Assignment 1A

Introduction to Machine Learning: CS-3410-1 (Spring 2025)

February 23, 2025

Question 1:
A Poisson-distributed random variable $X \sim \text{Poisson}(\lambda)$ has the probability mass function (PMF):

$$p(x \mid \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x \in \{0, 1, 2, \dots\}, \lambda > 0. \tag{1}$$

1. Write the probability mass function (PMF) of the Poisson distribution as a member of the Exponential Family. Identify $b(x)$, the natural parameter η and the log-partition function $A(\eta)$. The general form of an exponential family distribution is:

$$p(x \mid \eta) = b(x) \cdot \exp\left[\eta \cdot x - A(\eta)\right]. \tag{2}$$

2. We know that $\mathbb{E}[X] = A'(\eta)$ and $\text{Var}(X) = A''(\eta)$.
   (a) Compute $A'(\eta)$ and verify that it equals $\mathbb{E}[X]$.
   (b) Compute $A''(\eta)$ and verify that it equals $\text{Var}(X)$.

NOTE:

$$\mathbb{E}[X] = \sum_{x=0}^{\infty} x \cdot p(x). \tag{3}$$

$$\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2. \tag{4}$$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{5}$$

Question 2:
Let $X = [X_1, X_2, \dots, X_N]^T$ be N correlated random variables with mean vector μ and covariance matrix Σ. Define a new random variable Z as a linear combination of these random variables:

$$Z = \sum_{i=1}^{N} \omega_i X_i = \omega^T X = \begin{bmatrix} \omega_1 & \omega_2 & \dots & \omega_N \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \tag{6}$$

The mean vector μ of the random variables $X_1, X_2, \dots, X_N$ is defined as:

$$\mu = \mathbb{E}[X] = \begin{bmatrix} \mathbb{E}[X_1] \\ \mathbb{E}[X_2] \\ \vdots \\ \mathbb{E}[X_N] \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}. \tag{7}$$

The covariance matrix Σ is defined as:

$$\Sigma = \text{Cov}(X) = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \dots & \text{Cov}(X_1, X_N) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \dots & \text{Cov}(X_2, X_N) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_N, X_1) & \text{Cov}(X_N, X_2) & \dots & \text{Var}(X_N) \end{bmatrix}. \tag{8}$$

(a) Show that the expectation (mean) of Z is given by:

$$\mathbb{E}[Z] = \omega^T \mu \tag{9}$$

(b) Show that the variance of Z is given by:

$$\text{Var}(Z) = \omega^T \Sigma \omega \tag{10}$$

(c) The correlation coefficient $\rho_{1,2}$ between two random variables X_1 and Y_2 is defined as:

$$\rho_{X_1,X_2} = \frac{\text{Cov}(X_1, X_2)}{\sigma_{X_1}\sigma_{X_2}} \tag{11}$$

Using the Cauchy-Schwarz inequality, derive the following bound for the correlation coefficient.

$$-1 \leq \rho_{1,2} \leq 1 \tag{12}$$

NOTE: Cauchy-Schwarz Inequality for X_1 and X_2 -

$$|\mathbb{E}[X_1 X_2]| \leq \sqrt{\mathbb{E}[X_1^2] \cdot \mathbb{E}[X_2^2]}. \tag{13}$$

(d) Derive the value of the following integral:

$$\int_{-\infty}^{\infty} x\, f(x)\, dx \tag{14}$$

Wherein:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{15}$$

Question 3:
Consider a training set where each example has multiple outputs:

$$\{(x^{(i)}, y^{(i)}), \quad i = 1, \dots, m\}$$

wherein:

$$x^{(i)} \in \mathbb{R}^n, \quad y^{(i)} \in \mathbb{R}^p.$$

For each training example, $y^{(i)}$ is vector-valued with p components. Consider a linear model for predicting the outputs, formulated as follows, where $\Theta \in \mathbb{R}^{n \times p}$ is the parameter matrix.

$$y = \Theta^T x, \tag{16}$$

(a) The cost function is given by $J(\Theta)$. Write $J(\Theta)$ in matrix-vector notation (i.e., without using any summations).

$$J(\Theta) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p}\left((\Theta^T x^{(i)})_j - y_j^{(i)}\right)^2. \tag{17}$$

Hint: Start with the $m \times n$ design matrix:

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \tag{18}$$

And, the $m \times p$ target matrix:

$$Y = \begin{bmatrix} (y^{(1)})^T \\ (y^{(2)})^T \\ \vdots \\ (y^{(m)})^T \end{bmatrix} \tag{19}$$

Then, express $J(\Theta)$ in terms of these matrices.

(b) Find the closed-form solution for Θ that minimizes $J(\Theta)$. This corresponds to solving the normal equations for the multivariate least squares case.

(c) Suppose instead of considering the multivariate vectors $y^{(i)}$ all at once, we instead compute each variable $y_j^{(i)}$ separately for each $j = 1, \dots, p$. In this case, we have p individual linear models of the form ($\theta_j \in \mathbb{R}^n$):

$$y_j^{(i)} = \theta_j^T x^{(i)}, \quad j = 1, \dots, p. \tag{20}$$

How do the parameters from these p independent least squares problems compare to the multivariate solution?

Question 4: Coding assignment
Objective: To derive the mathematical formulation of a multiclass logistic regression model and implement it in Python.
Dataset: The Iris Flower Dataset consists of 150 samples of iris flowers, each described by four features: Sepal Length (cm), Sepal Width (cm), Petal Length (cm), and Petal Width (cm), with the goal of classifying them into one of three species: Iris Setosa, Iris Versicolor, or Iris Virginica. The dataset consists of 150 samples, with 50 samples per class. Each sample belongs to one of three classes (0, 1, 2), representing the flower species. The dataset can be accessed at: `UCI Machine Learning Repository` or load it directly using `sklearn`.
Workflow:

(a) Load the Dataset

1. Load the Iris dataset using `sklearn.datasets.load_iris()`.
2. Extract all four features as input and extract the class labels (0, 1, or 2).

(b) Preprocess the Data

1. Split the dataset into training (60%) and test (40%) sets.
2. Implement the model separately using three approaches: normalizing the feature values, standardizing the feature values, and leaving the feature values unprocessed.
   Normalization (Min-Max Scaling) rescales feature values to the range $[0, 1]$.

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{21}$$

   Standardization (Z-score Normalization) transforms feature values so that they have a mean of 0 and a standard deviation of 1.

$$x_{\text{std}} = \frac{x - \mu}{\sigma} \tag{22}$$

   Here, μ is the mean and σ is the standard deviation of x.

(c) Implement Softmax Function

1. Write a function that applies the softmax transformation to a set of logits.

(d) Implement Multiclass Logistic Regression

1. Use the softmax function to compute probabilities.
2. Implement the categorical cross-entropy loss function.
   For a dataset with m samples, the average cross-entropy loss is defined as:

$$L = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} y_{ij} \log(\hat{y}_{ij}) \tag{23}$$

   where:

   - L is the average cross-entropy loss over all samples.
   - m is the total number of samples in the dataset.
   - y_{ij} is the j-th component of the one-hot encoded true label for the i-th sample.

- $\hat{y}_{ij}$ is the j-th component of the predicted probability for the i-th sample.

3. Train the Model
    1. Implement gradient descent and stochastic gradient descent (separately) to train the model.
    2. Track the loss during training.
4. Evaluate the Model
    1. Compute accuracy on the test set.
    2. Generate a classification report (precision, recall, F1-score) and confusion matrix.
5. Visualize the Training Performance
    1. Plot the loss curve over epochs.

Question 5: Coding assignment
Objective: To explore the differences between linear and polynomial regression, evaluate their performance, and analyze key concepts such as overfitting, underfitting, and the bias-variance tradeoff.
Dataset: Use the Concrete Compressive Strength Dataset, which provides data on concrete strength based on its composition. The dataset can be accessed at: `Concrete Compressive Strength Dataset`
Workflow:

(a) Data Loading & Preprocessing

1. Download the dataset and load it into a Pandas DataFrame.
2. Display the first few rows to understand its structure.
3. Identify:
    - Independent variables (features)
    - Dependent variable (compressive strength)
4. Check for missing values and handle them appropriately.
5. Split the dataset into 80% training and 20% testing.

(b) Implementing Linear Regression

1. Train a Linear Regression model using `scikit-learn`.
2. Evaluate the model's performance using:
    - Mean Squared Error (MSE):
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
    - Coefficient of Determination (R^2):
$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$
3. Plot the predicted vs. actual values to visualize model fit.

(c) Implementing Polynomial Regression ($k = 2, 3, 4$)

1. Transform features to include polynomial terms using `PolynomialFeatures` from `sklearn.preprocessing`.
2. Train Polynomial Regression models for degrees $k = 2, 3, 4$.
3. Evaluate performance using:
    - Mean Squared Error for degree k:
$$\text{MSE}_k = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{(k)})^2$$
    - Coefficient of Determination for degree k:
$$R_k^2 = 1 - \frac{\sum(y_i - \hat{y}_i^{(k)})^2}{\sum(y_i - \bar{y})^2}$$
4. Plot the predicted vs. actual values for each polynomial degree.

(d) Visualizing & Comparing Results

1. Overlay the linear regression line and polynomial regression curves on the scatter plot of the actual data.
2. Compare the MSE and R^2 scores for all models.

(e) Bias-Variance Tradeoff Analysis

1. Discuss the bias-variance tradeoff based on the results:
   - Which model has high bias and low variance?
   - Which model has low bias but high variance?
   - Which model achieves a balance between bias and variance?
2. Explain why higher-degree polynomials tend to overfit the data.

Question 6: Coding assignment
Objective: To simulate and fit a normal distribution to data and analyze the impact of outliers.

(a) Data Simulation:

1. Simulate a dataset of 1000 samples drawn from a univariate normal distribution with mean $\mu = 50$ and standard deviation $\sigma = 10$.
2. Plot a histogram of the simulated data.

(b) Normal Distribution Fitting:

1. Using the dataset from part 1, fit a normal distribution by estimating the parameters (mean and standard deviation) using Maximum Likelihood Estimation (MLE).
2. Overlay the probability density function (PDF) of the fitted normal distribution on the histogram.

(c) Handling Outliers:

1. Simulate an additional 50 samples from a uniform distribution over the interval $[100, 150]$ and add these to your original dataset, introducing outliers.
2. Fit a normal distribution to this new dataset (which now contains outliers) using MLE.
3. Compare and discuss how outliers affect the estimated parameters, and describe an approach to detect any outliers in the dataset.