Data Mining 2

Assignment

CBA BATCH 12

Submission Instructions

- This is an **individual** assignment with **honor code 2N-c**. You *cannot* discuss the assignment generally or specifically with anyone else. References to DMG course material and external references is acceptable but must be cited appropriately.
- The date of submission is **21st March 11:55 PM.** All submissions must be on LMS.
- This assignment carries **50%** of the overall module weight.
- You will submit **2 files:** a) .IPYNB Jupyter notebook and b) word/pdf document.
- There are a total of 8 questions and 4 datasets for the entire DMG 2 assignment. For the first 6 questions, you could resubmit the files or do it again freshly in the same .IPYNB file with all the 8 questions.
- The code and subsequent analysis will be done entirely on a *single* **jupyter notebook in Python 3.**
- You *may* use whichever library you deem appropriate.
- You **must** set.seed equal to your PGID before you begin answering the questions. This will ensure each person will have their own unique train test-split.
- You **must** ensure the code output is visible and the analysis is done in the markdown cells of jupyter notebook. Ensure the question number is clearly visible
- Apart from the final answers, you will also be submitting a **word/pdf** document containing the code, figures, and analysis for all the problems.
- zip folders or any other file format will *not* be considered for evaluation.

Datasets

- **ISIR dataset**
  - 4 numeric features,
  - 3 classes,
  - 50 examples per class..
- **MUSHROOM dataset**
  - 20+ categorical features,
  - 2 classes
- **MNIST dataset**
  - 10 classes, 28 x 28 features,
  - Already comes with training and test splits as well
- **NEWSGROUP20 dataset**
  - 20 classes, Bag-of-Words datasets

P1 : IRIS – HIERARCHICAL FISHER

- Two classes in IRIS are more "similar" to each other. Find which ones using scatter plots. Lets say class 1 and class 2.
- Lets create a "meta-class" combining class 1 and class 2 (or whichever are the two most similar classes). Lets call it class 4.
- Create the first Fisher projection by trying to discriminate class 3 (the different class) from class 4 (the meta-class).
  - Do this on **training** data only
- Create the second Fisher projection by trying to discriminate class 1 from class 2 (the original two similar classes).
  - Do this on **training** data only
- Now project the entire data in these two projections and color code the class points.
  - Do this on **test** data only.
- Comment on what you observed and did.

10 points

P2 : MUSHROOM information gain

- Take the MUSHROOM **training** data. There are 20+ features and 2 classes. We want to find the BEST feature using the three purity measures: Accuracy, Gini Index, and Entropy.
- For each feature, partition the data into k regions where k is the number of values the feature can take.
- Measure the Information gain due to each feature. Generate a table with the following columns:
  - Feature_name
  - Accuracy
  - GINI index
  - 1- Entropy (NOTE: Use log_k for a feature with k values)
- Plot accuracy vs. 1 – Entropy scatter plot where each point is a feature.

10 points

P3 : MUSHROOM NB/DT

- Build Naïve Bayes and Decision Tree classifiers on the MUSHROOM training dataset.
  - In **Naïve Bayes** classifier plot the value of lambda (x-axis) for Laplacian smoothing against training and test set accuracy.
  - Lambda =0, 1, 2, …, 50
- For decision tree classifier plot the SizeThreshold (x-axis) against training and test set accuracy.
  - SizeThreshold =4, 8, 12, 16, 20, …, 64.
- Find the best values of lambda and SizeThreshold where the test set accuracies starts to decrease.
- Compare those accuracies across the two classifiers.

20 points

P4 : MNIST Bayesian

- Take the MNIST dataset. Lets call it D0 dataset
- Do a **9 dimensional PCA projection**. Lets call it D1 dataset
- Do a **9 dimensional FISHER projection**. Lets call it D2 dataset
- Build a Bayesian classifier on D1 (single Gaussian per class)
  - Diagonal Covariance matrix (i.e.set non-diagonals to zero)
  - Full Covariance matrix
- Build a Bayesian classifier on D2 (single Gaussian per class)
  - Diagonal Covariance
  - Full covariance
- Compare the test accuracies of the four classifiers and comment.

20 points

P5 : MNIST – kNN / Parzen window

- Take the two datasets D1 and D2 from P4.
- Build k-Nearest neighbors classifier with:
  - K = 1, 3, 5, 7, 9, 11, 13, 15, 17
  - Plot training and test accuracy with these values of k on x axis
- Build Parzen window classifier with:
  - Sigma =0.1, 0.2, 0.3, …, 3.0
  - Plot training and test accuracies with these values of sigma.
- Do both on D1 and D2 datasets.
- Comment on the optimal k and optimal sigma and compare those classifiers across D1 and D2 and see which one has highest test accuracy.

20 points

P6 : News group Text Classifier

- Build a Naïve Bayes Classifier on Newsgroup dataset
- DICTIONARY:
  - Compute the document frequency of all words (how many documents each word occurred in)
  - Sort this in descending order of document frequency
  - Pick the top 5000 and 10000 words as the dictionary.
- Learn P(w|c) for all words and classes
- Apply Laplacian smoothing of 30
- Compute the training and test set accuracy of the model.

20 points

P7 : Pair-wise Classifier Features

- For all pairs of classes in MNIST data
- Compute the Fisher Discriminant for that pair of classes
- This is again a D dimensional vector where each dimension corresponds to a pixel.
- Convert the D-dimensional vector into an image – scale it and draw the image
- Comment on how Fisher discriminant for a few pairs of classes is learning to "focus" on different pixels in the image.

10 points

P8 : Binary Hierarchical Classifier

- Build a Bottom-up-Binary Hierarchical Classifier
  (Use MNIST dataset)
- Use Linear SVM to build each of the pair-wise classifiers
- Merge two classes that give the highest training error
- Draw the entire tree that was discovered automatically
- For each node in the tree – write the training and test accuracies.

20 points