\boldsymbol{c}_1 = \frac{1}{2}\left(\begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}1\\2\end{bmatrix}\right) = \begin{bmatrix}1\\\frac{3}{2}\end{bmatrix}, \boldsymbol{c}_2 = \frac{1}{2}\left(\begin{bmatrix}4\\2\end{bmatrix} + \begin{bmatrix}3\\2\end{bmatrix}\right) = \begin{bmatrix}\frac{7}{2}\\2\end{bmatrix}. \quad (4)$$

One can easily verify that when we repeat the update process, the codevectors are not changed anymore.

Table 1. Partition result in the first round.

Data vector	Cluster
$\boldsymbol{x}_1$	$\boldsymbol{c}_1$
$\boldsymbol{x}_2$	$\boldsymbol{c}_2$
$\boldsymbol{x}_3$	$\boldsymbol{c}_2$
$\boldsymbol{x}_4$	$\boldsymbol{c}_2$

Table 2. Partition result in the second round.

Data vector	Cluster
$\boldsymbol{x}_1$	$\boldsymbol{c}_1$
$\boldsymbol{x}_2$	$\boldsymbol{c}_1$
$\boldsymbol{x}_3$	$\boldsymbol{c}_2$
$\boldsymbol{x}_4$	$\boldsymbol{c}_2$

In the report, please work out the following case by hand:

Question 1:

$\boldsymbol{x_1} = \begin{bmatrix}1\\2\end{bmatrix}, \boldsymbol{x}_2 = \begin{bmatrix}2\\2\end{bmatrix}, \boldsymbol{x}_3 = \begin{bmatrix}3\\2\end{bmatrix}, \boldsymbol{x}_4 = \begin{bmatrix}4\\4\end{bmatrix}$. Suppose we choose the initial codevectors as as $\boldsymbol{c}_1 = \begin{bmatrix}3\\3\end{bmatrix}, \boldsymbol{c}_2 = \begin{bmatrix}4\\4\end{bmatrix}$.

(a) What are the codevectors and the MSE after the k-means the process?

(b) Perform Part (a) one more time with the initial codevectors as $\boldsymbol{c}_1 = \begin{bmatrix}1\\2\end{bmatrix}, \boldsymbol{c}_2 = \begin{bmatrix}2\\2\end{bmatrix}$.

(c) Discuss the results of Part (a) and Part (b).

3. **Data Compression**

Suppose we have many data n-vectors: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$. If we use 8-bit to store each element in a vector, **the total storage requirement** is $8nN$ bits

Consider that we perform the k-means clustering on the data vectors, where $k = 2^v$. After clustering, each data vector $\boldsymbol{x}_i$is assigned to one of codevectors $\boldsymbol{c}_j$. **The idea is to use $\boldsymbol{c}_j$ to replace $\boldsymbol{x}_i$**, i.e., $\widehat{\boldsymbol{x}}_i = \boldsymbol{c}_j$**, where** $\widehat{\boldsymbol{x}}_i$ is the reconstruction of $\boldsymbol{x}_i$.

Instead of storing **the quantized** $\widehat{\boldsymbol{x}}_i$ directly, we store the index j of the codevector $\boldsymbol{c}_j$ **(that is the codevector assigned to** $\boldsymbol{x}_i$ 0. Of course, we need to store all the codevectors : $\boldsymbol{c}_1, \dots, \boldsymbol{c}_k$, first.

Example:

If $\boldsymbol{x}_i$ belongs to $\boldsymbol{c}_5$, we do not store $\boldsymbol{x}_i$, we store the index "5", denoted as $\theta_i = 5$.

In the reconstruction, we need to pack up $\boldsymbol{c}_{\theta_i}$ as the reconstruction $\widehat{\boldsymbol{x}}_i$.

In addition, **The distortion for this case is** $\|\boldsymbol{x}_1 - \boldsymbol{c}_5\|^2$.

In general case, the MSE for this replacement is given by

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2 = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{x}_i - \boldsymbol{c}_{\theta_i}\right\|^2. \quad (5)$$

In data compression, we store all codevectors (calling codebook) and the cluster indices θ_i's of the data vectors in our system. When we process the data, we use indices θ_i's and the codebook to restore $\widehat{\boldsymbol{x}}_i's$. For example, if $\theta_7 = 6$, then the reconstruction $\widehat{\boldsymbol{x}}_7 = \boldsymbol{c}_6$.

With the above scheme, the storage requirement for the codebook is $8nk$ bits. The storage requirement for the indices θ_i's are $N \log_2 k$ bits. Hence the total storage requirement is

$(8nk + N \log_2 k)$ bits.

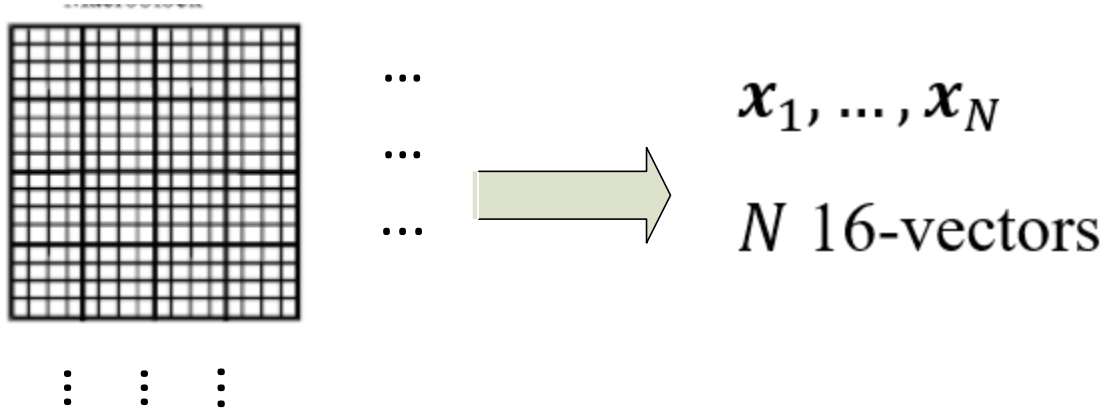

Figure 1. Illustration of decomposing an image into a number of 16-vectors.

4. k-means for image compression

In this experiment, we consider image compression. An image with $M \times M$ pixels can be considered a matrix object. Each element of the matrix represents the intensity of a pixel.

As shown in Figure 1, we decompose an 512x512 image into a number of non-overlapping 4x4 blocks. Each block is represented by a 16-vector. After the decomposition process, the image is reformulated as a number of data vectors, we can perform the k-means compression process mentioned in Section 3.

Question 2

If the resolution of the image 512x512 and the data vectors are 16×1 . How many data vectors we have ? Explain.

Question 3

If the resolution of the image 256x256 and the data vectors are 16×1 . How many data vectors we have ? Explain.

Question 3

In clustering for data compression, we usually set $k = 2^v$, where v is a positive integer. Why?

Question 4

Is it good to increase the value of k to 2,048? Explain.

5. Our matlab files and data file.

There are three image files: woman.gif, bird.gif and house.gif. Each of them stores the image in the gif format.

Also, **there are three M-files used in this experiments**. They **`lab1test.m`, `im_to_data.m`,** and **`data_to_im.m`.** You must down those files into your MatLab working directory.

Table 3 shows the matlab program (**`lab1test.m`**) used in this experiment. Besides, there are two more functions (m-files). One is **`im_to_data.m`** that convert an image into a data matrix. Each row is of the data matrix is a 16-vector (data vector). Each row of the data matrix is a 16-vector (data vector). Another is **`data_to_im.m`** that convert the data matrix into an image. Table 3 shows the sufficient information for you to perform the experiments.

After running the program, the Matlab displays the MSE value in the command window, as shown in Figure 2. Also, there are two figure windows to show the original image and the reconstructed image. You can save the images by clicking file manual bar in the figure windows. Choose "save as" and select the file format "jpg".

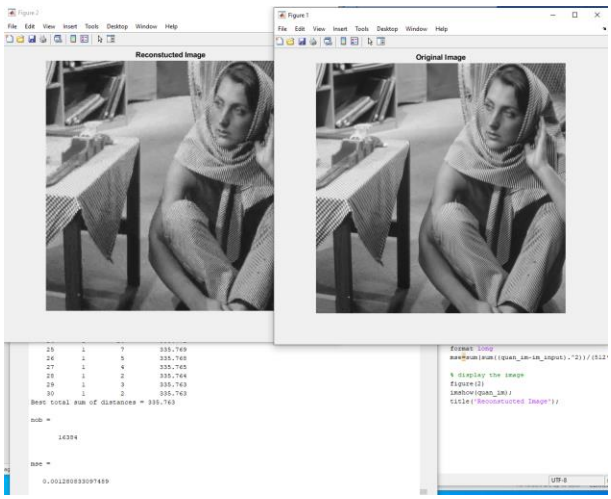

Figure 2. After running program, you can get the MSE value in the command window and two images in the figure windows.

6. The experiment Steps.

- Perform k-means for the image "woman" with k=16, 32, 64, 128, 256, 512.
- Record all the MSE values and summarizes the MSE values in a table.
- Examine the image quality of the reconstruction images.
- Save some meaningful images that are important for discussion in the report.
- Inspect the reconstruction images, discuss the visual artifacts created the compression process.

7. The report

1. Part one: Answer Questions 1-4.
2. Part two: Present your results and discuss your observation.
3. **If you want to get a high mark, you should include your results for all test images, and discuss the advantages and disadvantages of using clustering for compression.**

Hint: MSE values versus *k*, the storage requirement for each setting, and visual quality.

Table 3. The program used in this experiment.

```
A = imread('woman','gif');
im_input=double(A)/255; % normailize the image
figure(1);
imshow(im_input); % display the image
title('Original Image');
bs=4; % Block Size (4x4)
nob=512*512/16 %
% convert an 512x512 image to a data matrix
% data matrix is 16xnob;
% each row vector is a data vector
data_mat=im_to_data(im_input);
% Perform k-means with the maximum iteration is 200
% k=256
% Afterwords, indx is an array to store cluster index of each data vector
% codevec is 16xk array. Each row is a codevector

k=128 % number of codevectors
[indx codevec]=kmeans(data_mat,k,'Display','iter','Maxiter',200);

% reconstruct the data matrix
quan_data=codevec(indx,:);

% convert data matrix to an image
quan_im=data_to_im(quan_data);

% compute the MSE
format long
mse=sum(sum((quan_im-im_input).^2))/(512*512)

% display the image
figure(2)
imshow(quan_im);
title('Reconstucted Image');
```