Name :- Pratik Mahajan

E.E 511 :- Homework 5

Problem 1:-

I have read and understood the general instructions at the top of HW5 and I formally declare that all work I turn in for everything in this course will not contain or involve any cheating at all.

Problem 2:-

a) Note that all printed evaluation metrics are w/o matching

(i) Please refer to the end of this pdf for this qn.

(ii) Please refer to the end of this pdf for this qn.

b) For the MNIST dataset with 10 classes & 10

(i) clusters, there are $(10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1) = 10!$
$= 3628800$ matchings possible.
(The above can be deduced by matching clusters to a class 1 by 1.
First, match cluster 1 to a class (there are 10 possible classes) then cluster 2 which would have 9 possibles till cluster 10, which would have only 1 class remaining.)

There are 2 parts of the Hungarian algorithm that are responsible for its worst case complexity :- equality graph construction & augmented path search.

The equality graph construction assigns labels to rows & columns of the cost matrix, and has a worst case complexity of $O(n^2)$ per iteration. The same also applies to the ~~aug~~ path search in the E_G, using BFS/DFS.

∴ the net worst case complexity would be $O(n^2 \times n)$, where n is the number of tasks/workers in the Hungarian algo..

⇒ net worst case time complexity $= O(n^3)$.

(ii) Please refer to the end of this pdf for this qn.

c) (i) Please refer to the end of this pdf for this qn.

(ii) Yes, spectral clustering improves the accuracy of K-means from $\simeq 50\%$ → $\simeq 60\%$.

This is mostly because spectral clustering is good at representing complex relationships b/w data & transforming data from a high dimensional manifold to a low dimensions one with these relationships.
K-means on the other hand is biased towards spherical relationships and so does not handle datasets like MNIST as well.

d) (i) Please refer to the end of this pdf for this qn.

(ii) The random sampled X_s ~~dataset~~ performs the worst, which is expected as doing any form of clustering would give you better reference points for K-NN than random sampling.

However, spectral KNN show a strange pattern:—

→ K=1 → KNN performs better
→ K=3 → Similar perf.
→ K=5 → Spectral performs better

My ~~mod~~ intuition says this is because spectral takes a more nuanced approach to representing relationships amongst various points in the dataset, that make it better for higher ks, as it more thorough in its "clustering analysis". However, for lower ks (like k=1) this level of thorough analysis in the data generalises it too much, which is why K-NN performs better

Problem 3:—

a) Please ~~update this~~ refer to the end of this pdf for this qn.

b) Please refer to the end of this pdf for this qn.

c) Please refer to the end of this pdf for this qn.

d) Please refer to the end of this pdf for this qn.

e) (next page)

Validation Accuracy:-

Logistic Regression	Random Forest	K-NN	Multilayer Perceptron
0.90264	0.01565	0.9418	0.9664

Overall, it seems like ~~K-NN~~ MLP > ~~MLP~~ KNN > Logistic > RF.

Unfortunately, in RF's, I suspect it's an implementation issue causing a poor accuracy. For the others, I like to view MLP as a complex version of logistic regression in a way. Since the given dataset is also complex & likely has ~~not~~ a ~~not~~ complex decision boundary, MLP performs better than the logistic regressions. K-NN also has a tendency to simplify decision boundaries and so, MLP outperforms K-NN in this dataset. Further, while K-NN may not ~~give~~ generate as complex decision boundaries as MLP, logistic regression tends to stay closer to a linear form, while K-NN is still good at representing non-linear boundaries. ∴ KNN outperforms the logistic regression in this dataset, which clearly has complex non-linear decision boundaries.

f) As MLP performed best in my previous runs, I chose MLP as the classifier for this run. I used sklearn to implement it & tuned its hyperparameters in 3d) (which I reused in 3f) using GridSearchCV. Please find the .txt file in the submission & the code in the attached .mpynb files.