INTRODUCTION TO MACHINE LEARNING

Assignment 2

Generalised Linear Models, Perceptron, Bias-Variance Tradeoff

Deadline: October 4th, 11:59 p.m. IST **Max points: 100**

1. **(Properties of softmax)** Recall that $\text{softmax}(t_1, \cdots, t_k) := \left(\frac{e^{t_1}}{\sum_{i=1}^{k} e^{t_i}}, \cdots, \frac{e^{t_k}}{\sum_{i=1}^{k} e^{t_i}} \right)$. Prove/disprove the following statements:

   (a) $\text{softmax}(t_1, \cdots, t_k) = \text{softmax}(t_1 + c, \cdots, t_k + c)$ for any $c \in \mathbb{R}$.

   (b) $\text{softmax}(ct_1, \cdots, ct_k) = c.\text{softmax}(t_1, \cdots, t_k)$ for any $c \in \mathbb{R}$.

   (c) $\text{softmax}(t_1, \cdots, t_k) = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$ if and only if $t_1 = \cdots = t_k$.

   (d) Logistic regression can be seen as a special case of two-class softmax classification.

(3 + 2 + 3 + 2 = 10)

2. **(Programming - Iris Classification using Softmax Regression)**
   Develop a three-dimensional classification system using softmax regression to classify Iris flowers into their respective species based on their four measured features (sepal_length, sepal_width, petal_length, petal_width). **Dataset:** Iris Flower Dataset
   Source: UCI Machine Learning Repository
   URL: `https://archive.ics.uci.edu/ml/datasets/iris`
   Tasks:

   (a) **Data Preparation and Preprocessing:**

   (i) Load the Iris dataset.

   (ii) Examine the dataset structure and check for any missing values.

   (iii) Split the data into features (X) and target (y).

   (iv) Perform any necessary feature scaling or normalization.

   (v) Split the data into training (70%) and testing (30%) sets.

   (b) **Model Development**
   Implement a softmax regression classifier from scratch:

   (i) Define the softmax function.

   (ii) Develop the gradient descent algorithm for parameter optimization.

   (iii) Train the model on the training data.

   (c) **Model Evaluation:**

   (i) Use the trained model to predict species for the test set.

   (ii) Compute the misclassification fractions.

(3 + 10 + 7 = 20)

3. **(Programming - Perceptron)** Follow these instructions step by step to build a perceptron.

   (a) **Load** the file "perceptron_assignment.csv". There are three columns, namely, "x","y" and "result", with 500 data points.

   (b) **Plot** the data points in (x,y) pairs, and assign the points two different colours on the basis of the column "result".

(c) **Implement** the perceptron algorithm.

(i) Initialize $\theta_0 \in \mathbb{R}^2$ to $\begin{bmatrix}1 & 1\end{bmatrix}$, a row vector.

(ii) Set learning rate $\alpha = 0.5$

(iii) Perform the update step at time t

$$\theta_{t+1} = \theta_t + \alpha\left(result^{(i)} - \mathbb{I}\left(\theta_t \begin{bmatrix}x^{(i)} & y^{(i)}\end{bmatrix}^T \geq 0\right)\right)\begin{bmatrix}x^{(i)} & y^{(i)}\end{bmatrix}^T, \ i = 1, 2, ..., 500$$

(where $\mathbb{I}$ is the indicator function; recall that T in the superscript denotes the transpose operation), until $\theta_{t+1} = \theta_t$ for some t. Observe that this is where convergence is attained.

(iv) Report the final t, which is the number of steps required for the algorithm to converge at the learning rate α.

(v) Report the final value of θ at which convergence is attained.

(d) **Experiment** with $\alpha = 0.01, 0.1, 1$. Report in each case where your algorithm converges and how many steps it takes to do so. What differences (if any) do you see with the changing learning rate?

(2 + 2 + 8 + 3 = 15)

4. **(Programming - Best hypothesis among finitely many)** Consider the dataset that you used above. Recall that training error of a hypothesis is the fraction of training examples misclassified by it.

(a) On the same dataset, "perceptron_assignment.csv", find the training errors $\hat{\varepsilon}(h_i)$, $i = 1, 2, ..., 5$, for the following hypotheses, where $h_i\left(\begin{bmatrix}x & y\end{bmatrix}^T\right) = \mathbb{I}\left(\theta_i \begin{bmatrix}x & y\end{bmatrix}^T \geq 0\right)$, and

(i) $\theta_1 = \begin{bmatrix}0 & -1\end{bmatrix}$

(ii) $\theta_2 = \begin{bmatrix}0.65 & -0.22\end{bmatrix}$

(iii) $\theta_3 = \begin{bmatrix}0.9 & -1\end{bmatrix}$

(iv) $\theta_4 = \begin{bmatrix}0.7 & -0.5\end{bmatrix}$

(v) $\theta_5 = \begin{bmatrix}0.5 & -0.4\end{bmatrix}$

(b) Identify the hypothesis $\hat{h}$ among the above which has the least training error $\hat{\varepsilon}(\hat{h})$.

(c) Given the current dataset of 500 training examples, the training error $\hat{\varepsilon}(\hat{h})$ of the best hypothesis $\hat{h}$ among the 5 given hypotheses, and $\delta = 0.01$, we can say that the generalisation error $\varepsilon(\hat{h})$ of $\hat{h}$ lies in an interval $[\lambda_1, \lambda_2]$ with probability $1 - \delta$. Find λ_1 and λ_2. How would λ_1 and λ_2 change if we wanted to bound the error margin between generalisation error $\varepsilon(\hat{h})$ and $\varepsilon(h^*)$ (h^* being the hypothesis among the five that minimises the generalisation error)?

(5 + 1 + 4 = 10)

5. **(Programming - Feature Selection and Model Complexity Analysis)** The Taiwan Real Estate dataset contains information about housing prices and various features that might influence these prices. Explore the relationship between different features and housing prices.
Dataset: Taiwan Housing dataset from the UCI Machine Learning Repository.
Dataset URL: `https://archive.ics.uci.edu/dataset/477/real+estate+valuation+data+set`
Tasks A. Data Preparation:

(i) Load the Taiwan real estate dataset.

(ii) Create three different train-test splits as follows:

- 80% train, 20% test
- 50% train, 50% test

- 10% train, 90% test

(iii) Use 'Y' (house price of unit area) as the target variable.

(iv) Select the following five features for analysis:

- X1: House age
- X2: Distance to the nearest MRT station
- X3: Number of convenience stores in the living circle
- X4: Latitude
- X5: Longitude

B. Forward Feature Selection (for 80:20 split):

(i) Univariate Feature Selection:

- Using the 80:20 split dataset, perform 5 separate simple linear regression analyses.
- For each analysis, use one of the features (X1 to X5) as the predictor variable and house price (Y) as the response variable.
- Calculate the training error (mean squared error) for each model.
- Identify the feature that yields the lowest training error on the training set.

(ii) Incremental Feature Addition:

- Start with the best feature identified in step (i).
- For each remaining feature:
  - Add it to your current set of predictors.
  - Train a multiple linear regression model using the current set of predictors.
  - Calculate the training error (mean squared error) for this model.
- Select the feature that results in the lowest training error when added. Repeat this process, adding one feature at a time, until all five features are included.

C. Model Evaluation: For each of the five train-test splits, and for each step of the feature selection process (1 to 5 features), calculate:

(i) Training error (MSE)

(ii) Testing error (MSE)

D. Visualization: For each of the three train-test splits, create a plot showing:

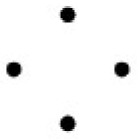
(i) Training and testing errors vs. number of features

(ii) Ensure each plot is properly labeled with title, axis labels, and legend.

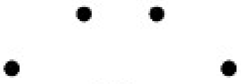
(4 + 12 + 3 + 3 = 22)

6. **(VC dimension)** Suppose that we are to use semicircles in the 2D plane to classify a given collection of 2-dimensional data points (diameters of these semicircles need not be parallel to either coordinate axis). Each semicircle classifier labels points in its interior as 0, and other points as 1. Let us call this collection of classifiers $\mathcal{H}$

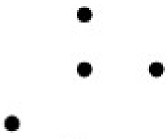
(i) Which of the following point sets can $\mathcal{H}$ shatter?



(a)



(b)



(c)

(ii) What does this tell you about the VC dimension of $\mathcal{H}$?

(6 + 1 = 7)

7. **(Single parameter exponential family)** Recall that distributions in the single parameter exponential family can be expressed as

$$p(y, \eta) = b(y)e^{\eta^T T(y) - a(\eta)}$$

A random variable Y is said to follow the Poisson distribution with parameter $\lambda > 0$ if its probability mass function is given by

$$\mathbb{P}(Y = y) = \frac{e^{-\lambda}\lambda^y}{y!}, \ y = 0, 1, 2, ...$$

(a) Show that Poisson distributions belong to the single parameter exponential family.

(b) Suppose you are given a data set $\mathcal{S} = \{(x^{(i)}, y^{(i)}), \ x^{(i)} \in \mathbb{R}^k, \ y^{(i)} \in \mathbb{Z}^+, \ i = 1, 2, ..., n\}$, and it is assumed that given x, y follows a Poisson distribution with rate $e^{\theta^T x}$.

(i) Write down the log-likelihood function of θ, given $\mathcal{S}$.

(ii) Compute the gradient of the log-likelohood.

(iii) Use this gradient to set up the gradient descent method to find the maximum-likelilihood estimator of $\theta \in \mathbb{R}^k$.

(2 + 6 = 8)

8. (a) **(Markov's Inequality)** For a non-negative random variable X with probability density function f, and any $a > 0$, prove that

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}(X)}{a}.$$

(**Hint:** Use $\mathbb{E}(X) = \int_0^\infty xf(x)dx$, and the fact that $x \geq a$ in $[a, \infty)$)

(b) Use a similar analysis to prove that

$$\mathbb{P}(X \leq a) \geq \frac{\mathbb{E}(X.\mathbb{I}(X \leq a))}{a}$$

where $\mathbb{I}$ is the indicator function.

(c) **(Chebyshev's Inequality)** *Markov's inequality is a tool that is used in proving several concentration bounds, including Hoeffding's inequality. Among the many applications of Markov's inequality, perhaps the simplest to prove is Chebyshev's inequality.*

Prove that for any random variable X with expectation $\mu < \infty$ and variance $\sigma^2 > 0$,

$$\mathbb{P}(|X - \mu| \geq n\sigma) \leq \frac{1}{n^2} \ \forall n > 0.$$

(3 + 2 + 3 = 8)