ELE888/EE8209 Intelligent Systems
Lab 4: Unsupervised Learning

Objective

To implement the K-means algorithm for clustering unlabeled data.

Background

In unsupervised learning, natural clusters within unlabeled data samples (i.e. with no categorical information) may be identified using an iterative learning process. When the functional form of the underlying probability densities of the data are assumed to be known, the only thing that must be learnt is the value of an unknown parameter vector. One elementary but popular approximate method that performs the above is the k-means clustering algorithm. The goal of the k-means clustering algorithm is to identify k mean vectors or cluster *centres* within the given unlabeled data.

In the k-means clustering algorithm, we begin with randomly initializing the mean vectors (k cluster centers) and then assigning the data points to the nearest cluster by computing the Euclidean distance. Once all the data points are assigned to one of the k clusters, the mean vectors of the k clusters are recomputed. The process is repeated until there is no change observed in the recomputed mean vectors of the k clusters. The following pseudo code describes the k-means clustering algorithm:

K-means algorithm:

```
begin initialize $n$, $c = k$, $\boldsymbol{\mu_1}$ ,$\boldsymbol{\mu_2}$ ,. . . , $\boldsymbol{\mu_c}$
do classify $n$ samples according to nearest $\boldsymbol{\mu_i}$
recompute $\boldsymbol{\mu_i}$
until no change in $\boldsymbol{\mu_i}$
```

5. return $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$, . . . , $\boldsymbol{\mu_c}$

6. end

Note: In the real algorithm design, for the pseudocode number 4, you need to use two stopping criteria (this is similar to the MNN stopping criteria):

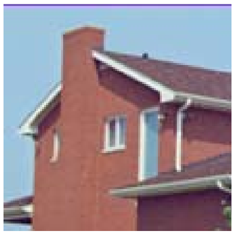
1. $\bigtriangledown\boldsymbol{\mu}$, which consists $[\bigtriangledown\boldsymbol{\mu_1}, \bigtriangledown\boldsymbol{\mu_2}, ... \bigtriangledown\boldsymbol{\mu_c}]$. You can choose your own distance measurement. For instance, sum of Euclidian norms of $\bigtriangledown\boldsymbol{\mu_1}$, $\bigtriangledown\boldsymbol{\mu_2}$, ... $\bigtriangledown\boldsymbol{\mu_c}$.
2. Maximum iteration number (your own choice).

Finding Dominant Colors in an Image:

One application in computer vision tasks is to identify dominant colors present in an image/video sequence. This can be used for matching similar images, or as a precursor for compression, object detection, etc.

In this lab, you are required to use k-means to *'discover'* the k most dominant colors from the sample image: *'house.tiff'*. The set of data samples is the set of pixels, where each pixel is represented by a vector representing the color content of the pixel. The objective is to group the set of pixel vectors (p_i) into k clusters, where each cluster can be described by a mean pixel vector.

To load an image into matlab and display it, use the following commands:
$\gg I = imread('filename.xxx');$
$\gg figure, imshow(I);$



In an $M \times N$ color image, each pixel (p_i) is a 3D vector p_i=$[r_i$,g_i,$b_i]^T$. This vector represents a color as a mixture of red (r_i), green (g_i) and blue (b_i). Each color channel is thus treated as a feature, and each pixel as a sample.

To identify clusters present in the color space of each image, the pixel data will need to be transformed into a 2D array of input samples. This can be done using the 'reshape' command:
$\gg X = reshape(I, M * N, 3);$
$\gg X = double(X);$

Once the pixels have been transformed into an array of input samples (X), they may be plotted in RGB space using a pre-specified color [$R\ G\ B$]:
$\gg figure,\ plot3(X(:,1), X(:,2), X(:,3),$ '.', 'Color',[$R\ G\ B$])

A version of X after clustering and labeling ($X_{labeled}$) can be reformed into an image for display:
$\gg I_{labeled} = reshape(X_{labeled}, M, N, 3);$
$\gg figure,\ imshow(\ uint8(\ I_{labeled}\ )\ )$

Laboratory Exercises

(a) Let $c = 2$, and run the k-means algorithm using your own initial state for the two means. Keep a record of the initial means you use, and show:
i. A plot of the Error Criterion J

$$J = \sum_{j=1}^{c} \sum_{k=1}^{n_j} \|\mathbf{x}_k - \boldsymbol{\mu_j}\|^2.$$

ii. A plot of the cluster means for at least two stages of the clustering process, and their final values.
iii. A plot of the labeled data samples (pixels) in RGB space (i.e. for each data sample, find out which cluster it belongs to, and plot it using a color unique to that cluster - e.g. you could use the mean vectors, as they represent a unique RGB value.
iv. Plot the image in labeled form (you may use artificial coloring e.g. black/white, or the mean colors discovered by your algorithm).

(b) Let $c = 5$, now perform two independant runs of the k-means algorithm, each beginning with a different initial state for the means. Record these initial states, plot the labeled clusters in RGB space for each run and comment on any discrepancies.

(c) Use the Xie-Beni (XB) index to assess the quality of the two clustering solutions found in part (b), and comment/justify any differences. The formula for XB is given:

$$XB(c) = \frac{1}{N} \cdot \sum_{k=1}^{N} \sum_{j=1}^{c} \frac{\mathbf{u}_{jk}\|\mathbf{x}_k - \boldsymbol{\mu_j}\|}{min\|\boldsymbol{\mu_i} - \boldsymbol{\mu_j}\|}$$

$$\text{where } \mathbf{u}_{jk} = \hat{P}(\omega_j|\mathbf{x}_k) = \{ \begin{array}{ll} 1 & \text{if } x_k \in j^{th} \text{ cluster} \\ 0 & \text{if } x_k \notin j^{th} \text{ cluster} \end{array}$$

Note, $\boldsymbol{\mu_i}$ is the mean value of any classes other than $\boldsymbol{\mu_j}$

Due date

The week in April 10. Reports should include all plots, figures and discussions