Lab Course Machine Learning
Exercise Sheet 7

Prof. Dr. Dr. Lars Schmidt-Thieme, Shayan Jawed
Information Systems and Machine Learning Lab
University of Hildesheim

December 10th, 2020
Submission on December 16th, 2020 at 12 noon, (on learnweb, course code 3116)

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a jupyter notebook detailing your solution.
2. Please set the seed(s) to 3116.
3. Please explain your approach i.e. how you solved a given problem and present your results in form of graphs and tables.
4. Please submit your jupyter notebook to learnweb before the deadline. Please refrain from emailing the solutions except in case of emergencies.
5. Unless explicitly noted, you are not allowed to use scikit, sklearn or any other library for solving any part.
6. Please refrain from plagiarism.

Exercise 0: Dataset Preprocessing (2 Points)

Time Series Classification Datasets
For this exercise sheet we shall be working with the University of California, Riverside's Time Series Classification Dataset Repository.

1. There are multiple datasets in this repository, you need to proceed to the page: `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/` and download the archive (260MB).
2. Preprocess the datasets in the following way:
   - Merge train and test splits and then re-split into train/validation/test splits according to ratios: 70%/15%/15%. Please take care that you do stratified splits.
   - If there are any datasets that do not have equal length samples, then pad the samples with 0s on the left. (This comes in useful for later distance calculations)
   - Standardize the datasets, by removing the mean and scaling to unit-variance ($\frac{x-\mu}{\sigma}$)
   - Ignore multivariate datasets (if any)
3. Plot interesting statistics:
   - A plot indicating the total the length of samples (across all datasets)
   - Similar to above show number of classes and number of samples.

Exercise 1: Dataset Imputation with KNN (6 Points)

We have been in the past ignoring missing values from the datasets, but for this lab, we shall try to implement imputation with K-Nearest Neighbor algorithm. The idea is to replace the missing value in a column by an average of its K-Nearest Neighbors where K acts as a hyperparameter.

1. List the datasets having missing values.
2. For each dataset with missing values, and for each feature (timestep) of it that has missing values impute the value by calculating the mean of its nearest K neighbors. You need to tune the hyperparameter K via grid search. If in case there are multiple feature values missing, use the same K for all during such tuning.
3. Next, train a K-Nearest Neighbour classifier (pseudo-code given in slides) with majority voting and euclidean distance to maximize accuracy on the validation split by tuning K via grid search.
4. Report the final test accuracy for each dataset by using the optimal K found for imputation and the optimal K found for classification. Please jointly tune the two Ks.

Exercise 2: Time Series Classification with Various Distance Measures (6 Points)

For this exercise, we shall be looking at various distance measures provided with the *scipy* library. The idea is to declare a single most optimal distance measure across all time series datasets. The tasks are the following:

1. For each distance measure on the page: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html` and for each dataset in the repository, use the validation samples to tune the parameter K and selecting one distance measure. With this optimal distance and value of K, compute the test accuracy.
2. Aggregate the results across all datasets, and rank all distance metrics according to the test accuracy.
3. Note: If you are not able to work through all datasets, please downscale the number of datasets.

Exercise 3: Accelerating K-Nearest Neighbour Classifier (6 Points)

The task for this exercise is to implement the following two speedup strategies covered in the lecture. Please refer to the lecture for pseudo-code and more details.

1. Partial Distances/Lower Bounding
2. Locality Sensitive Hashing (Bonus (6pts))

Please use the Euclidean distance and the dataset with the largest number of samples for this exercise. Justify the choice of associated hyperparameters of the above two strategies accordingly.