Ira A. Fulton Schools of Engineering
Arizona State University

CSE 572: Data Mining
Fall 2022 (Kerner)

Homework 2: Dimensionality Reduction

In this homework, you will compare the effect of multiple dimensionality reduction techniques on the classification performance for the Covertype dataset. The classification task for this dataset is to predict the forest cover type of a 30 m x 30 m patch of forested land described by 54 attributes. The attributes include features such as elevation, aspect, slope, soil characteristics, etc. The dataset was created by the Department of Forest Sciences at Colorado State University and the US Forest Service in 1998.

You will implement 2 dimensionality reduction techniques:

- PCA (linear)
- Autoencoder neural network (non-linear)

You will compare the classification results using these two dimensionality reduction techniques to a classifier that uses no dimensionality reduction.

You may use any libraries that we have used in labs, including Scikit-learn, numpy, pandas, and keras/tensorflow.

Data Preparation

You will split your dataset into a training (70% of the total data) and test (30%) set. You do not need to create a validation set because you will not be doing any hyperparameter tuning (the hyperparameter settings are specified in the next section). You will need to scale your data to have 0 mean (e.g., using standardization) which is required by PCA. Make sure to split your dataset *before* applying dimensionality reduction techniques.

Dimensionality reduction

You will implement 2 dimensionality reduction techniques:

- PCA (linear)
- Autoencoder neural network (non-linear)

For PCA, you will create a plot of the total fraction of explained variance by the first 1 through 10 principal components (as we did in Lab 11). Choose the number of principal components to retain based on the inflection point of this plot, i.e., the point at which the increase in total explained variance begins to plateau (as we did in Lab 11).

For the autoencoder neural network, implement a network with the following layers:

1. Input layer (# units = 54) [encoder]
2. Hidden layer (# units = 32) [encoder]
3. Hidden layer (# units = number of PCs retained for PCA) [encoded/bottleneck layer]
4. Hidden layer (# units = 32) [decoder]
5. Output layer (# units = 54) [decoder]

For example, if you chose to use 3 principal components in PCA, you will have a bottleneck layer of 3 units in your autoencoder.

Use ‘relu’ activation for hidden layers and ‘sigmoid’ activation for the output layer, ‘sgd’ (stochastic gradient descent) for the optimizer, and ‘mse’ (mean squared error) as the loss function. Train your model for 100 epochs with a batch size of 64. Lab 12 will be a useful guide for this implementation. Note that you will use the predict() function with only the encoder part of the model to transform your features into the encoded (reduced-dimension) representation.

Classification

You will use a Random Forest classifier with 100 trees for the classification model (using Scikit-learn). Leave all other hyperparameters as their default values. You will train 3 separate random forest classifiers with 1) input data transformed using PCA, 2) input data transformed using autoencoder, 3) no dimensionality reduction (original data attributes).

Evaluation

Your final model evaluation should be performed on the test set. You will compare the results of the two dimensionality reduction + Random Forest methods (PCA + RF, Autoencoder + RF) as well as a baseline Random Forest classifier that does not use any dimensionality reduction (the original attributes will be the input feature vector). For each of the 3 methods, print the classification report (including class-wise precision, recall, F1 + overall accuracy) and plot the confusion matrix.

Discussion

Briefly summarize the results from your three compared models and how the different dimensionality reduction techniques affected the random forest classifier performance.

Other considerations

Be sure to set your random seed at the beginning of your code and take any other steps to maximize reproducibility of your model results. Use a random seed = 0 whenever a seed is required; this will maximize consistency across solutions by many students.

Submission

You will add your code to the notebook provided in the assignment instructions which contains starter code for loading the dataset (`cse572-homework2.ipynb`). Rename the notebook to `cse572-homework2-<lastname>.ipynb` and submit the following three deliverables:

1. a link to your Colab notebook (as a comment on the submission)
2. your .ipynb file (`cse572-homework2-<lastname>.ipynb`)
3. a pdf of the executed notebook (`cse572-homework2-<lastname>.pdf`)

Grading

Grading will be based on your code and the accuracy of your results.

Points will be distributed as follows:

- 10 points for Data Preparation
- 30 points for dimensionality reduction (15 for PCA, 15 for autoencoder)
- 30 points for random forest classifier training (10 points for each of the 3 models)
- 24 points for Evaluation (8 points for each of the 3 models)
- 6 points for Discussion