CM146, Winter 2024
Problem Set 4: Boosting, Unsupervised learning
Due March 15, 11:59pm (Math), March 17, 11:59pm (Coding)

1 AdaBoost [5 pts]

In the lecture on ensemble methods, we said that in iteration t, AdaBoost is picking (h_t, β_t) that minimizes the objective:

$$\begin{aligned}(h_t^*(\boldsymbol{x}), \beta_t^*) &= \underset{(h_t(\boldsymbol{x}),\beta_t)}{\arg\min} \sum_n w_t(n) e^{-y_n \beta_t h_t(\boldsymbol{x}_n)} \\ &= \underset{(h_t(\boldsymbol{x}),\beta_t)}{\arg\min} (e^{\beta_t} - e^{-\beta_t}) \sum_n w_t(n) \mathbb{I}[y_n \neq h_t(\boldsymbol{x}_n)] \\ &\qquad + e^{-\beta_t} \sum_n w_t(n)\end{aligned}$$

We define the weighted misclassification error at time t, ϵ_t to be $\epsilon_t = \sum_n w_t(n)\mathbb{I}[y_n \neq h_t(\boldsymbol{x}_n)]$. Also the weights are normalized so that $\sum_n w_t(n) = 1$.

(a) **(3 pts)** Take the derivative of the above objective function with respect to β_t and set it to zero to solve for β_t and obtain the update for β_t.

Solution:

$$J(\beta) = (e^{\beta_t} - e^{-\beta_t})\epsilon_t + e^{-\beta_t}$$

$$\begin{aligned}\frac{\partial J(\beta_t)}{\partial \beta_t} &= (e^{\beta_t} + e^{-\beta_t})\epsilon_t - e^{-\beta_t} \\ &= 0\end{aligned}$$

Solving gives:

$$\begin{aligned}e^{2\beta_t} + 1 &= \frac{1}{\epsilon_t} \\ e^{2\beta_t} &= \frac{1-\epsilon_t}{\epsilon_t} \\ \beta_t &= \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\end{aligned}$$

(b) **(2 pts)** Suppose the training set is linearly separable, and we use a hard-margin linear support vector machine (no slack) as a base classifier. In the first boosting iteration, what would the resulting β_1 be?

Solution: The value of β_1 is infinite. Increasing β_1 will decrease all the training losses since $y_n h(\boldsymbol{x}_n) > 0$ for all n.

Parts of this assignment are adapted from course material by Jenna Wiens (UMich) and Tommi Jaakola (MIT).

That is, if the training set is linearly separable and we use a hard-margin SVM with no slack as a base classifier, then a single classifier is sufficient for the ensemble classifier. Thus, in stage $t = 1$, $h(\boldsymbol{x})$ will correspond to a linear decision boundary that correctly classifies all points ($\epsilon_1 = 0$) so that $\beta_1 = \infty$.

2 K-means for single dimensional data [5 pts]

In this problem, we will work through K-means for a single dimensional data.

(a) **(2 pts)** Consider the case where $K = 3$ and we have 4 data points $x_1 = 1, x_2 = 2, x_3 = 5, x_4 = 7$. What is the optimal clustering for this data ? What is the corresponding value of the K-means objective ?

Solution:

$$c_1 = 1.5, c_2 = 5, c_3 = 7, objective = 0.5^2 + 0.5^2 + 0 + 0 = 0.5$$

(b) **(3 pts)** One might be tempted to think that Lloyd's algorithm is guaranteed to converge to the global minimum when dimension $d = 1$. Show that there exists a suboptimal cluster assignment (*i.e.*, initialization) for the data in the above part that Lloyd's algorithm will not be able to improve (to get full credit, you need to show the assignment, show why it is suboptimal *and* explain why it will not be improved).

Solution:

$$c_1 = 1, c_2 = 2, c_3 = 6, objective = 0^2 + 0^2 + 1^2 + 1^2 = 2$$

When running Lloyd's on this configuration, the new cluster assignment will be $z_1 = 1, z_2 = 2, z_3 = 3, z_4 = 3$.

3 Gaussian Mixture Models [8 pts]

We would like to cluster data $\{x_1, \ldots, x_N\}$, $x_n \in \mathbb{R}^d$ using a Gaussian Mixture Model (GMM) with K mixture components. To do this, we need to estimate the parameters $\boldsymbol{\theta}$ of the GMM, *i.e.*, we need to set the values $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ where ω_k is the mixture weight associated with mixture component k, and $\boldsymbol{\mu}_k$ and Σ_k denote the mean and the covariance matrix of the Gaussian distribution associated with mixture component k.

If we knew which cluster each sample x_n belongs to (i.e., we have the complete data), we showed in the lecture on Clustering that the log likelihood l is what we have below and we can compute the maximum likelihood estimate (MLE) of all the parameters.

$$\begin{aligned} l(\boldsymbol{\theta}) &= \sum_n \log p(\boldsymbol{x}_n, z_n) \\ &= \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \end{aligned} \tag{1}$$

Since we do not have the complete data, we use the EM algorithm. The EM algorithm works by iterating between setting each γ_{nk} to the posterior probability $p(z_n = k|\boldsymbol{x}_n)$ (step 1 on slide 26 of the lecture on Clustering) and then using γ_{nk} to find the value of $\boldsymbol{\theta}$ that maximizes l (step 2 on slide 26). We will now derive updates for one of the parameters, *i.e.*, $\boldsymbol{\mu}_j$ (the mean parameter associated with mixture component j).

(a) **(2 pts)** To maximize l, compute $\nabla_{\boldsymbol{\mu}_j} l(\boldsymbol{\theta})$: the gradient of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\mu}_j$.
Solution: Gaussian/Normal distribution:

$$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)\right)$$

The log-likelihood function:

$$l(\boldsymbol{\theta}) = \text{Const} + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

which can be simplified as

$$l(\boldsymbol{\theta}) = \sum_k \left\{ \sum_n \gamma_{nk} \left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k) \right] \right\}$$

The gradient is computed as,

$$\nabla_{\boldsymbol{\mu}_j} l(\boldsymbol{\theta}) = \sum_n \gamma_{nj} \boldsymbol{\Sigma}_j^{-1} \left(\boldsymbol{x}_n - \boldsymbol{\mu}_j\right)$$

Note that we've applied,

$$\frac{\partial \left(\boldsymbol{x}^T A \boldsymbol{x}\right)}{\partial \boldsymbol{x}} = \boldsymbol{x}^T \left(A + A^T\right)$$

(b) **(2 pts)** Set the gradient to zero and solve for $\boldsymbol{\mu}_j$ to show that $\boldsymbol{\mu}_j = \frac{1}{\sum_n \gamma_{nj}} \sum_n \gamma_{nj} \boldsymbol{x}_n$.
Solution: Set the results in (1) as zero and we can directly get

$$\boldsymbol{\mu}_j = \frac{1}{\sum_n \gamma_{nj}} \sum_n \gamma_{nj} \boldsymbol{x}_n$$

(c) **(4 pts)** Suppose that we are fitting a GMM to data using $K = 2$ components. We have $N = 5$ samples in our training data with $x_n, n \in \{1, \ldots, N\}$ equal to: $\{5, 15, 25, 30, 40\}$.

We use the EM algorithm to find the maximum likeihood estimates for the model parameters, which are the mixing weights for the two components, ω_1 and ω_2, and the means for the two components, μ_1 and μ_2. The standard deviations for the two components are fixed at 1. Suppose that at the end of step 1 of iteration 5 in the EM algorithm, the soft assignment γ_{nk} for the five data items are as shown in Table 1.

What are updated values for the parameters ω_1, ω_2, μ_1, and μ_2 at the end of step 2 of the EM algorithm?
Solution: The one-dimensional GMM is defined as

$$p(x) = \sum_{i=1}^{K} \omega_i \mathcal{N}\left(x \mid \mu_i, \sigma_i\right)$$

γ_1	γ_2
0.2	0.8
0.2	0.8
0.8	0.2
0.9	0.1
0.9	0.1

Table 1: Entry in row n and column k of the table corresponds to γ_{nk}

$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

$$\sum_{i=1}^{K} \omega_i = 1$$

Update ω_1, ω_2 in the M-Step:

$$\hat{\omega}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}$$

we have

$$\omega_1 = \frac{0.2 + 0.2 + 0.8 + 0.9 + 0.9}{5} = \frac{3}{5}, \ \omega_2 = \frac{0.8 + 0.8 + 0.2 + 0.2 + 0.1}{5} = \frac{2}{5}$$

Update μ_1, μ_2 in the M-Step:

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

we have

$$\mu_1 = \frac{0.2 \times 5 + 0.2 \times 15 + 0.8 \times 25 + 0.9 \times 30 + 0.9 \times 40}{0.2 + 0.2 + 0.8 + 0.9 + 0.9} = \frac{87}{3} = 29$$

$$\mu_2 = \frac{0.8 \times 5 + 0.8 \times 15 + 0.2 \times 25 + 0.1 \times 30 + 0.1 \times 40}{0.8 + 0.8 + 0.2 + 0.1 + 0.1} = \frac{28}{2} = 14$$