CSci 5521 Homework 3
Due: Friday 09 November 2018 at 11:59 PM CST

- This assignment should be done individually, unless you decide to tackle the "Extra Item", in which case you can do this in pairs.
- You may use the built-in MATLAB functions: `mean`, `cov`, `inv`, `det`, and `eig`, `svd`, `pca`, but no built-in clustering functions. If you have doubts about a function, ask Professor Boley or a TA.

The `Digits089.csv` dataset consists of 3000 data points, each representing an image of the three digits 0, 8, 9. Each row in the file is one data sample consisting of 786 numbers in "CSV" format:

flag	label	. . . 784 pixel values . . .
(1,...,5)	(0,8,9)	(values 0,...,255)

Separate the data into three parallel arrays one with the flag values, one with the labels, and one with the pixel values representing 28×28 images of the digits `0, 8, 9`. Use the entire data set together for this assginment. Use the entries with flags `1, 2, 3, 4` as training data and samples with flag value `5` as a test set.

There are also two color images in which you will cluster the colors.

1. Apply PCA to the digits data set to reduce to dimensions needed to capture 90% of the variance.
2. Write your own K-means algorithm and apply it to the Digits data set, after reducing the dimension using the PCA in the previous step. Use $k = 6$ clusters and initial centers equal to the elements # 1, 1000, 1001, 2000, 2001, 3000 in the original data set. Print out a confusion matrix showing how many `0`'s, `8`'s, `9`'s there are in each cluster. If there are $k = 6$ clusters, this matrix should be 6×3. If each cluster were assigned a class based on the majority label among members of the cluster, what would be error rate be?
3. Repeat the above, but start by using only 2 principal components, followed by $k = 6$ clusters. Initialize K-means using the same 6 data samples (projected onto the first two principal components).
4. Apply the k-means algorithm to the colored pixel values in image `goldy.ppm` and `stadium.ppm`. The data in this case are the RGB pixel values: points in $\mathbb{R}^3$. Try $k = 3, 4, 7$. Replace each pixel RGB contents with its corresponding cluster centroid, and re-form the image using the newly substituted pixel values. Redraw the resulting pictures using the modified pixel values. In Matlab, you can read in the picture using the `imread` function, and display it with `imagesc`. You can use a combination of `reshape`, `permute` to map the 3D array to a $n \times 3$ array of pixel values (where n is the number of pixels in the image), and back again.
5. (Extra Item – required if you do this in groups of 2) Implement a Gaussian-mixture model for a Expectation-Maximzation algorithm and apply it to the image data. The initial means should be set to the means resulting from the K-means process in the previous question. Use a spherical gaussian (Covariance matrix is a multiple of the Identity: $\sigma^2 I$), and initialize the σ's to $1/n \times$ the total within-cluster scatter obtained upon the completion of the K-means process. Use $k = 4$ gaussians.

The EM algorithm yields updated values for the parameters μ_j, σ_j, and the priors, as well as the set of posterior probabilities $\Pr(\text{cluster}_j \mid \mathbf{x}_i)$ and joint probabilities $\Pr(\text{cluster}_j \ \& \ \mathbf{x}_i)$ for each point and cluster. There are also the joint counts of number of samples in cluster_j & class_i, $j = 1, \ldots, 6$, $i = 0, 8, 9$. The modified confusion matrix is 6×3 whose $j - i$-th element is the joint count (cluster_j & class_i).

Replace each pixel $\mathbf{x}_i$ in the image with a weighted sum of the centers, weighted by the computed affinity (posterior probability), and redraw the picture. $\Pr(\text{cluster}_j \mid \mathbf{x}_i)$.

Instructions

Follow the rules strictly. All code must be written in MATLAB. If we cannot run your code, you get 0 points.

- **Things to submit**

  1. `hw3_sol.pdf`: A document which contains the solutions. The front page of the PDF file should have names and UMN email addresses of the student(s) submitting the document. Also include the summary of results, like the confusion matrices resulting from the clustering vs the true labels.

  The following is to be zipped into a single ZIP file:

  2. `DoKmeans.m` starting with
     `function [assignments, centers,StepCount]=DoKmeans(data,InitialCenters);`
     carries out the k-means algorithm (coded by you). The `assignments` has the cluster assignment for eash data sample in `data`. `StepCount` holds the number of k-means iterations to convergence.
  3. `GetConfusion.m` starting with
     `function [ConfusionMatrix]=GetConfusionMatrix(TrueLabels,Assignments);`
     which takes two set of labellings for all the data and returns the corresponding confusion matrix.
  4. `DoEM.m` (assuming you answer this question) starting with
     ```
     function [NewMus,NewSigmas,NewPriors,JointProbs,Posteriors] ...
       = DoEM(Xvec,InitMus,InitSigmas,InitPriors)
     ```
     where `Xvec` is an $n \times d$ matrix of data, `InitMus` is a $k \times d$ matrix of initial centers, `InitSigmas` is a k-vector of initial standard deviations, `InitPriors` is a k-vector of initial priors (all equal to $1/k$). The outputs `NewMus`, `NewSigmas`, `NewPriors` should have the same shape as the corresponding inputs. The outputs should also include `JointProbs`, `Posteriors`, which are both $n \times k$ matrices.
  5. Any other files, except the data, which are necessary for your code.

- We may test these matlab functions you submit on our own test data.