EE5177: Machine Learning for Computer Vision

Programming Assignment - 2 : Modeling data densities using EM algorithm

Due date: March 5th 2017, 11:55pm IST

Note:

1. For any questions, please schedule a time with TAs before deadline according to their convenience. Please use moodle dicussion threads for posting your doubts and also check it before mailing to TAs, if the same question has been asked earlier.
   (a) Problem - 1 and 2: Anil Kumar, email: ee15s055@ee.iitm.ac.in
2. Submit a single zip file in the moodle named as PA1_Rollno.zip containing the report and the folders containing corresponding codes. Please submit in **ZIP format only not RAR** or others.
3. Read the problem fully to understand the whole procedure.
4. Late submissions will be evaluated for reduced marks and for each day after the deadline we will reduce the weightage by 10%.

Problem-1 [15 marks]: In this assignment we will be doing face detection by fitting the Gaussian Mixture Models (GMMs) using Expectation Maximization (EM) algorithm. The special case of this is the PA1 Q2 where you were asked to fit the single Gaussian distribution instead. For this assignment we will closely follow the similar experiment in section 7.9.1 in the text book (http://computervisionmodels.com/), please go through it before proceeding further.

Data description: provided data.zip file has train and test mat files of face and non-face dataset. All the images are of size 22x22 and are arranged along rows in the mat files. You have 800 training samples for both the classes.

1. To start with, we will fit the GMM models directly on the training data without any preprocessing. Since we are not using any preprocessing, the estimated covariance will be ill-conditioned given the dimensionality of the data. To avoid this we will make simplifying assumptions:

   - We will restrict the covariance matrix to be diagonal so we will need to estimate only n parameters for each component instead of $O(n^2)$.
   - To avoid ill-conditioning nature of covariance matrix we will add a small value (0.01) to the diagonal elements of covariance matrix.

   Experiment: Using the above assumptions fit GMM models for face and non-face data using EM algorithm and classify the test data using the criterion mentioned in the text book (Eqn. 7.53). Do this with number of components in GMM as 1, 2 and 3 and report the overall accuracies. In all these cases try to visualize the mean of both classes as gray scale image. In addition, for the case of one component visualize the diagonal covariance. If we increase the number of components beyond 3 the covariance matrix can't be estimated properly and accuracy won't improve.

   Questions

   (a) Comment on what are the means of components in GMM modeling for face and non-face data.

   (b) Visaulize the diagonal covariance of face and nonface in case of a single component GMM and comment on its variation.

Note: To fit the Gaussian distribution use built-in command *gmdistribution.fit* in matlab and use *pdf* to evaluate the likelihood. Also, don't get mislead by high accuracies this dataset is very restrictive in terms of aligned faces and there is no pose variation.

Problem-2: **[20 marks]** In this problem, we will study the K-means clustering algorithm. Given a set of data points $\{\mathbf{x}_j\}_{j=1}^P$, K-means clustering aims to assign every point to one of the clusters $\mathcal{C}_i$ so as to minimize the sum of distance of each point to its cluster centre $\boldsymbol{\mu}_i$. We will take the 3D RGB colour value as our data point based on which we segment an image by assigning each of its pixels to different clusters. In our case, $\{\mathbf{x}_j\}_{j=1}^P$ is the set of all pixel colour values.

Steps:

0. Fix the desired number of clusters K.

1. Initialize random values (picked from the input set) to the mean of the K clusters, $\{\boldsymbol{\mu}_i^{(1)}\}_{i=1}^K$, where $\boldsymbol{\mu}_i^{(1)} \in [0-255]^3$ and the superscript (t) denotes tth iteration.

2. Assign every pixel $\mathbf{x}_j$ of the image to the cluster which has its mean located closer to this point compared to the means of other clusters, i.e. each cluster assignment is as follows:

$$\mathcal{C}_i^{(t)} = \{\mathbf{x}_j : \|\mathbf{x}_j - \boldsymbol{\mu}_i^{(t)}\|_2 \leq \|\mathbf{x}_j - \boldsymbol{\mu}_k^{(t)}\|_2, \forall k, k \neq i\} \tag{1}$$

   The cost at this iteration is given by

$$d^{(t)} = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in \mathcal{C}_i^{(t)}} \|\mathbf{x}_j - \boldsymbol{\mu}_i^{(t)}\|_2. \tag{2}$$

3. Generate and store a segmented image by assigning the colour value of each pixel as the mean value of the cluster it is assigned to. Store the number of changed assignments (i.e. the number of pixels which are assigned to one cluster in the previous iteration and a different cluster in this iteration) in the variable $s^{(t)}$. (For $t = 1$, let $s^{(t)}$ = number of image pixels.) If $s^{(t)} == 0$ or $t == 100$, stop.

4. Update the mean of all clusters using the newly assigned colour values.

$$\boldsymbol{\mu}_i^{(t+1)} = \frac{1}{|\mathcal{C}_i^{(t)}|} \sum_{\mathbf{x}_j \in \mathcal{C}_i^{(t)}} \mathbf{x}_j \tag{3}$$

5. Repeat 2 to 4.

Questions:

(a) [**4 marks**] We will segment an image of a colour spectrum `spectrum.png` in which the colours vary in the horizontal direction from red to violet. Fix $K = 7$. Pick the initial mean values from pixels equally spaced in the horizontal direction in the centre row of the image. Show the segmented image.

(b) [**5 marks**] Repeat (a) with the initial mean values randomly picked from the input image itself. Show the segmented images with two different random initializations. Compare with the segmented image from (a). Comment on the results.

(c) [**10 marks**] Run K-means on `rio.png` with $K = 2, 4, 8$ and with random initial means. Show the final segmented image, and plot $d^{(t)}$ vs. t and $s^{(t)}$ vs. t for each of these Ks. Show the segmented images for iterations $t < 11$ for the cases $K = 2$ and 4.

(d) [**1 mark**] Choose one: K-means algorithm [guarantees global minimum/ guarantees only a local minimum/ does not guarantee a minimum].

–end–