University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2020
Assignment 2

Due: 11:59pm March 1, 2020

Overview

Collaboration: Do your work and report individually. You can collaborate on the right tools to use and setting up your programming environment.

Hand in: One report per person, via the CROWDMARK in PDF format. You will need to divide the PDF up into multiple 5 files and drag and drop each onto the relevant question. You should receive an invite to crowdmark by email. Don't worry if the division is not perfect, as long as each question's answer is fully contained within that questions file. Also submit the code/scripts needed to reproduce your work as a python jupyter notebook to the LEARN dropbox.

Specific objectives:

- Loading a simple dataset and perform some basic data preprocessing.
- Studying how to apply some of the methods discussed in class on different datasets. The emphasis is on analysis and presentation of results not on code implemented or used (except for the implementation question).

Tools: You can use libraries available in python. You need to mention explicitly which libraries you are using, any blogs or papers you used to figure out how to carry out your calculations.
Notes: Use random state 42 in all functions.

1 Question 1: Effect of Normalization, Feature Extraction and Distance Metrics

For this question you will use the Wine Quality Data Set:

- Wine Quality Dataset: `https://archive.ics.uci.edu/ml/datasets/wine+quality`
- Original Data Directory: `http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/`
- Sample Python Jupyter notebook with data loaded : `asg2.ipynb` on LEARN. *Note* that this notebook loads the data and adds an additional column for wind `color` where `color` = 0 for white wine and `color` = 1 for red wine. This notebook can be copied and used as a starting point.

1.1 Tasks

In the class we talked about how to normalize data and introduced basic distance metrics. We will explore the effect of a variety of these methods on the wine quality simple dataset for a distance based classifier, `k nearest neighbours (knn)`. For this question you can use `scikitlearn` libraries for all the tasks.

- **Train/Test Data Split**: as for assignment 1, set `random seed=42` and perform all training on a an `80/20 Train/Test` split. No validation set is needed this time around.
- **Normalization:** Normalize the data using two normalization levels `none` and `z-score` normalization. **Show two pair plots**, one with the normalized features and one with unnormalized features. to compare and comment on the differences before and after normalization.
- **Classification:** Perform knn classification across a range of neighbourhood sizes $k \in [1, 50]$ using a variety of data point weighting schemes, including at least:
  - `uniform` weighting (default)
  - distance based weight with `manhattan distance`
  - distance based weight with `euclidean distance`
  - *[**optional bonus:**5%] :* You can try additional schemes as well to see if they work better, bonus if you can find a scheme and setting that works better than `manhattan+znorm` across a wide (¿15) range of k values.
- *[**optional bonus:**5%]* **Feature Selection:** try selecting your own subset of the data features (columns) to improve the performance. Bonus if you can find a set of 4 that does better than all data on same metrics.
- **Feature Extraction:** fit PCA and LDA models to the training dataset, then tranform the test data and run the same knn weighting schemes again, compare performance differences.
- **Analysis and Discussion**
  - **k Plots:** For each combination of weighting scheme, label (wine colour, wine quality), and feature subset (all features, your subset, first 5 principle components, LDA components) create a plot of the `classification accuracy vs. neighbourhood size` as in the Jupyter notebook example. You should be able to code all this up in
  - **Features:** Some discussion on the relationship between the features from any analysis you performed.

- **Selected Features:** Were you able to find a subset of features that worked better than all features? How about compared to PCA or LDA?
- **PCA vs. LDA:** Did either of these methods help in this situation? Which worked better for the task? Did normalization impact the performance of either of them?
- **Plot** project he data on the first two components for PCA and LDA and colour the points by the two labels ('quality','color') so four plots in total. How does this compare or inform your understanding of the data from the pairplots or other results?

2 Question 2: Linear Dimensionality Reduction

2.1 Dataset

dataset B: Handwritten digits of 0, 1, 2, 3, and 4 (5 classes). This dataset contains 2066 samples with 784 features corresponding to a 28 x 28 gray-scale (0-255) image of the digit, arranged column-wise. This data is used to illustrate the difference between feature extraction methods. (File: DataB.csv)

2.2 Principal Component Analysis (PCA)

2.2.1 Practical Questions

1. In PCA, compute the eigenvectors and eigenvalues. Plot the scree plot and visually discuss which cut-off is good.
2. Using subplot in python matplotlib, plot the scatter plot of the projected data with the top 20 eigenvalues (although PCA does not use labels but use colors and legend to show the class instances). Is there a clear point where you could cut off the dimensions? Compare your analysis with the analysis from previous section.
3. Plot two 2-dimensional representations of the data points based on the first vs second principal components and 5th vs 6th displaying the data points of each class with a different color (you will need to project the data). Explain the results versus the known classes and compare between the two plots.
4. *Implement* (1) PCA and (2) dual PCA with singular value decomposition. [Note: No libraries for PCA, dual PCA or SVD should be used for this question. In PCA and dual PCA, only left and right matrices of singular vector are used. So, use the relationship of SVD and eigenvalue decomposition to find the left and right singular vectors and the singular values. And then, use them in PCA and dual PCA.] Save the time of computations and compare the times. Analyze your comparison.

2.2.2 Theoretical Question

Prove that PCA is the best linear method for reconstruction (with orthonormal bases). Hint: write down the optimization problem and solve it.

2.3 Fisher Discriminant Analysis (FDA)

As discussed in the class, Fisher Discriminant Analysis (FDA) and Linear Discriminant Analysis (LDA) are equivalent.

2.3.1 Practical Questions

1. As the class labels are already known, you can use the FDA or LDA to reduce the dimensionality. Using any implementation of FDA or LDA you wish, and subplot in python matplotlib, plot the scatter plot of the projected data with the top 20 eigenvalues (use colors and legends for classes). Explain the results obtained in terms of the known classes. Which LDA directions separate which classes better (which LDA directions are responsible for separating which classes)?

2. Compare the results of the LDA with the results obtained by using PCA.

2.3.2 Theoretical Question

We can consider the total scatter as the summation of the within and between scatters: $\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B \implies \boldsymbol{S}_B = \boldsymbol{S}_T - \boldsymbol{S}_W$. By substituting this into the Fisher criterion, the FDA optimization can be slightly modified to:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \mathbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_T \boldsymbol{U}) \\ \text{subject to} \quad & \boldsymbol{U}^\top \boldsymbol{S}_W \boldsymbol{U} = \boldsymbol{I}. \end{aligned}$$

Compare this with the optimization in PCA. Explain and analyze your comparison. After this analysis, compare your theoretical comparison (this question) and the practical comparison (question 2 in practical questions).

3 Question 3: Nonlinear Dimensionality Reduction

3.1 Dataset

Use dataset B again.

3.2 Practical Questions

1. Apply the following methods on the dataset (use the default sklearn parameters except for n_components and kernel. Use n_components=2 for all because of 2D visualization. For kernel PCA, use RBF kernel.):

- kernel PCA
- Isomap
- Locally Linear Embedding (LLE)
- Laplacian Eigenmap (sklearn.manifold.SpectralEmbedding)
- t-SNE

Save the embeddings for the next part.

2. Plot the scatter plot of the embeddings. Compare the embeddings of the different above methods. Completely analyze your comparisons including questions suchg as : Which methods do better on which parts of the data? Give at least three clear performance differences between a pair of methods that you can explain based on the nature of methods and the data. What tradeoffs might need to be considered in order to decide which method is 'best' to use for this dataset?