In the name of GOD.

Sharif University of Technology

Machine Learning

Spring 2025

Hamid R. Rabiee, Zahra Dehghanian

Homework 2 | Deadline: 1403/12/26

1. [20] **Consider the following two distributions and answer the question:**

   Distribution 1:

   A scientist is studying the decay of a radioactive material. The measured radiation levels $Y_1, Y_2, \ldots, Y_N$ over time are assumed to be independent and normally distributed:

   $$Y_i \sim N(\mu_i, \sigma^2)$$

   where the mean follows an exponential decay model:

   $$\mu_i = \theta e^{-t_i}$$

   Here, t_i represents time, and the decay process suggests that as time increases, the expected radiation level decreases exponentially. The values of Y_i are assumed to be independent given the parameter θ.

   Here, θ is the unknown parameter, and σ^2 is a known constant variance.

   Distribution 2:

   Suppose we want to model a 1-dimensional dataset of N real-valued features ($x^{(i)}$) and target values ($y^{(i)}$) using the following exponential transformation model:

   $$y^{(i)} \sim N(\exp(wx^{(i)}), 1)$$

   where w is the unknown scalar parameter.

   The data points $y^{(i)}$ are assumed to be independent given the parameter w.

   Can the maximum conditional negative log-likelihood estimator of the unknown parameter (θ in Distribution 1, w in Distribution 2) be solved analytically?

   - If so, find the expression for the MLE of the parameter.
   - If not, explain why and provide the gradient update rule for the parameter using gradient descent.

2. [15] Consider the linear regression model:

$$Y_i = \alpha + \beta X_i + \epsilon_i.$$

(a) Suppose the errors ϵ_i are independent but **not identically distributed** normal variables where the variance σ_i^2 depends on X_i as follows:

$$\epsilon_i \sim N(0, \sigma_i^2), \quad \sigma_i^2 = \sigma^2 X_i^2,$$

where σ^2 is a constant.

i. Derive the estimators $\hat{\alpha}$ and $\hat{\beta}$ for α and β, respectively, that minimize the weighted sum of squared residuals:

$$S(\alpha, \beta) = \sum_{i=1}^{n} \frac{(Y_i - \alpha - \beta X_i)^2}{\sigma_i^2}.$$

ii. Discuss the differences between the derived estimators and those obtained using ordinary least squares (OLS).

(b) Suppose the errors ϵ_i are i.i.d. exponential random variables with a mean of 1. Given a dataset of 1,000 points, where the X_i values are uniformly distributed over the interval $[0, 100]$, determine the distribution of the least squares estimators $\hat{\alpha}$ and $\hat{\beta}$ for α and β fitted to Y on X. Provide a detailed explanation and compute the parameters of the distributions as precisely as possible.

3. [15] (Linear Classification) Suppose you are given the following dataset:

Example Number	X_1	X_2	Y
1	-1	2	-1
2	-2	-2	$+1$
3	1	-1	$+1$
4	-3	1	-1

Please perform the Batch Perceptron algorithm on this data. Assume you start with initial weights $\theta^T = [0, 0]$, bias $b = 0$, and that we pass all of our examples through in order of their example number.

(a) What would be the updated weight vector θ be after we pass example 1 through the Perceptron algorithm?

(b) What would be the updated bias b be after we pass example 1 through the Perceptron algorithm?

(c) What would be the updated weight vector θ be after we pass example 2 through the Perceptron algorithm?

(d) What would be the updated bias b be after we pass example 2 through the Perceptron algorithm?

(e) What would be the updated weight vector θ be after we pass example 3 through the Perceptron algorithm?

(f) What would be the updated bias b be after we pass example 3 through the Perceptron algorithm?

4. [20] (Multi class Classification) Consider the following multi-class classification problem:

   - We have a total of m classes and a total of l training examples.
   - The l training examples are uniformly split across the m classes, such that each class has $\frac{l}{m}$ examples.
   - All training examples are of dimensionality d.

   We will assess the efficiency of the two training paradigms of all-vs-all and one-vs-all under different learning scenarios.

   - **All-vs-All:** For every pair of labels $\langle i, j \rangle$, a classifier is learned over the following dataset: the examples labeled with one class $i \in \{1, \ldots, m\}$ are considered "positive", and those labeled with the other class $j \in \{1, \ldots, m\}, j \neq i$ are considered "negative".
   - **One-vs-All:** For every label $i \in \{1, \ldots, m\}$, a classifier is learned over the following dataset: the examples labeled with the label i are considered "positive", and examples labeled with any other class $j \in \{1, \ldots, m\}, j \neq i$ are considered "negative".

   ### (a) Training Complexity with $O(dn^2)$ Learning Algorithm

   We are given a magical black-box binary classification algorithm (we don't know how it works, but it just does!) which has a learning time complexity of $O(dn^2)$, where n is the total number of training examples supplied (positive+negative) and d is the dimensionality of each example.

   What are the overall training time complexities of the all-vs-all and the one-vs-all paradigms, respectively, and which training paradigm is most efficient?

   ### (b) Training Complexity with $O(d^2n)$ Learning Algorithm

   We are now given another magical black-box binary classification algorithm which has a learning time complexity of $O(d^2n)$, where n is the total number of training examples supplied (positive+negative) and d is the dimensionality of each example.

   What are the overall training time complexities of the all-vs-all and the one-vs-all paradigms, respectively, and which training paradigm is most efficient, when using this new classifier?

   ### (c) Evaluation Complexity

   Suppose we have learned an all-vs-all multi-class classifier and now want to proceed to predicting labels on unseen examples.

   We have learned a simple linear classifier with a weight vector of dimensionality d for each of the $\frac{m(m-1)}{2}$ classifiers ($w_i^T x = 0$ is the simple linear classifier hyperplane for each $i = 1, \ldots, \frac{m(m-1)}{2}$).

   We have two evaluation strategies to choose from. For each example, we can:

   - **Counting:** Do all predictions, then perform a majority vote to decide the class label.
   - **Knockout:** Compare two classes at a time; if one loses, never consider it again. Repeat until only one class remains.

   What are the overall evaluation time complexities per example for Counting and Knockout, respectively?

5. [15 + 15] Complete the attached notebooks. You are permitted to use chatbots or other resources for assistance, but you must ensure that you fully understand the code and implementations. During the online sessions, you may be asked to explain specific functions, code lines, and your overall approach. Be prepared to demonstrate your understanding in detail.