Kai Wong

UID: 704451679

CS 145 Fall 2018, Homework 4

1. Clustering Evaluation

Purity:

N = 20

Labels	Predicted IDs	Ground Truth IDs
1	10,12,14,16,18	3,4,5,13,17
2	1,2,7,8,15,17	10,12,14,16,18
3	3,4,5,9,13	1,2,7,8,15
4	6,11,19,20	6,9,11,19,20

Match predicted cluster labels to ground truth cluster labels:

1 → 2, 2 → 3, 3 → 1, 4 → 4

Let C_i be the predicted clustering results and W_i be the ground truth clustering results.

$| C_1 \cap W_2 | = 5$

$| C_2 \cap W_3 | = 5$

$| C_3 \cap W_1 | = 4$

$| C_4 \cap W_4 | = 4$

purity($\mathbf{C}$, $\mathbf{W}$) = $(1/20) * (5 + 5 + 4 + 4) = 0.9$

Precision, Recall & F-measure:

pairs of data points = $\binom{20}{2} = 190$

Using a python script, I computed the following:

TP = 32

FP = 9

TN = 141

FN = 8

Precision = TP/(TP + FP) = 32/41 = 0.780

Recall = TP/(TP + FN) = 32/40 = 0.8

F-measure = (2 * Precision * Recall) / (Precision + Recall) = 0.790

Normalized Mutual Information (NMI):

I used sklearn.metrics.normalized_mutual_info_score() to compute the NMI:

NMI = 0.815

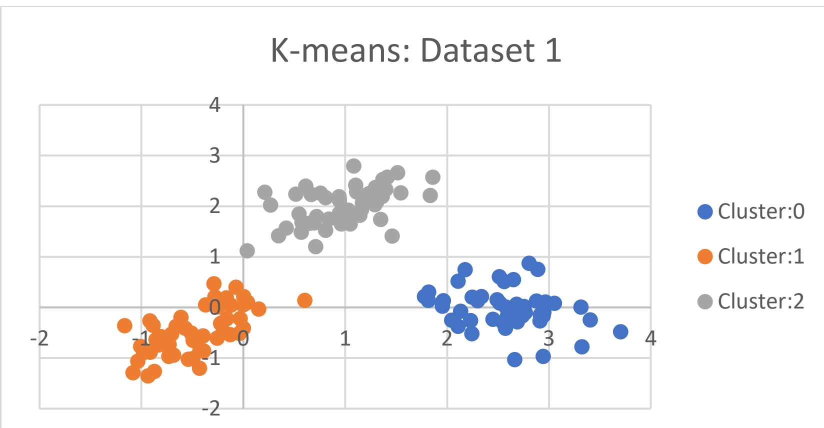
2. K-means

(1):

```
C:\Users\kyles\Desktop\CS145\hw4\CS145_HW3_Release_Python3\HW3_Programming
λ python KMeans.py dataset2.txt
dataset 1:
Iteration :3
Purity is :1.0
NMI :1.0
Cluster 0 size :50
Cluster 1 size :50
Cluster 2 size :50

dataset 2:
Iteration :6
Purity is :0.764
NMI :0.046851365954979234
Cluster 0 size :230
Cluster 1 size :108
Cluster 2 size :162

dataset 3:
Iteration :4
Purity is :0.76
NMI :0.14502499937722388
Cluster 0 size :102
Cluster 1 size :98
```

(2):

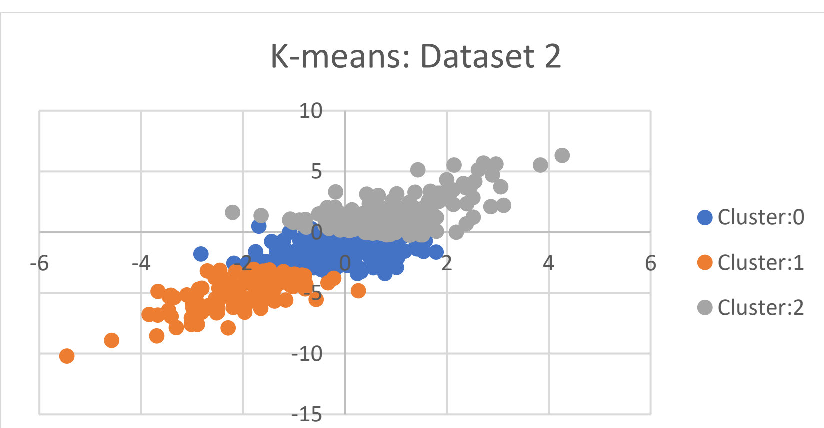
Dataset 1:



Purity: 1.0

NMI: 1.0

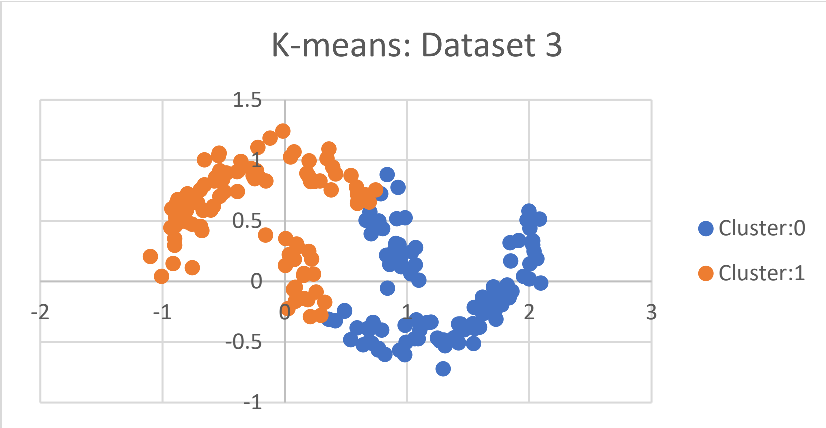
Dataset 2:



Purity: 0.764

NMI: 0.046851365954979234

Dataset 3:



Purity: 0.76

NMI: 0.14502499937722388

(3):

Strengths:

- Efficient: O(tkn) where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.

Weaknesses:

- Applicable only to objects in a continuous n-dimensional space.
- Need to specify k, the number of clusters, in advance.
- Sensitive to noisy data and outliers.
- Not suitable for discovering clusters with non-convex shapes.

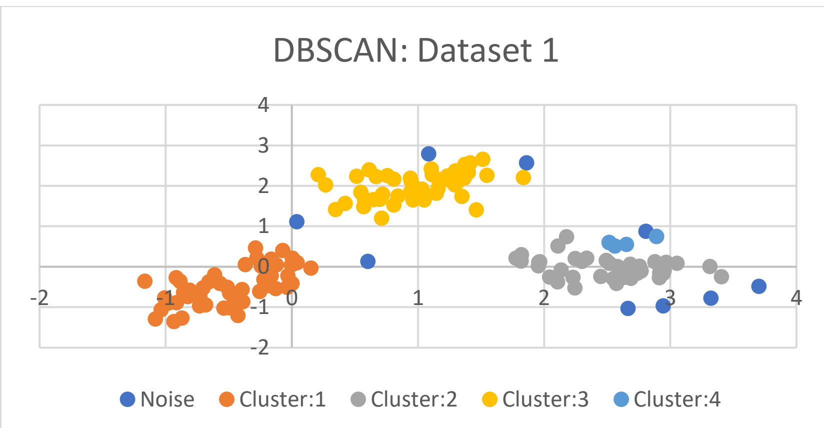
3. DBSCAN

(1):

```
C:\Users\kyles\Desktop\CS145\hw4\CS145_HW3_Release_Python3\HW3_Programming
λ python DBSCAN.py
For dataset1
Esp :0.3560832705047313
Number of clusters formed :4
Noise points :11
Purity is :0.94
NMI :0.9590647490609898
Cluster 0 size :49
Cluster 1 size :41
Cluster 2 size :47
Cluster 3 size :4

For dataset2
Esp :0.4656824188773015
Number of clusters formed :2
Noise points :32
Purity is :0.714
NMI :0.011352036737124684
Cluster 0 size :467
Cluster 1 size :4

For dataset3
Esp :0.18652096476712493
Number of clusters formed :3
Noise points :3
Purity is :0.985
NMI :0.8173489274692755
Cluster 0 size :99
Cluster 1 size :51
Cluster 2 size :47
```

(2):

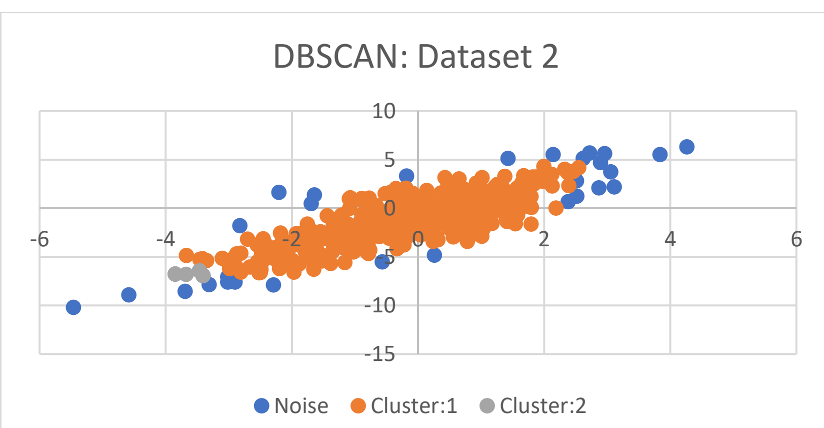
Dataset 1:



Purity: 0.94

NMI: 0.9590647490609898

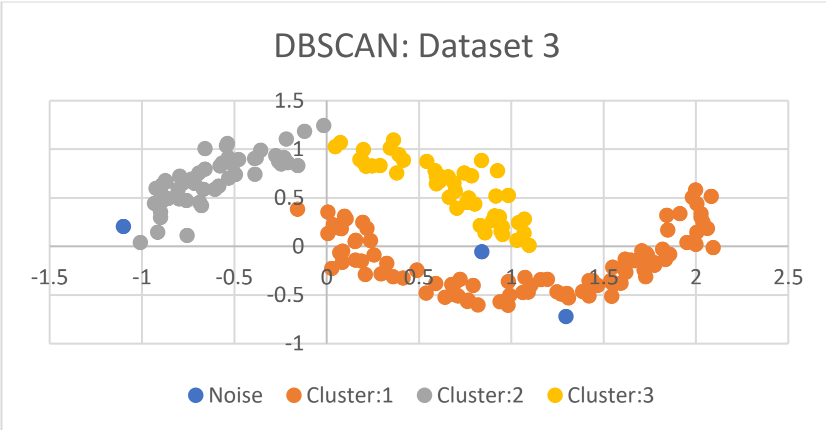
Dataset 2:



Purity: 0.714

NMI: 0.011352036737124684

Dataset 3:



Purity: 0.985

NMI: 0.8173489274692755

(3)

Strengths:

- Has a notion of noise and is resistant to outliers.
- Does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
- Can discover clusters of arbitrary/non-convex shapes.

Weaknesses:

- Not entirely deterministic, depends on the order of visiting data points.
- Cannot cluster data sets with large differences in densities or with high-dimensional data well.

4. GMM

(1)

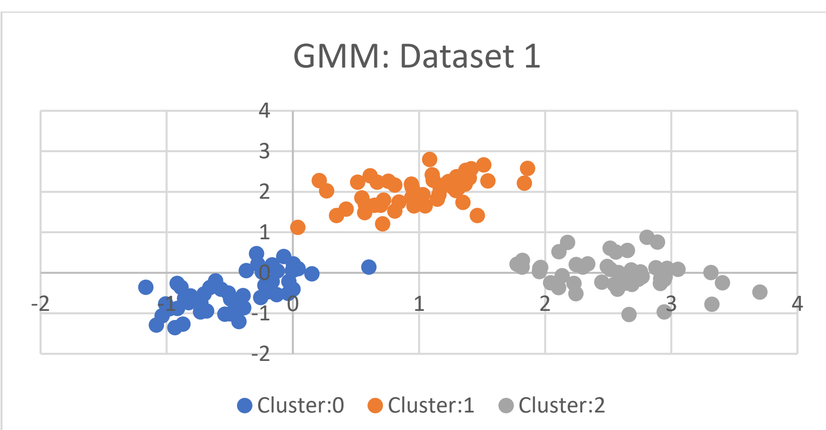
```
C:\Users\kyles\Desktop\CS145\hw4\CS145_HW3_Release_Python3\HW3_Programming
λ python GMM.py
For dataset1
Number of Iterations = 23

After Calculations
Final mean =
-0.46247235077244025  -0.4638732589718203
0.9898953303569601  2.0118020604745777
2.5734294675626277  -0.0271126823762391

Final covariance =
For Cluster : 1
0.1491891185722639  0.11734693334069811
0.11734693334069811  0.21555109631105548

For Cluster : 2
0.1602815973792115  0.07486778264520946
0.07486778264520946  0.13939761616849025

For Cluster : 3
0.18038917499050736  -0.04672187097056099
-0.04672187097056099  0.1520593676641215

Purity is :1.0
NMI :1.0
Cluster 0 size :50
Cluster 1 size :50
Cluster 2 size :50
```

```
C:\Users\kyles\Desktop\CS145\hw4\CS145_HW3_Release_Python3\HW3_Programming
λ python GMM.py

For dataset2
Number of Iterations = 87

After Calculations
Final mean =
-0.7703234448462363  -2.7071464417306315
0.22867541965487978  -0.30418511007085397
0.04847862307749028  -0.378887381602658

Final covariance =
For Cluster : 1
2.1720769845571386  2.9798174034107014
2.9798174034107014  5.62196430461088

For Cluster : 2
1.9572594626303021  3.625806143258927
3.625806143258927  8.013890194694229

For Cluster : 3
0.5439241917453285  -0.1367851216874869
-0.1367851216874869  2.08686453358924

Purity is :0.764
NMI :0.07776497220479159
Cluster 0 size :250
Cluster 1 size :106
Cluster 2 size :144
```

```
C:\Users\kyles\Desktop\CS145\hw4\CS145_HW3_Release_Python3\HW3_Programming
λ python GMM.py

For dataset3
Number of Iterations = 90

After Calculations
Final mean =
0.7473095483981508  0.4578357079369342
0.284036903949417 94  -0.058987655772655516

Final covariance =
For Cluster : 1
0.7685843955293243  -0.2877705231319804
-0.2877705231319804  0.1896045112103438

For Cluster : 2
0.6842343552257826  -0.30083945677053164
-0.30083945677053164  0.17604427879314985

Purity is :0.69
NMI :0.07594783950403936
Cluster 0 size :106
Cluster 1 size :94
```

(2)

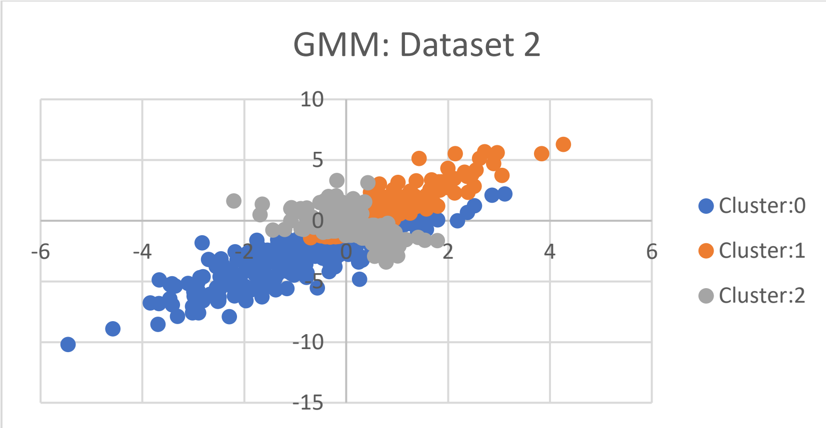
Dataset 1:



Purity: 1.0

NMI: 1.0

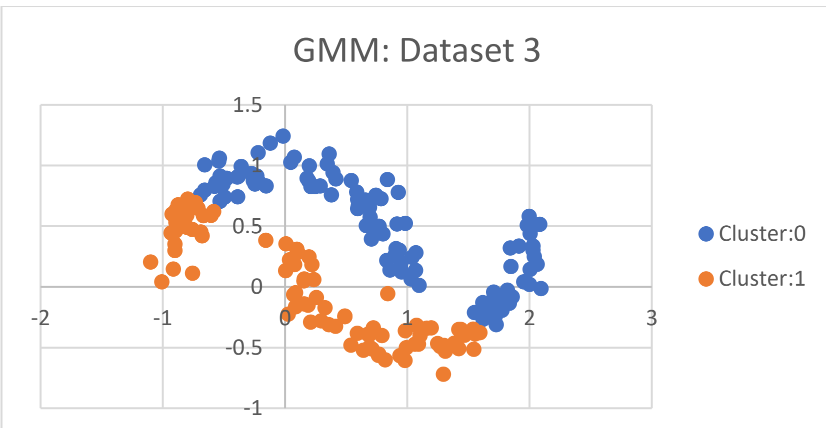
Dataset 2:



Purity: 0.764

NMI: 0.07776497220479159

Dataset 3:



Purity: 0.69

NMI: 0.07594783950403936

(3)

Strengths:

- Mixture models are more general than partitioning: can work well with clusters of different densities and sizes
- Clusters can be characterized by a small number of parameters
- The results may satisfy the statistical assumptions of the generative models

Weaknesses:

- Converges to local optima. (can overcome by running multiple times with random initializations)
- Computationally expensive if the number of distributions is large
- Hard to estimate the number of clusters
- Can only deal with spherical clusters