C S 487/519 Applied Machine Learning I

Dimensionality reduction techniques

1 Objective

In this *individual* project homework, you are required to understand and compare several dimensionality reduction techniques.

2 Requirements

- (36 points) Write code to conduct dimensionality reduction by
  - (12 points) using Principal Component Analysis (PCA) approach offered by scikit-learn library,
  - (12 points) using the Linear Discriminant Analysis method offered by scikit-learn library, and
  - (12 points) using a kernel PCA method offered by scikit-learn library.
- (22 points) Compare the performance of the different dimensionality reduction techniques. Please verify the performance of these techniques by feeding the dimension reduced data to a decision tree classifier. You may want to change different parameters (e.g., the number of components, kernels, etc.)
- (22 points) The algorithms need to be tested using two datasets: (1) the `iris` dataset which can be loaded from `sklearn.datasets`, and (2) the MNIST dataset, which can be loaded from `sklearn.datasets` using the function `fetch_mldata` (scikit-learn version before 0.19) or `fetch_openml` function (scikit-learn version from 0.20).
- (15 points) Properly analyze the different algorithms' behavior by applying the knowledge that we discussed in class. Such analysis should include classification accuracy (or F1, precision, recall, ROC, AUC) and running time. You can also use other metrics that look reasonable to conduct the analysis. Put the analysis to **report.pdf** file.
- (5 points) Write a readme file **readme.txt** with the commands to run your code. Your code needs to run in command line, accepting as input parameters the algorithm name, the dataset filename, and any required parameter. For example, "`python main.py pca dataset.csv -n_components 2`"
- Please properly organize your Python code. Each required task had better be implemented in a separate python file and imported into the main script. For example, to use Linear Discriminant Analysis method, you can create the script **mylda.py**, then use "`import mylda`" in the **main.py** file to test your implementation.
- Your Python code should be written for Python version 3.5.2 or higher.

3 Submission instructions

Put all your files (Python code, readme file, report, etc.) to a zip file named **hw.zip** and upload it to Canvas.

4 Grading criteria

(1) The score allocation has already been put beside the questions.

(2) Please make sure that you test your code **thoroughly** by considering all possible test cases. For this project, your code will **NOT** be tested using more datasets. Thus, it does not need to be flexible to accept different datasets as input. However, you still need to be flexible to identify parameters and make sure not to hardcode any parameter values.

(3) At least 5 points will be deducted if submitted files (including files types, file names, etc.) do not follow the instructions.