Collin Gros
03-27-2021
CS-487
Homework 5

**PROBLEM**

Conduct dimensionality reduction using PCA, LDA, and KPCA on the IRIS dataset and the MNIST dataset. Compare the performance of the different techniques. Analyze their behavior and report on classification accuracy and running time.

**METHODOLOGY**

For each reduction technique:

- 70% of the IRIS dataset was used for training, and 30% was used for testing.
- 5000 samples of the MNIST dataset were used for training, and 10000 were used for testing (due to the massive size of the dataset, I had to use a small portion to use the .transform() methods of the reduction techniques).
- 70% of the DIGITS dataset was used for training, and 30% was used for testing.

All three reduction techniques were implemented using SCIKIT-LEARN's classes. The program code was developed modularly, with each class having its own Python file.

Testing was done via a BaSH script, calling the program with default parameters as follows:

```
# DT
args.criterion = 'gini'
args.max_depth = 4
args.random_state = 1
# PCA/LDA/KPCA
args.n_components = 2
```

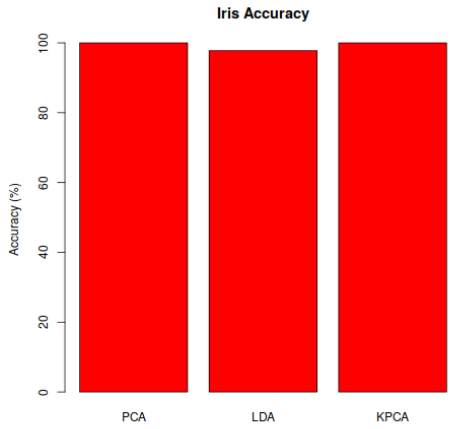
'args.n_components' was varied between 2 and 1 for testing.

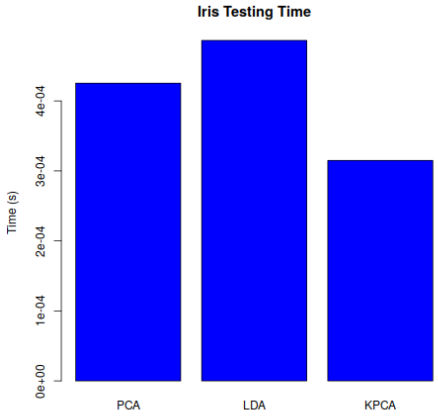
The DIGITS dataset was not part of the requirements, but I wanted to gather more information for my analysis.

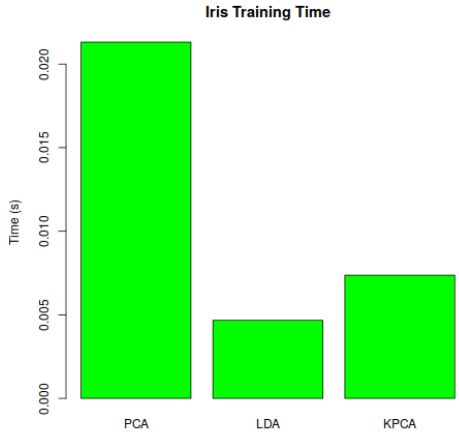
**RESULTS**

**N_COMPONENTS = 1**

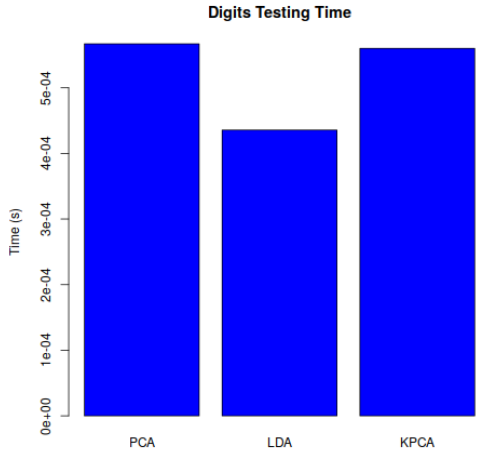
**IRIS**

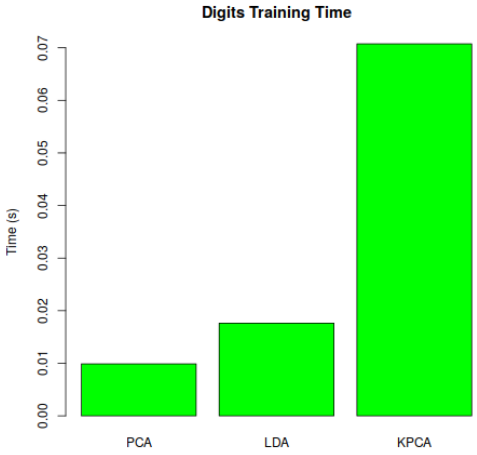


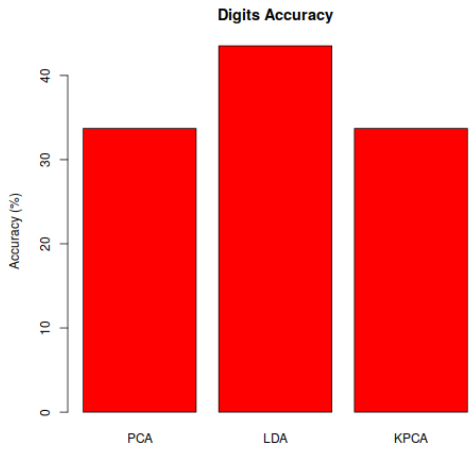



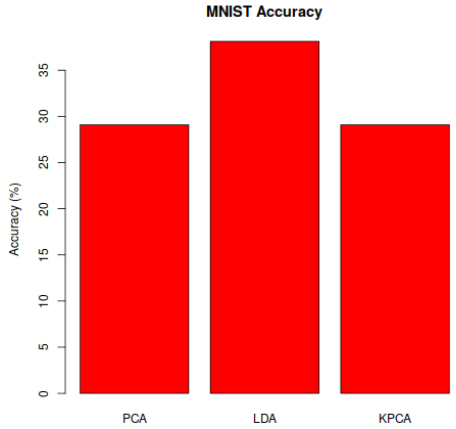
## DIGITS

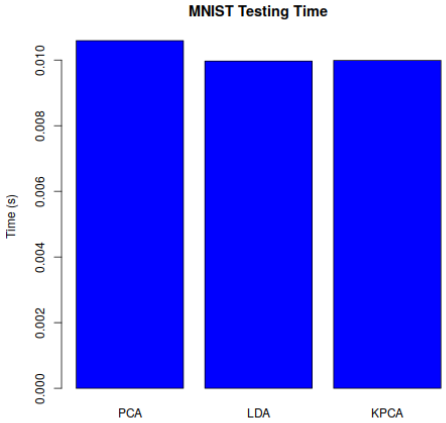

Digits Testing Time
Time (s)
0e+00
1e-04
2e-04
3e-04
4e-04
5e-04
PCA
LDA
KPCA

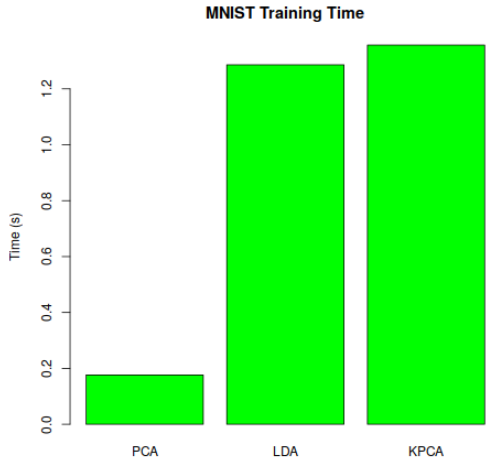

Digits Training Time
Time (s)
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
PCA
LDA
KPCA


Digits Accuracy
Accuracy (%)
0
10
20
30
40
PCA
LDA
KPCA

## MNIST

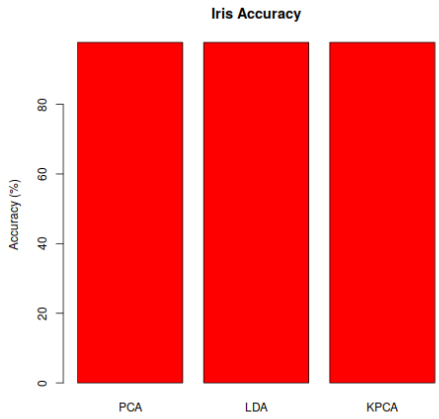

MNIST Accuracy
Accuracy (%)
0
5
10
15
20
25
30
35
PCA
LDA
KPCA


MNIST Testing Time
Time (s)
0.000
0.002
0.004
0.006
0.008
0.010
PCA
LDA
KPCA


MNIST Training Time
Time (s)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
PCA
LDA
KPCA

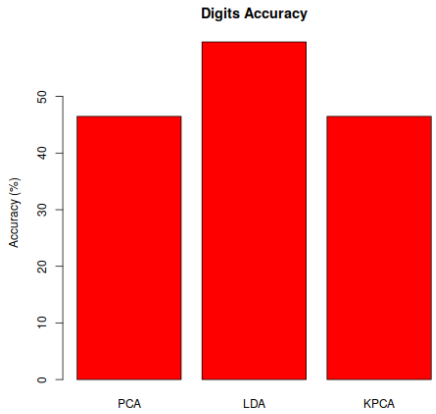
**N_COMPONENTS = 2**

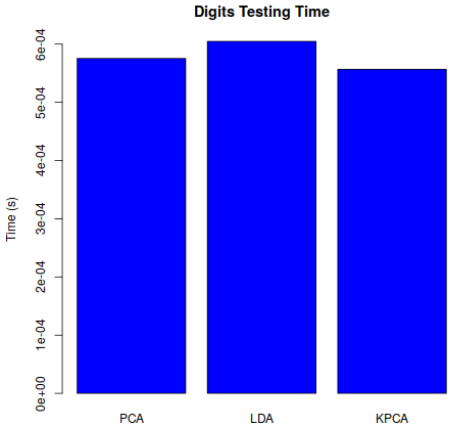
**IRIS**

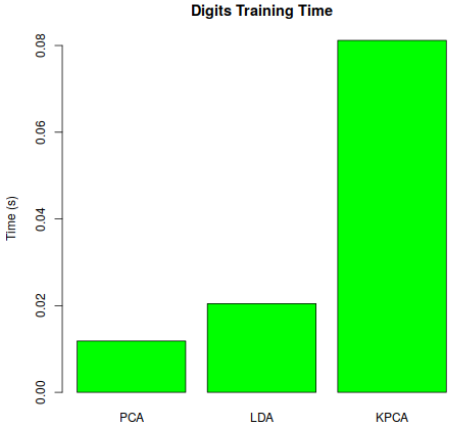




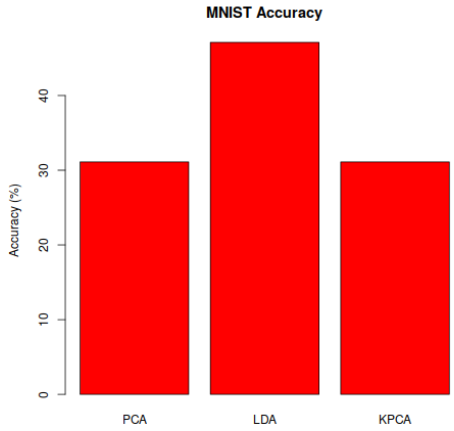
## DIGITS

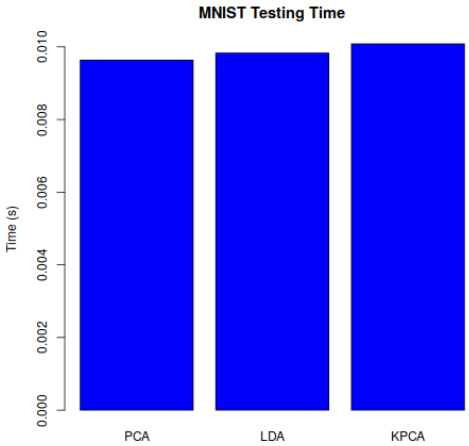

Digits Accuracy
Accuracy (%)
0
10
20
30
40
50
PCA
LDA
KPCA

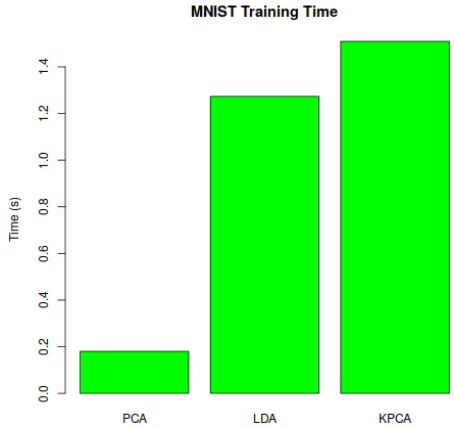

Digits Testing Time
Time (s)
0e+00
1e-04
2e-04
3e-04
4e-04
5e-04
6e-04
PCA
LDA
KPCA


Digits Training Time
Time (s)
0.00
0.02
0.04
0.06
0.08
PCA
LDA
KPCA

# MNIST


MNIST Accuracy
Accuracy (%)
0
10
20
30
40
PCA
LDA
KPCA


MNIST Testing Time
Time (s)
0.000
0.002
0.004
0.006
0.008
0.010
PCA
LDA
KPCA


MNIST Training Time
Time (s)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
PCA
LDA
KPCA

**ANALYSIS**

All three techniques reached nearly 100% accuracy on the IRIS dataset. I'm assuming it is because this dataset is more separable than the others.

In both of the DIGITS and MNIST cases, the reduction technique with the highest accuracy is LDA, while PCA and KPCA are about even. Therefore, in all cases, LDA outperformed PCA and KPCA.

All accuracies were decreased in the **N_COMPONENTS = 1** set, as compared to the **N_COMPONENTS = 2** set. I think this means that **N_COMPONENTS** should be maximized to provide the best accuracy results.

Testing times varied across the board, and no pattern could really be found from analyzing them. I think this is because of the different processes occurring in the background while the program was running, and has no major impact on the testing time of the reduction technique.

Training times varied, however, KPCA was the slowest in ⅔ of the cases (with IRIS being the exception), and PCA was fastest in ⅔ of the cases (with IRIS also being the exception).

In my tests, I found LDA to be the best performing reduction technique. I would use LDA in future classification problems, if I had to choose between LDA, PCA, and KPCA.