Instructions

- If you have any question about problems mail to mlta40717@gmail.com with the subject in the format: QHW(homework number)-P(question number) (e.g., QHW(4)-P(2)).
- Send your assignments with subject HW(homework number)-(Student-ID) to mlta40717@gmail.com.

Problem 1: EM

1.1. (12 Pts) EM & Bayesian Networks
Consider the Bayesian network shown in Figure 1 that contains five Boolean variables (below, we use the first letter of the variable names). Suppose that the variables $B,E,J,M$ are observed and $A$ is unobserved. Given a set of $n$ training examples $\{(b^{(i)}, e^{(i)}, j^{(i)}, m^{(i)})\}_{i=1}^{n}$. Show that in the M-step of the EM algorithm $\theta_{a|kl}$ denoting $P(A = a|B = k, E = l)$ is updated as

$$
\theta_{a|kl} \leftarrow \frac{\sum_{i=1}^{n} I(b^{(i)}=k, e^{(i)}=l) P(a^{(i)}=a|b^{(i)}, e^{(i)}, j^{(i)}, m^{(i)}, \theta)}{\sum_{i=1}^{n} I(b^{(i)}=k, e^{(i)}=l)}.
$$

Indeed, this update rule corresponds to the M-step $\theta \leftarrow \arg\max_{\theta} \mathbb{E}_{P(a^{(i)}|b^{(i)}, e^{(i)}, j^{(i)}, m^{(i)}, \theta)} \{ \log P(a^{(i)}, b^{(i)}, e^{(i)}, j^{(i)}, m^{(i)}|\theta) \}$.

![Figure 1](image_url)

1.2. Practical EM & GMM
Please download "EM.zip" and use this implementation of the EM algorithm estimating the parameters of a Gaussian mixture. Download "clust_data.m" data set (containing the first two features of the iris data set from the UCI repository). Below, you will find clusters of this data set using the EM algorithm.

a. [5 Pts] The implementation of GMEM uses k-means clustering algorithm for parameter initialization. Find the clustering results (for two clusters) on the above data set. Plot the cluster centroids and around each centroid draw an ellipsoid showing the respective covariance.

b. [4 Pts] Plot also the clustering result of the k-means algorithm (used in the initialization phase of EM) and compare it with the result obtained in Part (a).

c. [4 Pts] Change GMEMInitialize code to use the following parameter initialization (instead of using k-means for parameter initialization):

$$
\mu_1 = [6 \ 3]^T, \mu_2 = [8 \ 4]^T, \Sigma_1 = \Sigma_2 = I, \pi_1 = \pi_2 = 0.5.
$$

Again plot the results as explained above.

d. [5 Pts] In general for the EM algorithm (with GMM assumption), which properties of datasets can cause it to get stuck in local minima more seriously?
Problem 2: k-NN

2.1. [10 Pts] Implement a k-NN classifier as a Matlab function \( \text{foundY} = \text{kNN}(k, \text{trainX}, \text{trainY}, X) \).
Therefore, k-NN must find the labels of the data points in \( X \) according to the training data.

2.2. In this part, k-NN classifier is used for text classification. Download the document classification data set "doc_data.zip" introduced in Homework 2. The features for classifying a document are the number of occurrences of each word (of the vocabulary) in the given document.

   a. [7 Pts] Find misclassification error of k-NN (k=1, k=3, k=5, k=15) on the training and test sets. Discuss about the obtained results.

   b. [3 Pts] Compare the results of k-NN with those of Naïve Bayes classifier that you found in Homework 2 and discuss about them.

Problem 3: Boosting

The AdaBoost algorithm (that we explained in the class) tries to minimize the exponential loss \( E = \sum_{i=1}^{n} e^{-y^{(i)}h_{t}(x^{(i)})} \) sequentially where \( h_{t}(x) = \frac{1}{2} [\alpha_{1}h_{1}(x) + \cdots + \alpha_{t}h_{t}(x)] \). Indeed, at the \( t \)-th iteration \( (1 \leq t \leq T) \), it finds the weak classifier \( h_{t}(x) \) and the corresponding coefficient \( \alpha_{t} \) so that the loss function (up to \( t \)-th iteration) is minimized.

3.1. [15 Pts] Suppose that we consider the sum of squares error \( E = \sum_{i=1}^{n} (y^{(i)} - h_{t}(x^{(i)}))^{2} \) as the objective function (instead of the exponential loss) and want to optimize it sequentially. What are the update rule for the coefficient \( \alpha_{t} \) and the classifier \( h_{t}(x) \)?

3.2. Consider the XOR dataset shown in the below figure ('X' corresponds to \( y = -1 \) and 'O' corresponds to \( y = 1 \)).

   a. [5 pts] Assume the base classifiers \( h_{t}(x) \) are linear classifiers. Will AdaBoost that minimizes the exponential loss get zero training error after sufficient iterations? What is the minimum number of iterations before it can reach zero training error?

   b. [5 pts] Will AdaBoost with Decision Stumps as the weak classifier ever achieve better than 50% classification accuracy on this dataset? (Justify your answer). What about the boosting algorithm explained in Problem 3.1 (with Decision Stumps as the base classifier)?

   ![XOR dataset diagram]

3.3. [5 Pts] How can we prevent Boosting from overfitting? How can we find the proper number of boosting iterations?

Problem 4: MDPs & Reinforcement Learning

4.1. Consider the grid world shown in the below figure. In each of the nine states of this world, four actions 'Up', 'Down', 'Left', and 'Right' are available. Each action achieves the intended effect with probability 0.7, but the rest of the time, the action moves the agent in one of the three unintended directions randomly (with the probability 0.1 in each of these directions). If the agent bumps into a wall, it stays in the same square. When the agent reaches the state 8 (this state is a terminal state), it receives +10 reward. The discount factor is \( \gamma = 0.9 \).
a. [8 Pts] Write down the numerical state values (of all states) after the first and second iterations of Value Iteration algorithm. Initial values of the states are set to zero.

<table>
<thead>
<tr>
<th></th>
<th>9</th>
<th>8</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

b. [2 Pts] After these two iterations, what is the currently selected policy? Is this sensible?

4.2. [10 Pts] Suppose that we don't know the transition probabilities and rewards and we want to use Q-learning algorithm. During this algorithm, we observe the following two episodes (above each transition, the action and the reward value have been shown):

\[
\begin{align*}
2 \rightarrow & U, 0 \rightarrow 3 \rightarrow R, 0 \rightarrow L, 0 \rightarrow R, 0 \rightarrow D, 0 \rightarrow U, 0 \rightarrow U, 0 \rightarrow R, 0 \rightarrow U, 0 \rightarrow L, 10 \\
4 \rightarrow & U, 0 \rightarrow L, 0 \rightarrow D, 0 \rightarrow D, 0 \rightarrow U, 0 \rightarrow L, 0 \rightarrow R, 0 \rightarrow U, 0 \rightarrow U, 10
\end{align*}
\]

What would be the resulting \( \hat{Q} \) values (if the initial values of \( \hat{Q} \) are zero) after the above episodes?