Project 2 Report

Data Cluster using K-means algorithm

Submitted By : Adithya Vadlamani (1001417682)

Description:

Using k-mean algorithm to split the data into k classes and calculating the accuracy using Hungarian algorithm,

1. We use the same number of classes in kmean as there is in original data.
2. Obtain the confusion matrix using the original labels and predicted labels.
3. Use hungarian algorithm to solve the confusion matrix and obtain the accuracy of k means.

Language Used	Python (2.7.15)
Library used	Numpy, sklearn
Files used	ATNTFaceImages400.txt, HandWrittenLetters.txt, project2.py

TASKS

Task A: Run k-means on AT&T 100 images, set K=10. Obtain confusion matrix. Re-order the confusion matrix and obtain accuracy.

Command	python project2.py A
Output	Confusion Matrix [[0 0 0 0 0 0 0 0 10 0] [0 0 10 0 0 0 0 0 0 0] [5 0 0 0 0 0 0 0 0 5] [4 0 0 0 0 0 0 0 0 6] [0 0 0 0 0 0 0 10 0 0] [0 0 0 10 0 0 0 0 0 0] [0 0 0 0 0 10 0 0 0 0] [0 0 0 0 10 0 0 0 0 0] [0 10 0 0 0 0 0 0 0 0] [1 0 0 0 0 0 9 0 0 0]] Reordered Matrix [[5 0 0 0 0 0 1 0 0 4] [0 10 0 0 0 0 0 0 0 0] [0 0 10 0 0 0 0 0 0 0] [0 0 0 10 0 0 0 0 0 0] [0 0 0 0 10 0 0 0 0 0] [0 0 0 0 0 10 0 0 0 0] [0 0 0 0 0 0 9 0 0 0] [0 0 0 0 0 0 0 10 0 0] [0 0 0 0 0 0 0 0 10 0] [5 0 0 0 0 0 0 0 0 6]] Accuracy = 90.00%

Task B: Run k-means on AT&T 400 images, set K=40. Obtain confusion matrix. Re-order the confusion matrix and obtain accuracy.

Command	python project2.py B
Output	Confusion Matrix [[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 9 0 0] ... [0 0 0 ... 0 0 0] [0 4 0 ... 0 0 0] [1 0 0 ... 0 0 4]] Reordered Matrix [[10 0 0 ... 0 0 0] [0 10 4 ... 0 0 0] [0 0 0 ... 0 0 0] ... [0 0 0 ... 9 0 0] [0 0 0 ... 0 9 0] [0 0 0 ... 0 0 6]] Accuracy = 73.25%

Task C: Run k-means on Hand-written-letters data, set K=26, as above.

Command	python project2.py C
Output	[0 0 16 1 11 0 0 3 0 0 0 0 0 0 2 0 3 0 0 0 0] [2 0 6 2 0 0 0 0 3 1 2 0 0 0 1 0 1 0 0 0 0] [0 1 0 13 2 2 1 1 0 0 5 0 0 0 0 0 0 3 0 0 0] [0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0] [0 0 0 4 0 0 0 0 0 0 0 1 0 3 0 7 0 0 0 17 10] [0 0 0 1 1 0 0 0 0 0 12 0 0 5 0 3 0 2 1 5 11]] Accuracy = 46.55%