Assignment 1

UMC 203: Artificial Intelligence and Machine Learning

February 2024

No copying is allowed. **Thorough plagiarism check will be run.** You are given four questions, each of which requires Python programming. **You will be using jupyter notebook to write your Python programs. You will use Google Colab to write jupyter notebooks.**

SUBMISSION INSTRUCTIONS

1. You should submit **four files** (NOT a zip file) with the following naming convention.
   - ▷ `AIML_2024_A1_LastFiveDigitsOfSRNumber.pdf` → Answers to all the problems.
   - ▷ `AIML_2024_A1_LastFiveDigitsOfSRNumber.ipynb` → Code for the all the problems.
   - ▷ `AIML_2024_A1_LastFiveDigitsOfSRNumber_q2_w.csv` → Output labels for question-2 e.
   - ▷ `AIML_2024_A1_LastFiveDigitsOfSRNumber_q4test.csv` → Output labels for question-4.

   For example, if the last five digits of your SR Number is 20000, then you should submit four files: `AIML_2024_A1_20000.pdf`, `AIML_2024_A1_20000.ipynb`, `AIML_2024_A1_20000_q2_w.csv`, `AIML_2024_A1_20000_q4test.csv`.
2. **Any deviation from the above rule will incur serious penalty!**
3. For the coding questions, you are asked to report some values, e.g., the number of iterations. These values should be reported in the `.pdf` file you submit.
4. At the top of the `.pdf` file you submit, write your name and SR Number.
5. You will get a bonus of 10% if your reports are typed neatly in LaTeX

ORACLE ACCESS INSTRUCTIONS

1. Look at `https://colab.research.google.com/` to familiarize yourself with Google Colab.
2. The instructions on accessing the Oracle are provided with the jupyter notebook template provided to you.
3. You will be provided a jupyter notebook `AIML_A1.ipynb`, which will act as a template and contain instructions on loading the Oracle.
4. All the datasets and oracles will be provided to you in the zip file titled `AIML_A1.zip`. You will need to upload this file to your drive.

1 Naive Bayes Classifier (10 marks)

Recall the Bayes classifier; the critical component of a Bayes classifier is computing the posterior. However, this computation scales with the dimension of the data. This can be seen in, $P(y, \mathbf{x}) = P(y, x_1, x_2, ..., x_d) = P(y)P(x_1|y)P(x_2|x_1, y)...P(x_d|x_1, x_2, ..., x_{d-1}, y)$, where $\mathbf{x} = (x_1, ..., x_d)$ is a d-dimensional vector, and y denotes the class label. To get around this issue, we make a simplifying assumption(A1 is the reason this technique is called naive):

(A1) Assume each coordinate of the vector is mutually independent, i.e.,
$P(x_i|x_1, ..., x_{i-1}, x_{i+1}, ..., x_d, y) = P(x_i|y)$. Therefore, $P(\mathbf{x}, y) = P(y)\Pi_{i=1}^{d} P(x_i|y)$

(A2) Assume that class-conditioned distributions are normally distributed, i.e.,
for $j \in \{0, 1\}$, $(X|Y = j) \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ where $X \in \mathbb{R}^d$

In class, you have studied the Bayes classifier under the $0 - 1$ loss function, which assumes that misclassification of any kind is punished equally. However, in the real world, some mistakes are costlier than others. Think of an intruder detection algorithm that fails to catch someone breaking in versus making an annoying beep whenever the dog snores too loud. We capture this by setting $l(0, 1)$ and $l(1, 0)$ to different values. We will call this new loss function the *modified loss function*. Please use the function `q1_get_loss` described below to obtain these values.

You are now given a dataset with **5-dimensional** features and asked to learn a classifier that minimizes the modified misclassification loss. You decide to make assumptions (A1) **and** (A2) and build the Bayes classifier for the modified loss. Please use the function `q1_get_train_set` and `q1_get_test_set` described below to obtain these datasets.

1.a (1 mark) The parameters for your model are $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1$. Using assumption (A1), what can you say about $\boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1$?.

1.b (1 mark) State the Bayes classifier under the modified loss function.

1.c (2 marks) Obtain sample estimates of the class conditional means and variances for your model with $n = 2, 10, 20, 50, 100, 500, 1000$ samples. Write down the estimates for all the parameters in a table for different values of n.

Hint: The Maximum Likelihood Estimate for the parameters of a one dimensional Gaussian distribution $x^{(i)} \sim \mathcal{N}(\mu, \sigma^2)$ are given by,

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}$$

$$\hat{\sigma_n^2} = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - \hat{\mu}_n)^2$$

1.d (3 marks) Compute and report the misclassification loss under the modified loss with the test set provided by the oracle for each Bayes classifier obtained with the different parameter estimates. Write down these losses in a table.

You are now provided 2000 samples each from 2 classes of the CIFAR10 dataset and asked to learn a classifier that minimizes the modified misclassification loss. Again, you build a classifier under assumptions (A1) and (A2). Note: Each image from CIFAR10 is represented as a **3072-dimensional** vector. Please use the function `q1_get_cifar10_train_test` described below to obtain this dataset.

1.e (3 marks) Build a Bayes classifier under the modified loss function after computing class conditional estimates of the parameters and report the test accuracy on the test set provided.

The Oracle file provided to you, `AIML_A1.py`, contains the following functions related to question 1.

- `q1_get_loss(srn)` returns a tuple with the values for $l(0,1)$ and $l(1,0)$.
- `q1_get_train_set(srn,num_samples)` returns your dataset. It returns a list of 2-tuples. The first element of the tuple is a 5-dimensional numpy array, and the second element indicates the class and is either 0 or 1.
- `q1_get_test_set(srn)` returns a test set on which you must report the accuracy. The format of the data is similar to the previous function.
- `q1_get_cifar10_train_test(srn)` returns a tuple containing the training set and test set for the second part of the first question. Each dataset is in the same format as before.

2 Perceptron Algorithm (10 marks)

Recall that the Perceptron algorithm is guaranteed to converge when the data is linearly separable. You will need to implement the perception algorithm. Use the Oracle `q2_perceive(srn)` to obtain data $\mathcal{D}$.

- Report the following:

  2.a (2 marks) The value of w obtained using the perceptron algorithm.

  2.b (1 mark) Number of errors made by the algorithm.

  2.c (1 mark) Report the Margin given by your final classifier.

  2.d (1 mark) Radius of the data set.

$$R = \max_{i \in (1, |\mathcal{D}|)} ||x_i||$$

- Use the Oracle `q2_mnist(srn)` to obtain linearly separable samples from the mnist data $\mathcal{D}_{\text{mnist}}$. Run the perceptron algorithm on this dataset. Report the following:

  2.e (2 marks) The value of w obtained using the perceptron algorithm. You must report this vector in a comma-separated file named `AIML_2024_A1_LastFiveDigitsOfSRNumber_q2_w.csv`

  2.f (1 mark) Number of errors made by the algorithm.

  2.g (1 mark) Report the Margin given by your final classifier.

  2.h (1 mark) Radius of the data set.

$$R = \max_{i \in (1, |\mathcal{D}_{\text{mnist}}|)} ||x_i||$$

The Oracle file provided to you, `AIML_A1.py`, contains the following functions related to question 2.

- `q2_perceive(srn)` returns a dataset. It returns a list of 2-tuples. The first element of the tuple is a 5-dimensional numpy array, and the second element indicates the class and is either +1 or -1.
- `q2_mnist(srn)` returns a dataset. It returns a list of 2-tuples. The first element of the tuple is a 784-dimensional numpy array, and the second element indicates the class and is either +1 or -1. Your data is linearly separable samples from the famous MNIST dataset (Wikipedia).

3 Fisher Linear Discriminant (10 marks)

One way to view a linear classification model is dimensionality reduction. As seen in class, the Fisher Linear discriminant projects the data into lower dimensions, ensuring a large separation between the projected class means while giving a small variance within each class, thereby minimizing the class overlap.

The Fisher Linear Discriminant is given by: $\mathbf{w}^T x = b$ where $\mathbf{w}$ is the Projection vector, and b is the threshold for classification.

3.1 Fisher Discriminant on IRIS Dataset

IRIS dataset is one of the earliest datasets available on UCI Machine Learning repository. Also known as Fisher's Iris data set is a multivariate data set used and made famous by the British statistician and biologist Ronald Fisher in his 1936 paper "The use of Multiple Measurements in Taxonomic Problems" as an example of linear discriminant analysis. It is used in the literature on classification methods and is widely used in statistics and machine learning. The data set contains three classes of 50 instances each, where each class refers to a type of iris plant.

Your task is to create a Fisher linear classifier on the iris dataset. You will be given the train set of the iris dataset containing two classes, namely 'setosa' and 'vertsicolor.'

3.1.a (3 marks) Find the normalized projection vector $\mathbf{w}$ as per the idea proposed in Fisher discriminant to project the data to one dimension and the threshold b to classify the two classes such that the error on train data is minimal. Report the vector $\mathbf{w}$ and the threshold b in the PDF file.

3.1.c (1 mark) Plot the histogram of the projected data with different colors for each class. Also, plot the classifier boundary obtained using the above threshold on the same plot. Add this plot in the PDF File.

3.2 Fisher Discriminant on Dataset given by Oracle

The Oracle file provided to you, `AIML_A1.py`, contains the following functions related to question 3.

`q3_get_data(srn)`: returns training dataset. It returns a list containing all the data points. Each entry of the list is a tuple of two elements; the first element is the 2-dimensional data $[x_1, x_2]$, and the second element is the label (1 or 0).

`q3_get_test_data(srn)`: returns test dataset. It returns a list containing all the data points. Each entry of the list is a tuple of two elements; the first element is the 2-dimensional data $[x_1, x_2]$, and the second element is the label (1 or 0).

Answer the following questions.

3.2.a (1 mark) Find the normalized projection vector $\mathbf{w}$ using fisher discriminant. plot the histogram of projected data. Give different colors for both classes. Is the projected data is linearly separable?

3.2.b (3 marks) Add two more dimensions to each data point to get a 4-dimensional dataset. call it as $\phi(x)$

Ex:- if $x = [x_1, x_2]$ then $\phi(x) = [x_1, x_2, x_1^2, x_2^2]$

Now use this new dataset $\phi(x)$ with the same labels to find normalized projection vector $\mathbf{w}$. Is the projected data linearly separable?

If the data is linearly separable, Find the threshold b such that the error on the training dataset is minimal. Report the vector $\mathbf{w}$ and the threshold b in the PDF file.

3.2.c (1 mark) Plot original data and the classifier boundary $\mathbf{w}^T\phi(x) = b$. If any entry of $\mathbf{w}$ is less than 10^{-6}, you can assume it to be zero. Add this plot to the PDF file. [**Hint** : Ideally, the first two entries w_0 and w_1 should be close to zero, hence we can ignore them.]

3.2.d (1 mark) Load the test dataset using the Oracle function. Find test accuracy with thresholds = $[b-0.3, b-0.2, b-0.1, b, b+0.1, b+0.2, b+0.3]$ where b is threshold found above. Report these values as a table in the PDF file.

4 Logistic regression (10 marks)

You will now use logistic regression on the Diabetes dataset, a real real-world dataset obtained from Kaggle. You will need to evaluate its performance of logistic regression using K-Fold Cross Validation. You will measure the performance using metrics like the confusion matrix.

4.1 Logistic Regression

The logistic function, denoted by $\sigma(z)$, is defined as: $\sigma(z) = (1 + e^{-z})^{-1}$. Where z is the linear combination of the predictor variables and their coefficients. It's the equation of the form:

$$z = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n = b_0 + b^T x$$

where b_0 is the intercept and $b_1, b_2, \ldots, b_n$ are the coefficients of the predictor variables $x_1, x_2, \ldots, x_n$. $\sigma(z)$ maps a real number z to $[0, 1]$. We can choose to interpret this as the probability of the positive class. The probability of the negative class is $1 - \sigma(z)$. Logistic Regression is a techinque to find coefficients $b_0, b_1, b_2, \ldots, b_n$. For more details check here. You can use Logistic Regression function from sklearn by using the following function. `from sklearn.linear_model import LogisticRegression`

4.2 K-Fold Cross Validation

K-Fold cross-validation is a technique used to assess the performance of a machine learning model. It involves splitting the dataset into K subsets (or folds), training the model K times each time using $K - 1$ folds for training and one fold for validation, and then averaging the performance metrics across all iterations. This helps obtain a more robust estimate of the model's performance and reduces the risk of overfitting; for more details, check here

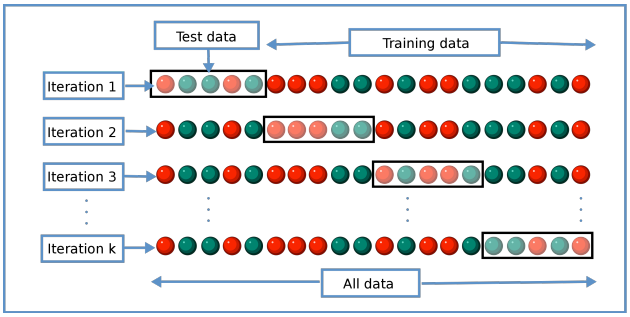

Figure 1: K-Fold Cross Validation

Using the above information, answer the following questions

4.a (4 Marks) Perform K-Fold Cross Validation with K = 6 and run logistic regression on the dataset using scikit-learn.

4.b (3 Marks) Evaluate the K models using a confusion matrix; for more details, check here and calculate the following metrics:

(a) Recall: Proportion of true positive predictions among all actual positive instances

(b) Precision: Proportion of true positive predictions among all positive predictions

(c) Accuracy: Proportion of correctly classified instances among the total instances

(d) F1-score: Harmonic mean of precision and recall, providing a balanced measure of model performance

4.c (1 Mark) Plot barplot of all Metrics for all K Models (Make every metric in $[0, 1]$).

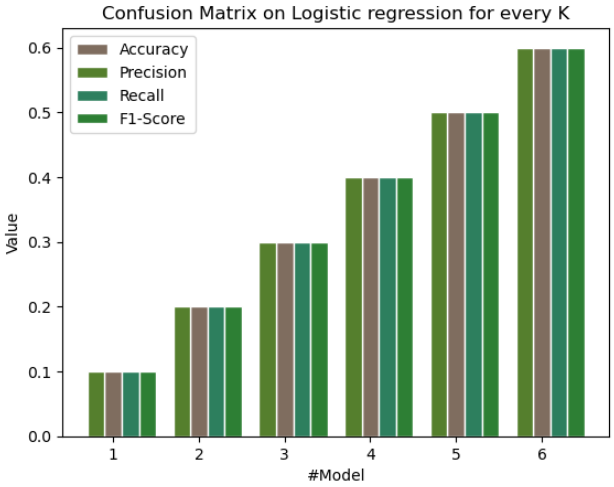

Figure 2: Example plot

4.e (2 Marks) Apply the trained model to the test data provided on the day of submission(25th Feb 2024) and submit the test results as `AIML_2024_A1_LastFiveDigitsOfSRNumber_q4test.csv`