CSCC11 – Introduction to Machine Learning and Data Mining

Assignment 2

Logistics

- This assignment is worth 13% of the course grade and is due on Dec 4, 2023.
- It can be done individually or in groups of two. Either way, you are required to self-signup to one of the A2 groups on Quercus and provide a single archive file (.rar or .zip) containing all the files/documents related to your solution.
- The answers/code you submit (excluding the starter code) must be your own.
- A starter code has been provided to help you complete this assignment. Please do not change the headers of the given Python functions.
- We prefer that you ask any related questions during the tutorial sessions or on Piazza. Otherwise, for your first point of contact, please reach out to your TAs:

  Arash Rasti Meymandi, through email: arash.rasti@mail.utoronto.ca
  Anindro Bhattacharya, through email: anindro.bhattacharya@mail.utoronto.ca
  Noor Nasri, through email: noor.nasri@mail.utoronto.ca

Question 1 [1 point]

Consider an ensemble binary classification technique that is based on N independent classifiers $\{C_i\}_{i=1}^{N}$ where N is odd. For a given data point, each base classifier C_i predicts the class label and the most commonly predicted class label across all base classifiers is selected as the final prediction. Assuming that all base classifiers have an error rate ε (i.e. the probability of making a wrong prediction), express the error rate of the ensemble technique in terms of ε and N, and explain your answer.

Question 2 [1 point]

Consider a binary classification scenario with one-dimensional data and Gaussian class-conditional densities. That is, $p(x|c_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(\frac{-(x-\mu_i)^2}{2{\sigma_i}^2}\right) where\ i \in \{1,2\}$.

Prove that, if $\sigma_1 = \sigma_2$, then the decision boundary is a straight line.

Question 3 [2 points]

Assume we want to build a logistic regression model to classify fruit as orange/non-orange using its width and height. The training data to be used is as follows:

Width	Height	Orange
4	4	Yes
6	4	Yes
6	5	Yes
6	8	No
6	10	No
8	8	Yes
8	10	No

a) Write the corresponding optimization problem in terms of the data provided above and specify the parameters to be estimated.

b) Perform 3 iterations of the gradient descent algorithm to determine the parameters assuming that the step size (λ) is 0.01 and the initial estimate is $[0.3, -0.2, 0.7]^T$ (note that 0.7 corresponds to the bias). For each estimate, including the initial one, you are required to report the following:

- The value of the estimate
- The accuracy of the resulting logistic regression model when applied to the training data

Note that you do not need to do the computations manually. You might want to use a spreadsheet or write a simple code to do that.

c) Classify the following data points using the model you obtained in part b: (3, 3), (4, 10), (9, 8), and (9, 10).

d) Discuss one advantage and one disadvantage of logistic regression.

Question 4* [4 points]

Using this dataset from the BBC, you are required to implement and evaluate two different classifiers to label news articles according to five categories: business, entertainment, politics, sport, and tech. Your code should not make use of any existing implementation of these classifiers.

The first step is to download the pre-processed dataset and inspect the different files to understand how to read the information in your code:

- The dataset includes 2225 articles listed in the *bbc.docs* file.
- The data has been pre-processed using stemming, stop-word removal, and low term frequency filtering. This resulted in 9635 terms listed in the *bbc.terms* file.
- Each row in *bbc.mtx*, except the first two, represents the frequency of a term in a given article. For example, row 812 ("2 528 5.0") indicates that term 2 ("sale") occurs 5 times in article 528 (*entertainment.018*).

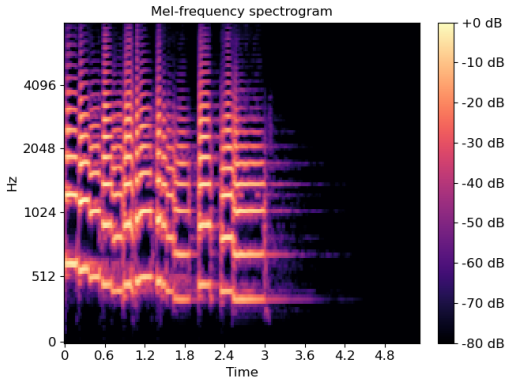
a) Write Python code that does the following:
   i. Reads the dataset and partitions it into training and testing subsets.
   ii. Trains a Naïve Bayes classifier using the training subset. Note that you need to represent the occurrence of each term as a binary value instead of a frequency. In addition, to avoid dealing with zero probabilities for words that do not occur in a particular class of articles, you can adjust the conditional probabilities $p(x_i|y)$ by adding 1 to the numerator and 2 to the denominator.
   iii. Determines the classification accuracy by computing the percentage of labels returned by your classifier that agrees with the labels attached to the dataset. This step should be done for the training and testing subsets separately. In other words, you need to compute the accuracy of your classifier when applied to the training subset (i.e. when used to predict the labels of the training data points) and then do the same for the testing subset. Keep in mind that the classifier should be built using the training subset only.

b) Repeat the steps of part a) assuming a Gaussian class conditionals classifier with frequency-based features instead of binary ones. For this part, you need to consider five Gaussian components (one for each category of articles) whose parameters should be determined using Maximum Likelihood Estimation.

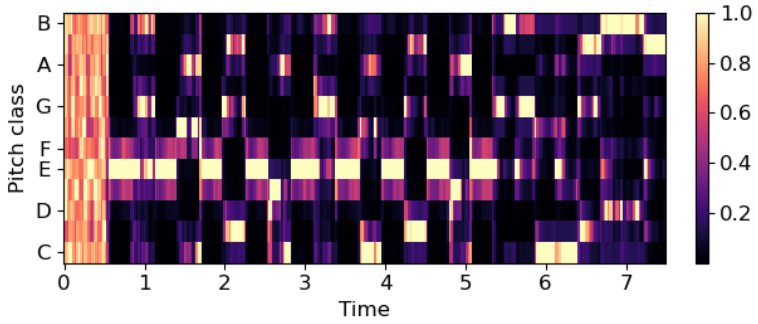
* The original version of this question was prepared by Prof. Francisco Estrada

Question 5 [5 points]

The *Mel spectrogram* is a frequency-based numerical representation of an audio signal. It is usually generated by fragmenting the signal into several windows, applying a Fourier Transform to identify the frequency components, and computing the amplitudes of the different frequencies. This representation can be visualized by mapping amplitude values to colours as follows[†]:



On the other hand, a *chromagram* is a related representation that aims at characterizing the pitch profile of a signal according to the twelve pitch classes used in music: C, C#, D, D#, E, F, F#, G, G#, A, A#, and B. The following is an example of a chromogram that shows the variations in the intensity of each chroma feature over time[‡]:



Requirements

Your task involves implementing and evaluating a *k-Nearest Neighbors* (*k-NN*) classifier to predict whether the emotion associated with a human audio signal is *positive* or *negative* based on Mel spectrogram and chromagram features, where the value of *k* is determined using cross-validation. In addition, you are required to repeat the whole process using a low-dimensional

[†] https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html
[‡] https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html

representation of the data obtained using *Principal Component Analysis* (*PCA*), and then compare the results in terms of accuracy and runtime.
Note that your code should not make use of any existing implementation of *k-NN* or *PCA*.

Dataset

We will use the audio speech dataset from *The Ryerson Audio-Visual Database of Emotional Speech and Song*[§]. This dataset includes 1440 audio files vocalized by 24 professional actors (60 files each). The files are named according to the following identifiers:

- Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
- Vocal channel (01 = speech, 02 = song).
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
- Emotional intensity (01 = normal, 02 = strong).
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

For example, the file named *03-01-06-01-02-02-05.wav* corresponds to the following:

- Modality: audio-only
- Vocal channel: speech
- Emotion: fearful
- Emotional intensity: normal
- Statement: "Dogs are sitting by the door"
- Repetition: 2nd repetition
- Actor: 05

You can use the starter code provided with this assignment to read the dataset and generate the Mel spectrogram and chromogram features. Regarding the emotions, we will only consider “calm”, “happy”, “angry”, and “fearful” (i.e. any audio data that is not associated with any of these emotions should be discarded). Furthermore, we will combine the emotions into two broad classes: *positive* which includes “calm” and “happy”, and *negative* which includes “angry” and “fearful”.

[§] https://zenodo.org/record/1188976#.Y3qkx3bMLIU

a) *k-NN* Classification using the Original Dataset

In this part, you are required to write Python code that implements a *k-NN* classifier to determine whether a given audio signal conveys a *positive* or *negative* emotion. You can use 70% of the audio dataset for training (T_1) and 30% for testing (T_2).

- Use cross-validation on T_1 to determine the most appropriate value of k within a set of values of your choice (a minimum of three)
- Using the k value obtained in the previous step, determine the accuracy and the runtime of *k-NN* when applied to T_2

b) *k-NN* Classification using a Low-dimensional Representation of the Data

Implement a PCA-based approach to determine a low-dimensional representation of the data and then repeat all the steps of part 1. Make sure to use the same set of values for cross-validation and to compare the runtime/accuracy results with those obtained previously.