CS534 — Homework Assignment 5 *Solution*

A few notes.

1. This is the last written assignment, i.e., the last chance to make sure you make the 80% completion cut-off.
2. This assignment contains a total of 6 problems. But problems 3, 4 and 6 each contain two subparts. So it will count as a total of 9 points. Please keep this in mind to ensure that you can make the 80% completion cut-off.
3. Please submit your solution electronically via TEACH.

———————————- Beginning of Questions ———————————

1. Please show that in each iteration of Adaboost, the weighted error of h_i on the updated weights D_{i+1} is exactly 50%. In other words, $\sum_{j=1}^{N} D_{i+1}(j)I(h_i(X_j) \neq y_j) = 50\%$.

   Solution: *Let ϵ_i be the weighted error of h_i, that is $\epsilon_i = \sum_{j=1}^{N} D_i(j)I(h_i(X_j) \neq y_j)$, where $I(\cdot)$ is the indicator function that takes value 1 if the argument is true, and value 0 otherwise. Following the update rule of Adaboost, let's assume that the weights of the correct examples are multiplied by $e^{-\gamma}$, and those of the incorrect examples are multiplied by e^{γ}. After the updates, to make sure that h_i has exactly 50% accuracy, we only need to satisfy the following:*

$$\epsilon e^{-\gamma} = (1-\epsilon)e^{\gamma} \Rightarrow$$

$$\frac{\epsilon}{1-\epsilon} = e^{2\gamma} \Rightarrow$$

$$\log \frac{\epsilon}{1-\epsilon} = 2\gamma \Rightarrow$$

$$\gamma = \frac{1}{2} \log \frac{\epsilon}{1-\epsilon}$$

   which is exactly the value Adaboost uses.

2. In class we showed that Adaboost can be viewed as learning an additive model via functional gradient descent to optimize the following exponential loss function:

$$\sum_{i=1}^{N} \exp(-y_i \sum_{l=1}^{L} \alpha_l h_l(x_i))$$

   Our derivation showed that in each iteration l, to minimize this objective we should seek an h_l that minimizes the weighted training error, where the weight of each example $w_l^i = \exp(-y_i \sum_{t=1}^{l-1} \alpha_t h_t(x_i))$ prior to normalization. Show how this definition of w_l^i is proportional to the $D_l(i)$ defined in Adaboost.

   In Adaboost, we have:

$$D_1 = 1/N$$

   and in each iteration l, the weight is multiplied by a factor $\exp^{\alpha_{l-1}}$ for examples that are mistaken (i.e., $y_i h_{l-1}(x_i) = -1$), and $\exp^{-\alpha_{l-1}}$ for examples that are correct (i.e., $y_i h_{l-1}(x_i) = 1$).

   In short, this can be expressed as:

$$D_l(i) \propto D_{l-1}(i) * \exp^{-\alpha_{l-1} y_i h_{l-1}(x_i)}$$

   As such, we have

$$D_l(i) \propto \exp^{-\alpha_1 y_i h_1(x_i) - \alpha_2 y_i h_2(x_i) - \ldots - \alpha_{l-1} y_i h_{l-1}(x_i)} = \exp^{-y_i \sum_{t=1}^{l-1} \alpha_t h_t(x_i)} = w_l^i$$

3. **HAC**. Create by hand the clustering dendrogram for the following samples of ten points in one dimension.

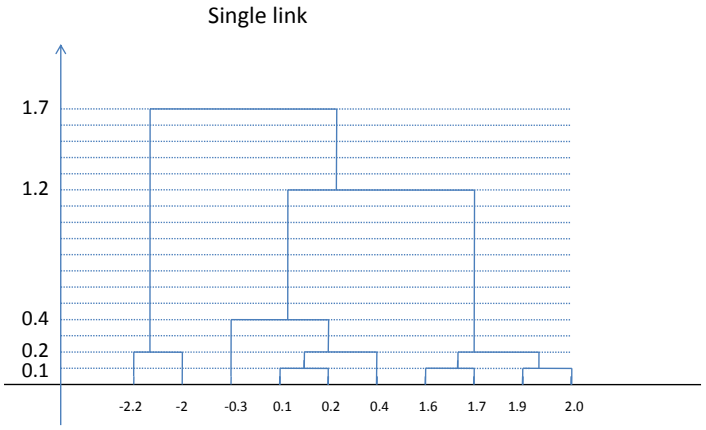
$$Sample = (-2.2, -2.0, -0.3, 0.1, 0.2, 0.4, 1.6, 1.7, 1.9, 2.0)$$

   a. Using single link.

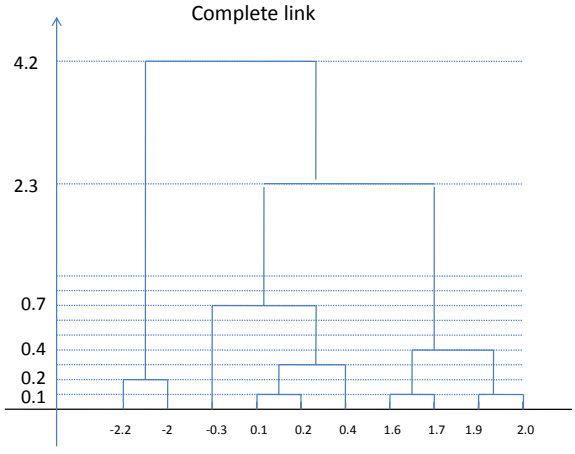
   b. Using complete link

   Solution:

   For the single link case we get the following:



   The complete link case turns out to yield the same hierarchy as the single link case. But the dendrogram looks slightly different.



4. **Picking k for Kmeans**. One shortcoming of Kmeans is that one has to specify the value of k. Consider the following strategy for pick k automatically: try all possible values of k and choose k that minimizes J_e. Argue (briefly) why this strategy is a good/bad idea. Provide an alternative strategy.
   Solution: *This strategy is a very bad idea. The reason is that the optimal distortion on the training data will always decrease as we increase k (until k equals the number of instances). In particular, note that when $k = n$ the number of instances that $J_e = 0$ so the approach will always select $k = n$.*

   Alternative strategy: one could use a hold-out validation set or cross-validation and choose the k value that minimizes J_e on the validation set. One could also select k to be the elbow point on the SSE curve, i.e., the k value where the decreasing rate of SSE decreases abruptly. Note other answers can be

used as well. For example stability based method. For all answers don't just provide a name, explain the gist of it as well.

5. **Gaussian Mixture Models**. Let our data be generated from a mixture of two univariate gaussian distributions, where $f(x|\theta_1)$ is a Gaussian with mean $\mu_1 = 0$ and $\sigma^2 = 1$, and $f(x|\theta_2)$ is a Gaussian with mean $\mu_2 = 0$ and $\sigma^2 = 0.5$. The only unknown parameter is the mixing parameter α (which specifies the prior probability of θ_1.). Now we observe a single sample x_1, please write out the likelihood function of x_1 as a function of α, and determine the maximum likelihood estimation of α.

   Solution:

   Thus we can write the likelihood function $L(\alpha) = p(x_1|\alpha)$ as:

$$p(x_1|\alpha) = \frac{\alpha}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} + \frac{1-\alpha}{\sqrt{\pi}} e^{-x^2}$$

   Consider that a single sample x_1 has been observed. Determine the maximum likelihood estimate of α.

   We can write the likelihood as follows:

$$p(x_1|\alpha) = \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_1^2} - \frac{1}{\sqrt{\pi}} e^{-x_1^2} \right) \alpha + \frac{1}{\sqrt{\pi}} e^{-x_1^2}$$

   Thus, we see that the likelihood is simply a linear function of alpha where the sign of the slope is determined by which Gaussian produces the larger response. Since we know that $0 \leq \alpha \leq 1$, this tells us that if the slope is positive that we should choose $\alpha = 1$ and otherwise if the slope is negative we should choose $\alpha = 0$. Using straightforward algebra one can show that the slope is positive whenever $x_1^2 \geq \log 2$ and we should set $\alpha = 1$ otherwise set $\alpha = 0$. Alternatively, one could also apply Expectation maximization for this problem (not an efficient solution). Starting with $\alpha = 0.5$ and applying EM, you would observe that in each iteration, your estimate of α will strictly increase or decrease depends on which of the two Gaussians fit x_1 better, eventually lead to 1 or 0 accordingly.

4. **Expectation Maximization for Mixture of Multinomials**
   Consider a random variable x that is categorical with M possible values $1, \cdots, M$. We now represent x as a vector such that for $i = 1, ..., M$, $x(i) = 1$ iff x takes the ith value, and $\sum_i^M x(i) = 1$. The distribution of x is described by a mixture of K discrete Multinomial distributions such that:

$$p(x) = \sum_{k=1}^{K} \pi_k p(x|\mu_k)$$

   and

$$p(x|\mu_k) = \prod_{j=1}^{M} \mu_k(j)^{x(j)}$$

   where π_k denotes the mixing coefficient for the kth component (aka the prior probability that the hidden variable $z = k$), and μ_k specifies the parameters of the kth component. Specifically, $\mu_k(j)$ represents the probabilities $p(x(j) = 1|z = k)$, and $\sum_j \mu_k(j) = 1$. Given an observed data set $\{x_i\}, i = 1, \cdots, N$, please derive the E and M step equations of the EM algorithm for optimizing the mixing coefficient and the component parameters $\mu_k(j)$ for this distribution. For your reference, here is the generic formula for the E and M steps. Note that θ is used to denote all the parameters of the mixture model.

   - **E-step.** For each i, calculate $Q_i(z_i) = p(z_i|x_i;\theta)$, i.e., the prob. that observation i belongs to each of the K cluster.
   - **M-step.** Set

$$\theta := \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \sum_{z_i} Q_i(z_i) log \frac{p(x_i, z_i;\theta)}{Q_i(z_i)}.$$

Solution: E-step: *We set*

$$\begin{aligned} Q_i(z_i = k) &= p(z_i = k|\mathbf{x}_i;\theta) \\ &= \frac{p(\mathbf{x}_i|z_i = k;\theta)p(z_i = k|\theta)}{p(\mathbf{x}_i|\theta)} \\ &= \frac{\pi_k p(\mathbf{x}_i|\mu_k)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_i|\mu_j)} \\ &= \frac{\pi_k p(\mathbf{x}_i|\mu_k)}{\sum_{j=1}^K \pi_j \prod_{l=1}^M \mu_j(l)^{\mathbf{x}_i(l)}} \\ &= \frac{\pi_k \prod_{j=1}^M \mu_k(j)^{x_i(j)}}{\sum_{j=1}^K \pi_j \prod_{l=1}^M \mu_j(l)^{\mathbf{x}_i(l)}} \end{aligned}$$

M-step: *We now update θ:*

$$\begin{aligned} \theta &:= \operatorname*{argmax}_\theta \sum_{i=1}^N \sum_{j=1}^K Q_i(z_i = j) \log \frac{p(\mathbf{x}_i, z_i = j;\theta)}{Q_i(z_i = j)} \\ &= \operatorname*{argmax}_\theta \sum_{i=1}^N \sum_{j=1}^K Q_i(z_i = j) \log p(\mathbf{x}_i, z_i = j;\theta) \\ &= \operatorname*{argmax}_\theta \sum_{i=1}^N \sum_{j=1}^K Q_i(z_i = j) \log p(\mathbf{x}_i|z_i = j;\theta)p(z_i = j;\theta) \\ &= \operatorname*{argmax}_\theta \sum_{i=1}^N \sum_{j=1}^K Q_i(z_i = j) \log \pi_j \prod_{k=1}^M \mu_j(k)^{\mathbf{x}_i(k)} \\ &= \operatorname*{argmax}_\theta \sum_{i=1}^N \sum_{j=1}^K \left(Q_i(z_i = j) \log \pi_j + Q_i(z_i = j) \sum_{k=1}^M \log \mu_j(k)^{\mathbf{x}_i(k)} \right) \\ &= \operatorname*{argmax}_\theta \sum_{i=1}^N \sum_{j=1}^K \left(Q_i(z_i = j) \log \pi_j + Q_i(z_i = j) \sum_{k=1}^M \mathbf{x}_i(k) \log \mu_j(k) \right) \end{aligned}$$

Optimizing μ_l: *We begin by eliminating terms that are constant with respect to μ_l:*

$$\sum_{i=1}^N Q_i(z_i = l) \sum_{k=1}^M \mathbf{x}_i(k) \log \mu_l(k)$$

We now use a Lagrangian to constrain μ_l to be a probability distribution:

$$\mathcal{L}(\mu_l) = \sum_{i=1}^N Q_i(z_i = l) \sum_{k=1}^M \mathbf{x}_i(k) \log \mu_l(k) + \beta(\sum_{j=1}^M \mu_l(j) - 1)$$

Solving for $\mu_l(k)$:

$$\begin{aligned}
\frac{\partial}{\partial_{\mu_l(k)}}\mathcal{L}(\mu_l) = \sum_{i=1}^{N} Q_i(z_i = l)\frac{\mathbf{x}_i(k)}{\mu_l(k)} + \beta &= 0 \\
\frac{1}{\mu_l(k)}\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(k) + \beta &= 0 \\
\frac{1}{\mu_l(k)} &= \frac{-\beta}{\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(k)} \\
\mu_l(k) &= \frac{\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(k)}{-\beta}
\end{aligned}$$

Knowing that $\sum_{j=1}^{M} \mu_l(j) = 1$, thus we have:

$$\begin{aligned}
\sum_{j=1}^{M} \frac{\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(j)}{-\beta} - 1 &= 0 \\
\frac{1}{-\beta}\sum_{j=1}^{M}\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(j) &= 1 \\
\sum_{j=1}^{M}\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(j) &= -\beta
\end{aligned}$$

Finally, substituting backwards gives the solution for the parameter μ_l:

$$\mu_l(k) = \frac{\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(k)}{\sum_{j=1}^{M}\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(j)}$$

Noting that $\sum_{j=1}^{M} \mathbf{x}_i(j) = 1$, we can replace the denominator with N:

$$\mu_l(k) = \frac{\sum_{i=1}^{N} Q_i(z_i = l)\mathbf{x}_i(k)}{\sum_{i=1}^{N} Q_i(z_i = l)}$$

This can be simply interpreted as among the total mass that was deemed to belong to cluster l (denominator), the portion that had $x(k) = 1$(numerator).

Optimizing π_l: *Again beginning by eliminating constant terms in the gradient with respect to π_l:*

$$\sum_{i=1}^{N} Q_i(z_i = l)\log \pi_l$$

Forming the Lagrangian with the constraint that $\sum_{j=1}^{K} \pi_j = 1$:

$$\mathcal{L}(\pi_l) = \sum_{i=1}^{N} Q_i(z_i = l)\log \pi_l + \beta(\sum_{j=1}^{K} \pi_j - 1)$$

Solving for π_l:

$$\begin{aligned}
\frac{d}{d\pi_l}\mathcal{L} = \sum_{i=1}^{N}\frac{Q_i(z_i=l)}{\pi_l} + \beta &= 0 \\
\frac{1}{\pi_l}\sum_{i=1}^{N} Q_i(z_i=l) &= -\beta \\
\sum_{i=1}^{N} Q_i(z_i=l) &= -\beta\pi_l \\
\pi_l &= \frac{\sum_{i=1}^{N} Q_i(z_i=l)}{-\beta}
\end{aligned}$$

Knowing that $\sum_{j=1}^{K}\pi_j = 1$

We have

$$\begin{aligned}
\sum_{j=1}^{K}\frac{\sum_{i=1}^{N} Q_i(z_i=j)}{-\beta} &= 1 \\
\frac{1}{-\beta}\sum_{j=1}^{K}\sum_{i=1}^{N} Q_i(z_i=j) &= 1 \\
-\beta &= \sum_{j=1}^{K}\sum_{i=1}^{N} Q_i(z_i=j)
\end{aligned}$$

Substituting backwards gives the solution for the parameter π_l:

$$\pi_l = \frac{\sum_{i=1}^{N} Q_i(z_i=l)}{\sum_{j=1}^{K}\sum_{i=1}^{N} Q_i(z_i=j)} = \frac{\sum_{i=1}^{N} Q_i(z_i=l)}{N}$$

This again can be simply interpreted as among a total of N examples, what proportion is deemed to belong cluster l.

6. **Dimension reduction.**

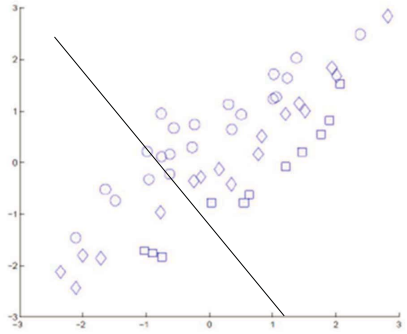
   a. Consider the following data set, please draw on the picture the the 1st Principal component direction, and the direction for LDA respectively. Note for PCA, please ignore the markers, and for LDA, we treat the circles as one class and the rest as the other class.

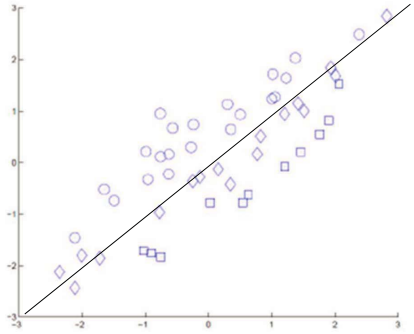
   Answer: *The projection lines by LDA and PCA are shown in the figure below:*

   b. Given three data points, (0,0), (1,2), (-1, -2) in a 2-d space. What is the first principal component direction (please write down the actual vector)? If you use this vector to project the data points, what are their new coordinates in the new 1-d space? What is the variance of the projected data?

   Answer: *One could simply noting that the three points lie on a straight line, which must be the principal component direction. As such, we can simply write out the line $2x - y = 0$. The direction vector for this line is simply: $(\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}})$. One could also follow the recipe given in class. Begin by forming the covariance matrix:*

$$S = \frac{1}{3}\left(\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}\right) = \frac{2}{3}\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$





We now solve for the eigenvectors by solving $S - I\lambda = 0$. Using the determinant:

$$\begin{aligned} \begin{vmatrix} 2/3-\lambda & 4/3 \\ 4/3 & 8/3-\lambda \end{vmatrix} &= 0 \\ (2/3-\lambda)(8/3-\lambda) - 16/3 &= 0 \\ \lambda^2 - 10/3\lambda &= 0 \\ \lambda(\lambda - 10/3) &= 0 \\ \Rightarrow \lambda_1 = 0, \lambda_2 = 10/3 & \end{aligned}$$

To recover the first eigenvector we solve the system

$$\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5x_1 \\ 5x_2 \end{pmatrix}$$

which gives

$$u_1 = \begin{pmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{pmatrix}$$

If we project the data by this vector, we get

$$\begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ 0 & 2 & -2 \end{pmatrix} = \begin{pmatrix} 0 & \sqrt{5} & -\sqrt{5} \end{pmatrix}$$

The variance of this data is

$$\frac{1}{3}\left(0^2 + \sqrt{5}^2 + (-\sqrt{5})^2\right) = \frac{10}{3}$$

Note that above we are using MLE for the variance. If one choose to use an unbiased estimator of the variance (i.e., replacing N with N-1 in the normalization term), the solution for PC does not change but the (co)variance before and after projection will change proportionally, and everything will work out in a similar way.