ASSIGNMENT-5

“Classification Accuracy on MNIST handwritten digits data by k-NNC algorithm after doing k-means clustering on the training data"

Submitted By

Sourabh Kumar

Stud ID: 15/8

Course Name- CPI 015 Pattern Recognition- Spring 2020

Under the guidance of:

Prof. MN Murthy

Department of Computer Science and Automation

Indian Institute of Science, Bangalore, India

Problem Statement

1. Download MNIST handwritten digit data. There are 10 classes (corresponding to digits 0, 1, ..., 9) and each digit is viewed as an image of size 28 × 28 (= 784) pixels; each pixel having values 0 to 255. There are around 6000-digit training patterns and around 1000 test patterns in each class and the class label is also provided for each of the digits. Visit http://yann.lecun.com/exdb/mnist/ for more details.

2. Run K-means algorithm as follows: (a) Consider classes 0 (digit zero) and 1 (digit one). Cluster patterns in each class separately into K clusters using the K-Means algorithm. Use these 2K centroids (K from 0 and K from 1) as training data to classify the test patterns and compute accuracy using the NNC. Vary the value of K from 100 to 500 in steps of 100. (b) Repeat the experiment in step 1 with the pair of classes 7 and 9. (c) Repeat the steps (a) and (b) by clustering the entire data set consisting of both classes 0 and 1. (d) Use the K-Means++ algorithm and repeat steps (a), (b), (c) above.

3. Report your results appropriately using tables and graphs for different scenarios.

4. The report must be brief giving a page on the resources used and how they are used. Two-three pages on the results of your experiments.

Technology and Programming Resources Used

- Spyder Programming Editor
- Python Programming Language 3.7
- Following popular sklearn python libraries for machine learning
    a. sklearn.datasets for fetching MNIST data (fetches data internally from the source web site- http://yann.lecun.com/exdb/mnist/)
    b. Sklearn.cluster import KMeans for K-Means clustering
    c. sklearn.preprocessing for binarizing the data based on below logic
        1. range [0,127] – Binary value 0
        2. range [128,255] – Binary value 1
    d. from sklearn.neighbors import KNeighborsClassifier
    e. matplotlib.pyplot library for plotting charts
- MNIST hand written digit data with
    a. Total Features -784 (pixel grid size- 28x28)
    b. Total Classes- 10 (Digit 0 to Digit 9)
    c. Total Training data- 60000 (6000 per class)
    d. Total Test data- 10000 (1000 per class)
    e. Two pairs of Class data used for experiments- “Class 0 & Class 1” and “Class 7 & Class 9” together

Dataset Pre-Processing: -

Data set – Class 0 & Class 1 data

Training data - First 12,000 records (starting from 0 to 11,999 row indices)
and all 784 feature fields

Training target- First 12,000 records (starting from 0 to 11,999 row indices)
and last785th target field.

Test data – First 2,000 records (starting from 60,000 to 61,999 row
indices) and all 784 feature fields

Test target - First 2,000 records (starting from 60,000 to 61,999 row
indices) and last 785th target field

Data set – Class 7 & Class 9 data

Training data - 6,000 records (starting from 42,000 to 47,999 row indices)
and

6,000 records (starting from 54,000 to 59,999 row indices)
for all 784 feature fields

Training target- 6,000 records (starting from 42,000 to 47,999 row indices)
and 6,000 records (starting from 54,000 to 59,999 row indices)
for last785th target field.

Test data – 1,000 records (starting from 67,000 to 67,999 row indices)
and

1,000 records (starting from 69,000 to 69,999 row indices)
and all 784 feature fields

Test target - 1,000 records (starting from 67,000 to 67,999 row indices)
and 1,000 records (starting from 69,000 to 69,999 row indices)
for last 785th target field

<u>Experiment-1</u>

Check Classification Accuracy for first pair (Class 0 & Class 1) -clustering first and combining later

- Data sets– Class 0 & Class 1 separately
- Program Name- CLASS_0_1_Kmeans.py
- K-means clustering algorithm on Class 0 and Class 1 separately and later combining their clusters in 2K manner for redefining test patterns.
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.
- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm on clustered data set as training data (2K patterns and 2000 Test patterns)
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 0 & 1)
- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 0 & 1)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

Dataset	K value (K-means algorithm)	Accuracy %
Class 0 & 1	200	64.4
Class 0 & 1	400	66.55
Class 0 & 1	600	68.5
Class 0 & 1	800	67.95
Class 0 & 1	1000	66.75

<u>Experiment-2</u>

Check Classification Accuracy for second pair (Class 7 & Class 9) -clustering first and combining later

- Data sets – Class 7 & Class 9 separately
- Program Name- CLASS_7_9_Kmeans.py
- K-means clustering algorithm on Class 7 and Class 9 separately and later combining their clusters in 2K manner for redefining test patterns.
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.

- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm- Fitting clustered data set as new training data and new training labels (2K patterns) and predicting remaining 2000 Test patterns using this model by comparing predicted label and actual labels.
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 7 & 9)
- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 7 & 9)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

Dataset	K value (K-means algorithm)	Accuracy %
Class 7 & 9	200	73.15
Class 7 & 9	400	73.7
Class 7 & 9	600	66.1
Class 7 & 9	800	70.45
Class 7 & 9	1000	75.4

Experiment-3

Check Classification Accuracy for first pair (Class 0 & Class 1) -Combining first and clustering later

- Data set – (Class 0 & Class 1) together
- Program Name- CLASS_0_1_Kmeans.py
- First combining Class 0 and Class 1 together and then K-means clustering algorithm on this data
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.
- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm- Fitting clustered data set as new training data and new training labels (2K patterns) and predicting remaining 2000 Test patterns using this model by comparing predicted label and actual labels.
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 0 & 1)

- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 0 & 1)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

Dataset	K value (K-means algorithm)	Accuracy %
Class 0 & 1	200	84.9
Class 0 & 1	400	88.3
Class 0 & 1	600	89.45
Class 0 & 1	800	89.75
Class 0 & 1	1000	91.45

Experiment-4

Check Classification Accuracy for first pair (Class 7 & Class 9) -Combining first and clustering later

- Data set – (Class 7 & Class 9) together
- Program Name- CLASS_7_9_Kmeans.py
- First combining Class 7 and Class 9 together and then K-means clustering algorithm on this data
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.
- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm- Fitting clustered data set as new training data and new training labels (2K patterns) and predicting remaining 2000 Test patterns using this model by comparing predicted label and actual labels.
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 7 & 9)
- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 7 & 9)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

Dataset	K value (K-means algorithm)	Accuracy %
Class 7 & 9	200	91.7
Class 7 & 9	400	92.6
Class 7 & 9	600	94.9
Class 7 & 9	800	94.15
Class 7 & 9	1000	94.25

Experiment-5

Check Classification Accuracy for first pair (Class 0 & Class 1) -clustering first and combining later

- Data sets– Class 0 & Class 1 separately
- Program Name- CLASS_0_1_Kmeans++.py
- K-means clustering algorithm on Class 0 and Class 1 separately and later combining their clusters in 2K manner for redefining test patterns.
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.
- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm on clustered data set as training data (2K patterns and 2000 Test patterns)
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 0 & 1)
- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 0 & 1)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

Dataset	K value (K-means++ algorithm)	Accuracy %
Class 0 & 1	200	63.45
Class 0 & 1	400	65.75
Class 0 & 1	600	69.65
Class 0 & 1	800	70.55
Class 0 & 1	1000	68.85

Experiment-6

Check Classification Accuracy for second pair (Class 7 & Class 9) -clustering first and combining later

- Data sets – Class 7 & Class 9 separately
- Program Name- CLASS_7_9_Kmeans++.py
- K-means clustering algorithm on Class 7 and Class 9 separately and later combining their clusters in 2K manner for redefining test patterns.
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.
- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm- Fitting clustered data set as new training data and new training labels (2K patterns) and predicting remaining 2000 Test patterns using this model by comparing predicted label and actual labels.
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 7 & 9)
- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 7 & 9)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

Dataset	K value (K-means++ algorithm)	Accuracy %
Class 7 & 9	200	75.4
Class 7 & 9	400	72.5
Class 7 & 9	600	71.05
Class 7 & 9	800	74.8
Class 7 & 9	1000	73.6

Experiment-7

Check Classification Accuracy for first pair (Class 0 & Class 1) -Combining first and clustering later

- Data set – (Class 0 & Class 1) together
- Program Name- CLASS_0_1_Kmeans++.py
- First combining Class 0 and Class 1 together and then K-means++ clustering algorithm on this data
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.

- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm- Fitting clustered data set as new training data and new training labels (2K patterns) and predicting remaining 2000 Test patterns using this model by comparing predicted label and actual labels.
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 0 & 1)
- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 0 & 1)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

Dataset	K value (K-means++ algorithm)	Accuracy %
Class 0 & 1	200	85.75
Class 0 & 1	400	89.25
Class 0 & 1	600	90.4
Class 0 & 1	800	90.95
Class 0 & 1	1000	90.95

Experiment-8

Check Classification Accuracy for first pair (Class 7 & Class 9) -Combining first and clustering later

- Data set – (Class 7 & Class 9) together
- Program Name- CLASS_7_9_Kmeans++.py
- First combining Class 7 and Class 9 together and then K-means++ clustering algorithm on this data
- This algorithm will return reduced number of **new training data patterns** which are cluster centroids.
- Later we need to derive the **new training labels** with the help of actual cluster labels and computing the most common label for training data within each cluster (this requires some look up and counting operation within a program).
- K'-NNC classification algorithm- Fitting clustered data set as new training data and new training labels (2K patterns) and predicting remaining 2000 Test patterns using this model by comparing predicted label and actual labels.
- K = [100,200,300,400,500] (Number of clusters formed on individual class data- 7 & 9)
- 2K = [200,400,600,600,1000] (Combined number of clusters on Class pair- 7 & 9)
- K' = 4 (number of neighbours used in K'-NNC classification algorithm)
- Distance metric in K'-NNC algorithm = L2

Result Table:

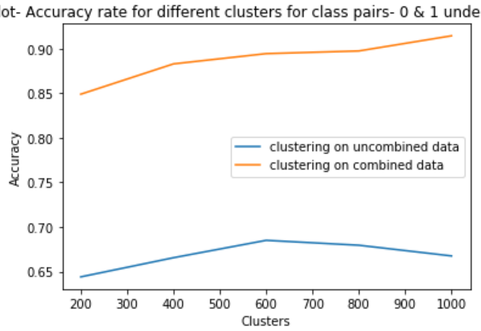
Dataset	K value (K-means++ algorithm)	Accuracy %
Class 7 & 9	200	91.9
Class 7 & 9	400	93.5
Class 7 & 9	600	93.85
Class 7 & 9	800	94.9
Class 7 & 9	1000	95.25

Final Conclusion (combined for all experiments based on the separate results tables and plots):

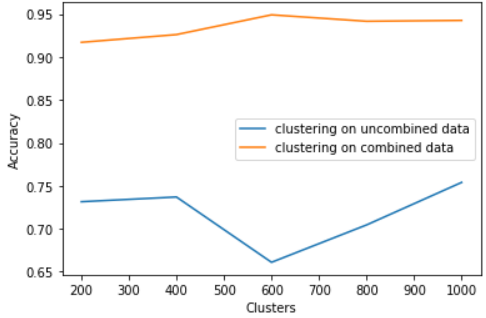
1. Experiments 1 to 4 have a similarity in terms of clustering algorithm- K-means. Similarly, experiments 5 to 8 are using k-means++ for clustering.
2. Experiment 1 to 4, there are couple of things observed-
   - If the data was clustered individually for classes 0, 1 and similarly classes 7 & 9 separately and then later combined, then the k-NNC classification accuracy was low. (). It appears this process introduces more outliers in the training set.
   - Whereas, if the data was combined first for both classes 0, 1 and similarly 7,9 and then clustered, then the k-NNC classification accuracy was high and also model was trained sooner.
3. Experiment 5 to 8 also showed similar trend as point 2 above.
4. K-means++ in general appears to be a better algorithm for clustering by seeing the accuracy rates.
5. Below plots will show the trend.

Plots:

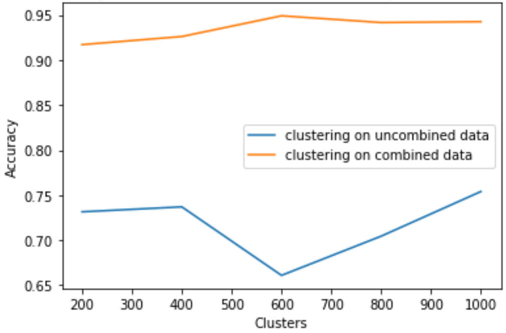
Line Plot- Accuracy rate for different clusters for class pairs- 0 & 1 under k-means++



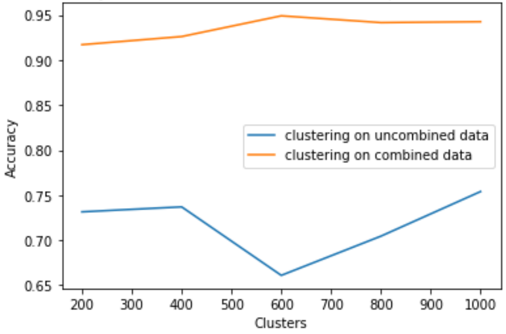
Line Plot- Accuracy rate for different clusters for class pairs- 7 & 9 under k-means++



Line Plot- Accuracy rate for different clusters for class pairs- 7 & 9 under k-means



Line Plot- Accuracy rate for different clusters for class pairs- 7 & 9 under k-means


Accuracy
clustering on uncombined data
clustering on combined data
Clusters