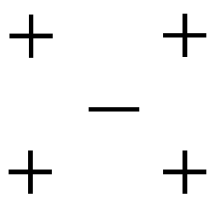
Intro to Big Data Science: Assignment 4

Due Date: Apr 29, 2025

✏ **Exercise 1** Consider training an AdaBoost classifier using decision stumps on the five-point data set (4 "+" samples and 1 "-" sample):



1. Which examples will have their weights increased at the end of the first iteration? Circle them.
2. How many iterations will it take to achieve zero training error? Explain by doing some computation using the above algorithm.
3. Can you add one more sample to the training set so that AdaBoost will achieve zero training error in two steps? If not, explain why.

✏ **Exercise 2** (Alternative objective functions) This problem studies boosting-type algorithms defined with objective functions different from that of AdaBoost. We assume that the training data are given as m labeled examples $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \{-1, +1\}$. We further assume that Φ is a strictly increasing convex and differentiable function over $\mathbb{R}$ such that: $\forall x \geq 0, \Phi(x) \geq 1$ and $\forall x < 0, \Phi(x) > 0$. Note that the exponential loss in AdaBoost is a special case.

(a) Consider the loss function $L(\alpha) = \sum_{i=1}^{m} \Phi(-y_i f(x_i))$ where f is a linear combination of base classifiers, i.e., $f = \sum_{t=1}^{T} \alpha_t b_t$ (as in AdaBoost). Derive a new boosting algorithm using the objective function L. In particular, characterize the best base classifier b_u to select at each round of boosting if we use coordinate descent (alternating iteration).

(b) Consider the following functions: (1) zero-one loss $\Phi_1(-u) = \mathbb{1}_{u \le 0}$; (2) least squared loss $\Phi_2(-u) = (1-u)^2$; (3) SVM loss $\Phi_3(-u) = \max\{0, 1-u\}$; and (4) logistic loss (binomial deviance) $\Phi_4(-u) = \log(1+e^{-u})$. Which functions satisfy the assumptions on Φ stated earlier in this problem?

(c) For each loss function satisfying these assumptions, derive the corresponding boosting algorithm. How do the algorithm(s) differ from AdaBoost?

✏ **Exercise 3** (Hierarchical Clustering)

Suppose that we have four observations, for which we compute a dissimilarity matrix, given by

$$\begin{pmatrix} 0 & 0.3 & 0.4 & 0.7 \\ 0.3 & 0 & 0.5 & 0.8 \\ 0.4 & 0.5 & 0 & 0.45 \\ 0.7 & 0.8 & 0.45 & 0 \end{pmatrix}$$

For instance, the dissimilarity between the first and second observations is 0.3, and the dissimilarity between the second and fourth observations is 0.8.

1. On the basis of this dissimilarity matrix, sketch the dendrogram that results from hierarchically clustering these four observations using complete linkage. Be sure to indicate on the plot the height at which each fusion (merge) occurs, as well as the observations corresponding to each leaf in the dendrogram.
2. Repeat 1, this time using single linkage clustering.
3. Suppose that we cut the dendrogram obtained in 1 such that two clusters result. Which observations are in each cluster?
4. Suppose that we cut the dendrogram obtained in 2 such that two clusters result. Which observations are in each cluster?
5. It is mentioned in the chapter that at each fusion in the dendrogram, the position of the two clusters being fused can be swapped without changing the meaning of the dendrogram. Draw a dendrogram that is equivalent to the dendrogram in 1, for which two or more of the leaves are repositioned, but for which the meaning of the dendrogram is the same.

✏ **Exercise 4** In this problem, you need to show that the within-cluster point scatter (or in other words, the sum-of-squared errors (SSE)) is **non-increasing** when the number of clusters increases.

Consider a data set $\mathscr{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ that contains N observations. Each sample $\mathbf{x}_i$ is a d-dimensional vector of continuous-valued attributes. You are performing K-means clustering.

1. Suppose all the N samples are grouped into a **single** cluster. Let μ be the centroid of the cluster. Express the total sum-of-squared errors SSE_T in terms of $\mathbf{x}_i$, μ and N. And show that SSE_T can be decomposed into d separate terms, one for each attribute, i.e., $SSE_T = \sum_{j=1}^{d} SSE_j$.

2. Now, suppose all the N observations are grouped into two clusters, C_1 and C_2. Let μ_1 and μ_2 be their corresponding cluster centroids while n_1 and n_2 are their respective cluster sizes ($n_1 + n_2 = N$). Express the sum-of-squared errors for each cluster, $SSE^{(j)}$ ($j = 1$ or 2), in terms of $\mathbf{x}_i$, n_j, and μ_j. You need to expand the quadratic term, $(a-b)^2 = a^2 - 2ab + b^2$, and simplify the expression.
3. By rewriting your expression for SSE_T in terms of $\mathbf{x}_i$, n_1, n_2, μ_1, μ_2 and N, show that $SSE_T \geqslant SSE^{(1)} + SSE^{(2)}$.

✏ **Exercise 5** (EM Algorithm)

Imagine a class where the probability that a student gets an "A" grade is $\mathbb{P}(A) = \frac{1}{2}$, a "B" grade is $\mathbb{P}(B) = \mu$, a "C" grade is $\mathbb{P}(C) = 2\mu$, and a "D" grade is $\mathbb{P}(D) = \frac{1}{2} - 3\mu$. We are told that c students get a "C" and d students get a "D". We don't know how many students got exactly an "A" or exactly a "B". But we do know that h students got either an "A" or "B". Let a be the number of students getting "A" and b be the number of students getting "B". Therefore, a and b are unknown parameters with $a + b = h$. Our goal is to use expectation maximization to obtain a maximum likelihood estimate of μ.

1. Use Multinoulli distribution to compute the log-likelihood function $l(\mu, a, b)$.
2. Expectation step: Given $\hat{\mu}^{(m)}$, compute the expected values $\hat{a}^{(m)}$ and $\hat{b}^{(m)}$ of a and b respectively.
3. Maximization step: Plug $\hat{a}^{(m)}$ and $\hat{b}^{(m)}$ into the log-likelihood function $l(\mu, a, b)$ and calculate for the maximum likelihood estimate $\hat{\mu}^{(m+1)}$ of μ, as a function of $\hat{\mu}^{(m)}$.
4. Iterating between the E-step and M-step will always converge to a local optimum of μ (which may or may not also be a global optimum)? Explain why in short.

✏ **Exercise 6** Online study and exercises.