CSCI 567, Spring 2023
Yan Liu

Homework #3

Released: Mar 6, 2023
Due: 11:59 pm, Mar 27, 2023
Total scores: 90 points

Problem 1 Decision Trees (15 points)

Consider a binary dataset with 400 examples, where half of them belongs to class A and the other half belongs to class B.

Next consider two decision stumps (i.e. trees with depth 1) $\mathcal{T}_1$ and $\mathcal{T}_2$, each with two children. For $\mathcal{T}_1$, its left child has 150 examples in class A and 50 examples in class B; for $\mathcal{T}_2$, its left child has 0 example in class A and 100 examples in class B. (You should infer what are in the right child.)

1.1 For each leaf of $\mathcal{T}_1$ and $\mathcal{T}_2$, compute the corresponding classification error, entropy (base e) and Gini impurity, rounding off your answer to 2 decimal places. (Note: the value/prediction of each leaf is the majority class among all examples that belong to that leaf.) **(6 points)**

1.2 Compare the quality of $\mathcal{T}_1$ and $\mathcal{T}_2$ (that is, the two different splits of the root) based on classification error, conditional entropy (base e), and weighted Gini impurity respectively, rounding off your answer to 2 decimal places. **(9 points)**

Problem 2 Boosting (20 points)

Recall the procedure of AdaBoost algorithm described in class:

Algorithm 1: Adaboost

Given: A training set $\{(\boldsymbol{x}_n, y_n \in \{+1, -1\})\}_{n=1}^N$, and a set of classifier $\mathcal{H}$, where each $h \in \mathcal{H}$ takes a feature vector as input and outputs $+1$ or -1.

Goal: Learn $H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^T \beta_t h_t(\boldsymbol{x})\right)$

Initialization: $D_1(n) = \frac{1}{N}$, $\forall n \in [N]$.

for $t = 1, 2, \cdots, T$ **do**

Find $h_t = \arg\min_{h \in \mathcal{H}} \sum_{n: y_n \neq h(\boldsymbol{x}_n)} D_t(n)$.

Compute

$$\epsilon_t = \sum_{n: y_n \neq h_t(\boldsymbol{x}_n)} D_t(n) \qquad \text{and} \qquad \beta_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}.$$

Compute

$$D_{t+1}(n) = \frac{D_t(n) e^{-\beta_t y_n h_t(\boldsymbol{x}_n)}}{\sum_{n'=1}^N D_t(n') e^{-\beta_t y_{n'} h_t(\boldsymbol{x}_{n'})}}$$

for each $n \in [N]$

2.1 We discussed in class that AdaBoost minimizes the exponential loss greedily. In particular, Adaboost seeks the optimal β_t that minimizes

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

where ϵ_t is the weighted classification error of h_t and is fixed. Show that $\beta^* = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ is the minimizer. **(8 points)**

2.2 Recall that at round t of AdaBoost, a classifier h_t is obtained and the weighting over the training set is updated from D_t to D_{t+1}. Prove that h_t is only as good as random guessing in terms of classification error weighted by D_{t+1}. That is **(12 points)**

$$\sum_{n: h_t(\mathbf{x}_n) \neq y_n} D_{t+1}(n) = \frac{1}{2}.$$

Hint: you can somehow ignore the denominator of $D_{t+1}(n)$ to simplify calculation.

Problem 3 Optimization over the simplex (25 points)

Many machine learning problems, especially clustering problems, involve optimization over a **simplex**. Thus, in this exercise, you will prove two optimization results over the simplex that we used multiple times in the lectures. Recall that a $K-1$ dimensional simplex Δ is defined as

$$\Delta = \{\mathbf{q} \in \mathbb{R}^K \mid q_k \geq 0,\ \forall k \text{ and } \sum_{k=1}^{K} q_k = 1\},$$

which means that a $K-1$ dimensional simplex has K non-negative entries, and the sum of all K entries is 1. The property that a simplex sums to one coincides with a probability distribution of a discrete random variable of K possible outcomes, and is thus frequently used in clustering problems.

3.1 Given $a_1, \ldots, a_K \in \mathbb{R}^+$, solve the following optimization over simplex (find the optimal value of q_k.) **(12 points)**

$$\begin{aligned} &\arg\max_{\mathbf{q}} \sum_{k=1}^{K} a_k \ln q_k, \\ &\text{s.t. } q_k \geq 0, \\ &\sum_{k=1}^{K} q_k = 1. \end{aligned}$$

3.2 Given $b_1, \ldots, b_K \in \mathcal{R}$, solve the following optimization problem **(13 points)**

$$\begin{aligned} &\arg\max_{\mathbf{q} \in \Delta} \sum_{k=1}^{K} \left(q_k b_k - q_k \ln q_k\right), \\ &\text{s.t. } q_k \geq 0, \\ &\sum_{k=1}^{K} q_k = 1. \end{aligned}$$

Problem 4 Gaussian Mixture Model and EM (30 points)

4.1 In the lecture we applied EM to learn Gaussian Mixture Models (GMMs) and showed the M-Step without a proof. Now it is time that you prove it.

Consider a GMM with the following PDF of $\mathbf{x}_n$:

$$p(\mathbf{x}_n) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k) = \sum_{k=1}^{K} \frac{\omega_k}{(\sqrt{2\pi})^D |\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)\right)$$

where $K \in \mathbb{N}$ is the number of Gaussian components, $D \in \mathbb{N}$ is dimension of a data point $\mathbf{x}_n$. This GMM has K tuples of model parameters $(\mu_k, \Sigma_k, \omega_k)$, which standards for the mean vector, covariance matrix, and component weight of the k-th Gaussian component. $|\Sigma|$ denotes the determinant of matrix Σ.

For simplicity, we further assume that all components are isotropic Gaussian, i.e., $\Sigma_k = \sigma_k^2 I$. Find the MLE of *the expected complete log-likelihood*. Equivalently, find the optimal solution to the following optimization problem.

$$\begin{aligned} \arg\max_{\omega_k, \boldsymbol{\mu}_k, \Sigma_k} \; & \sum_n \sum_k \gamma_{nk} \ln \omega_k + \sum_n \sum_k \gamma_{nk} \ln \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k) \\ \text{s.t. } & \omega_k \geq 0 \\ & \sum_{k=1}^{K} \omega_k = 1 \end{aligned}$$

where γ_{nk} is the posterior of latent variables computed from the E-Step. Hint: you can make use of the result from Problem 3.1. **(15 points)**

4.2 In the lecture we derived EM through a lower bound of the log-likelihood function. Specifically, we find the tightest lower bound by solving

$$\arg\max_{\mathbf{q}_n \in \Delta} \mathbb{E}_{z_n \sim \mathbf{q}_n} \left[\ln p(\mathbf{x}_n, z_n \,; \theta^{(t)})\right] - \mathbb{E}_{z_n \sim \mathbf{q}_n} \left[\ln \mathbf{q}_n\right].$$

Find the optimal solution to this optimization problem using the results from Problem 3.2. (The solution was already given in the lecture, but here we require you to derive it.)

(5 points)

4.3 The posterior probability of z in GMM can be seen as a *soft* assignment to the clusters; In contrast, K-means assign each data point to one cluster at each iteration (*hard* assignment). Describe how to set the model parameters such that the GMM in the previous question reduces to a K-means; please also derive $p(z_n = k|\boldsymbol{x}_n)$ in this case.

(10 points)