Machine Learning
Homework Assignment 3
Due Date: Sept 30th

Attempt all questions. Each question is 10 points.

1. **[10 points]** Lets say we have a **THREE**-dimensional real valued data set with **N** point

$$\mathbf{X} = \left\{ \mathbf{x}^{(n)} = \left(x_1^{(n)}, x_2^{(n)}, x_3^{(n)} \right) \right\}_{n=1}^{N}$$

Lets say instead of doing K-**Means** clustering we want to do K-**Lines** clustering where each data point is associated with a **LINE** and not a cluster **MEAN**. Parameters associated with the k^th line are: $\mathbf{m}_k = \left(w_0^{(k)},\ w_1^{(k)}, w_2^{(k)} \right)$ representing a line of the form:

$$X_3 = w_0 + w_1 X_1 + w_2 X_2$$

"**Distance**" between a cluster "center" (i.e. a line) and a data point is given by:

$$\Delta\left(\mathbf{x}^{(n)}, \mathbf{m}_k \right) = \left(x_3^{(n)} - \left(w_0^{(k)} + w_1^{(k)} x_1^{(n)} + w_2^{(k)} x_2^{(n)} \right) \right)^2$$

Given the ASSOCIATION paramteres in any iteration:

$$\delta_{n,k} \in \{0,1\}$$

Derive an expression for the update rule for all the three parameters:

$$w_0^{(k)} = ??$$
$$w_1^{(k)} = ??$$
$$w_2^{(k)} = ??$$

2. **[10 points] SOFT K-Means Clustering:**

   (a) Write a Soft K-Means Clustering where the **DEGREE** of association of a data point to all the clusters in each iteration is given by the following:

$$\delta_{n,k}^{(t)} = \exp\left(-\left(\frac{\left\| \mathbf{x}_n - \mathbf{m}_k^{(t)} \right\|}{\sigma(t)} \right)^2 \right)$$

$$\sigma(t) = \lambda \times \sigma(t-1) = \lambda^t \sigma(0)$$

Here: $\lambda \le 1 = \text{Decay constant}$, $\sigma(0) = \text{Initial variance}$ are the ***hyper-parameters*** controlling the degree of assignment in each iteration. We start with a high value of variance and slowly as clusters converge we decrease this.

   (b) Sample 100 points from each class in the **MNIST data**. Take the PCA of these 1000

Points into a **50 dimensional space** and do the above soft clustering on them.

(c) Using different values of the two hyper-parameters plot the convergence of the SOFT K-MEANS clustering by measuring the **clustering cost** vs. **iterations** and plotting them (x-axis = iteration, y-axis = clustering cost). Make observations about the effect of the decay constant on convergence rate.

3. [10 points] Agglomerative Clustering on Newsgroup20 data

(a) Find the frequency of all words in the Newsgroup20 data. Pick the top 1000 words and ignore all the remaining words.

(b) Sample 25 documents from each of the 20 classes in the Newsgroup data. Represent each of the 25 x 20 = 500 documents as bag of words in terms of the top 1000 words selected in (a)

(c) Convert each of the 500 documents in the 1000 word space into TFIDF where IDF is computed based on (a).

(d) Using cosine similarity as a similarity between documents, do agglomerative clustering using the FOUR methods.

(e) Break the four dendograms at 10, 20, 30, 40, and 50 clusters. Measure the purity of each clustering for the four methods and plot on the x-axis the cluster size and y-axis the purity of the cluster. Compare the four methods (min, max, avg, mean link).

4. [10 points] K-Nearest Neighbor Classifier on Newsgroup20 data

Using the data sampled in 3(b) perform a K-nearest neighbor classifier. You may use the similarity matrix created for 3(d) above. Use 50% data points in each class as TRAINING and 50% as TESTING data. So for each of the TEST examples, using a certain K and the Similarity with the nearest K points, compute its class label. Plot the accuracy of the K-NN classifier for K = 1, 3, 5, 7, 9, 11, 13, 15 (x-axis is K, y-axis is TEST set accuracy).

5. [10 points] Naïve Bayes Classifier on MUSHROOM data

Build the Naïve Bayes Classifier for the **MUSHROOM** data. Divide the data into 60% training and 40% test examples. Using the 60% training examples, learn the parameters for the Naïve Bayes classifier and use this model to predict the class of the TEST set. Submit the code as well as the TEST accuracy of the NB classifier.

6. [10 points] Naïve Bayes Classifier on Newsgroup20 data

Using the top 10000 words in the Newsgroup20 data, build a Naïve Bayes classifier. Divide the data into 60% training and 40% test (per class). Report the accuracy on the test set and submit the code as well.

7. [10 points] Bayesian Classifier on MNIST data

Build a Bayesian classifier on MNIST data as follows:

(a) Treat 50% data as Training and 50% data as Testing
(b) For the 50% training data, first take the PCA projection onto top 9 dimensions
(c) Model class conditional probability of each class $p(\mathbf{x}|c)$ using a uni-modal gaussian in this 9-dimensional space.
(d) Train the classifiers (learn the mean and covariance parameters of each class using the FULL covariance matrix.
(e) Evaluate the this model and report TEST set accuracy.

(f) Also do a FISHER transform of the raw data into the top 9 dimensions.
(g) Again, compute the parameters (mean and full covariance matrix) for each of the 10 classes. Use this to predict the test accuracy.
(h) Do you notice any difference in performance? In one case we projected on the top 9 dimensions of the FULL data PCA (without taking class labeles into account). In second case we project on the top 9 FISHER dimensions.

8. [10 points] Mixture-of-Gaussians on MNIST data
In problem 8 we considered a single Gaussian in each class. Now we will consider 3 Gaussians per class. For each class data do the following:
(a) Take the data in the original space 28 x 28 (no PCA or Fisher).
(b) Learn for data in each class a 3-mixture model. Using this represent the probability of data | class instead of a single Gaussian.
(c) Compare the performance of this model with 3-mixture of Gaussians per class with 1 Gaussian per class. Also draw the MEAN vector of the 3 sub-clusters within each Mixture of Gaussians for each class.

9 [10 points] Consider a 4 dimensional data. Prove that if the non-diagonals in the co-variance matrices o each class are zeros, then the **independence assumption** holds and that the probability p(**x** | c) = p(x_1 | c) * p(x_2 | c) * p(x_3 | c) * p(x_4 | c).

10 [10 points] Consider the UNSUPERVISED problem of estimating the DENSITY DISTRIBUTION of a dataset. The entire data is represented by a single Gaussian with full co-variance matrix. We define the **Maximum Likelihood objective function** for this problem as the **product of the probability density of each data point**. Instead of using the product we take the LOG LIKELIHOOD of the data by taking the log of the product.

$$\log J = \log \prod_{n=1}^{N} P\left(\mathbf{x}^{(n)}\right)$$

$$P\left(\mathbf{x}^{(n)}\right) = N\left(\mathbf{x}^{(n)}, \mathbf{m}, \Sigma\right)$$

Maximize this log likelihood and compute the **mean** and **covariance** matrix parameters obtained by solving this maximization problem.