tutorial-3-1

September 7, 2023

1 MM20B007 Tutorial 3

[99]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering, DBSCAN, SpectralClustering
from sklearn.metrics import adjusted_rand_score, adjusted_mutual_info_score,␣
 ↪silhouette_score
```

1.0.1 Loading the data set

[100]:

```
data = np.load ("/content/drive/MyDrive/sem 7/ID5055/Tutorial 3/test_data.npy",␣
 ↪allow_pickle = True ).item()
# Data is a DICT with keys --- " data " and " labels "

X , true_labels = data["data"], data["labels"]

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Helper Function to plot

[101]:

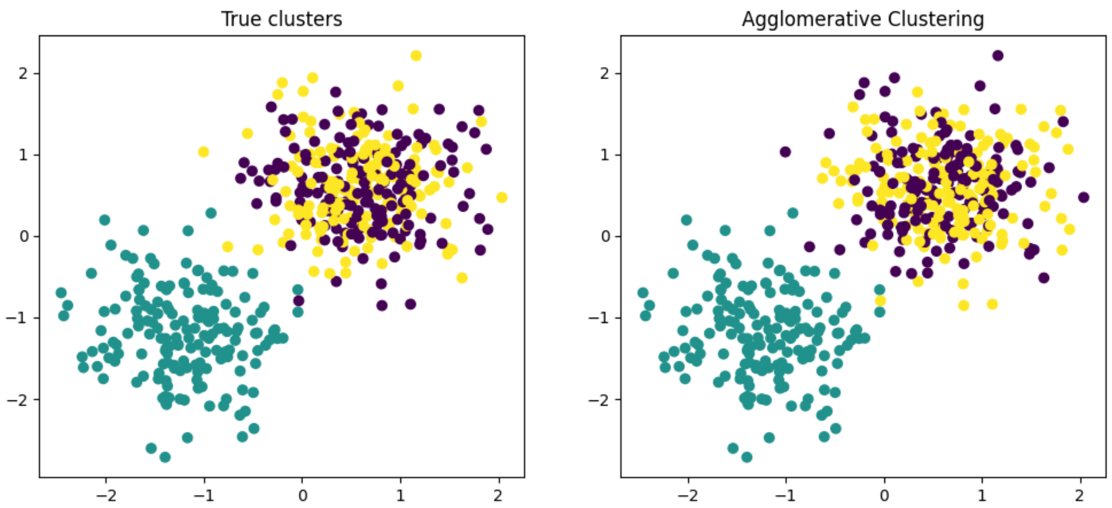
```
def plt_cluster(data, true_labels = None, cluster_labels = None, title_true =␣
 ↪'True clusters', title_cluster = 'Agglomerative clustering'):
  fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (12, 5))
  ax1.scatter(data[:, 0], data[:, 1], c = true_labels)
  ax1.set_title(title_true)

  if cluster_labels is not None:
    ax2.scatter(data[:, 0], data[:, 1], c = cluster_labels)
    ax2.set_title(title_cluster)

  plt.show()
```

1.0.2 Hierarchical Clustering

[102]:

```
# Define clustering algorithms
linkages = ['single', 'complete', 'average']
score = []
for items in linkages:
  # Perform clustering
  hierarchical = AgglomerativeClustering(n_clusters=len(np.
 ↪unique(true_labels)), linkage = items)
  hierarchical_labels = hierarchical.fit(X_scaled).labels_

  # Calculate Rand Score and Mutual Information
  silhouette_score_hierarchical = silhouette_score(X_scaled,␣
 ↪hierarchical_labels)
  rand_score_hierarchical = adjusted_rand_score(true_labels,␣
 ↪hierarchical_labels)
  mi_hierarchical = adjusted_mutual_info_score(true_labels, hierarchical_labels)

  score.append((items, silhouette_score_hierarchical, rand_score_hierarchical,␣
 ↪mi_hierarchical))

print(score)
print("So it is clear that we getting best scores corresponding to complete␣
 ↪linkage")

print(f"Hierarchical Clustering - Silhouette Score: {score[1][1]}")
print(f"Hierarchical Clustering - Rand Score: {score[1][2]}")
print(f"Hierarchical Clustering - Mutual Information: {score[1][3]}")
print('\n')

# Scatter plots for each clustering algorithm
hierarchical = AgglomerativeClustering(n_clusters=len(np.unique(true_labels)),␣
 ↪linkage = 'complete')
hierarchical_labels = hierarchical.fit(X_scaled).labels_
plt_cluster(X_scaled, true_labels, hierarchical_labels, title_cluster =␣
 ↪'Agglomerative Clustering')
```

```
[('single', -0.11912125769335813, 9.638785547720615e-08, 7.183077256948358e-08),
('complete', 0.5878535006929564, 1.0, 1.0), ('average', 0.5852277876646795,
0.9820733060105458, 0.9690231356896917)]
So it is clear that we getting best scores corresponding to complete linkage
Hierarchical Clustering - Silhouette Score: 0.5878535006929564
Hierarchical Clustering - Rand Score: 1.0
Hierarchical Clustering - Mutual Information: 1.0
```

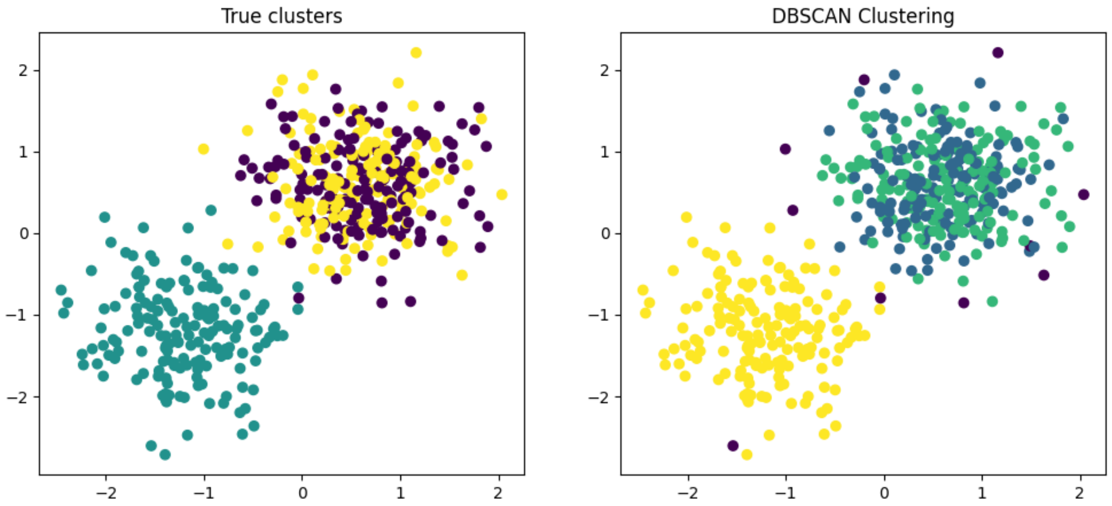


1.0.3 DBSCAN

[103]:

```
# Define clustering algorithms
score = []
for i in range(4, 10):
  e = i/10
  s = 20
  dbscan = DBSCAN(eps= e, min_samples= s)

  # Perform clustering
  dbscan_labels = dbscan.fit(X_scaled).labels_

  # Calculate Rand Score and Mutual Information
  silhouette_score_dbscan = silhouette_score(X_scaled, dbscan_labels)
  rand_score_dbscan = adjusted_rand_score(true_labels, dbscan_labels)
  mi_dbscan = adjusted_mutual_info_score(true_labels, dbscan_labels)

  score.append((e, s, silhouette_score_dbscan, rand_score_dbscan, mi_dbscan))



print(score)
print('So it is clear that we getting best scores corresponding to eps = 0.7␣
 ↪when min_sample is 20')

# Print Silhouette score, Rand Scores and Mutual Information
print(f'Scores corresponding to eps: 0.7, and min smaple: 20')
print("DBSCAN - Silhouette Score:", score[3][2])
print("DBSCAN - Rand Score:", score[3][3])
print("DBSCAN - Mutual Information:", score[3][4])
print('\n')

# Scatter plots for each clustering algorithm
dbscan = DBSCAN(eps= 0.7, min_samples= 20)
dbscan_labels = dbscan.fit(X_scaled).labels_
plt_cluster(X_scaled, true_labels, dbscan_labels, title_cluster = 'DBSCAN␣
 ↪Clustering')
```

```
[(0.4, 20, 0.0404209543877421, 0.23307310461933817, 0.4535500901775322), (0.5,
20, 0.42032584991504635, 0.7001176783567878, 0.7209125200224381), (0.6, 20,
0.5360100701930632, 0.8978231910163619, 0.8664956184470873), (0.7, 20,
0.5679472142355035, 0.9640147403798555, 0.9388051939549018), (0.8, 20,
0.3387959034258973, 0.565999509245958, 0.7105439436507259), (0.9, 20,
0.16680089616415433, 2.4434993885939573e-05, 0.00037653689428283096)]
So it is clear that we getting best scores corresponding to eps = 0.7 when
min_sample is 20
Scores corresponding to eps: 0.7, and min smaple: 20
DBSCAN - Silhouette Score: 0.5679472142355035
DBSCAN - Rand Score: 0.9640147403798555
DBSCAN - Mutual Information: 0.9388051939549018
```

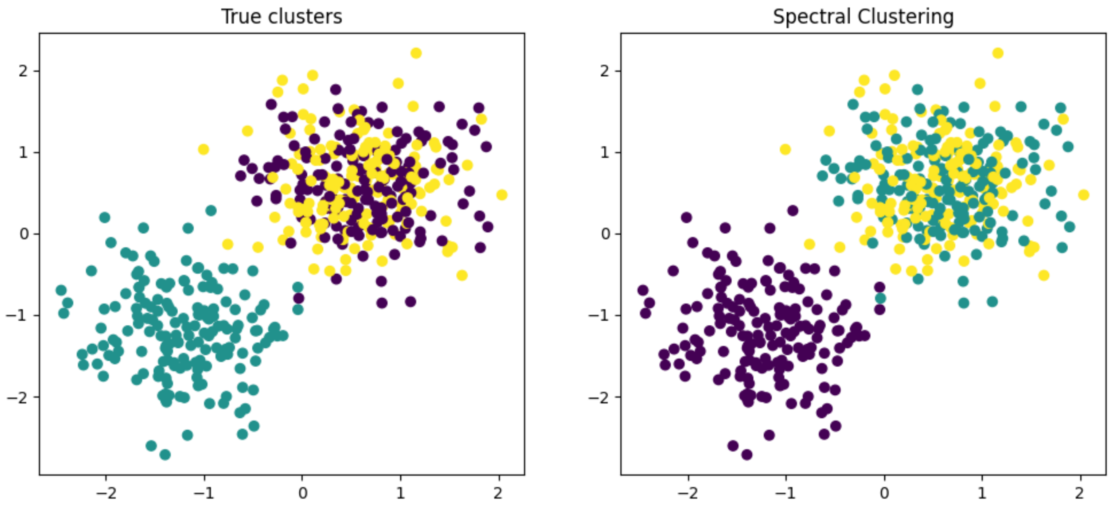


1.0.4 Spectral Clustering

[104]:
```
# Define clustering algorithms
spectral = SpectralClustering(n_clusters=len(np.unique(true_labels)), affinity␣
 ↪= 'nearest_neighbors')
```

```
# Perform clustering
spectral_labels = spectral.fit(X_scaled).labels_

# Calculate Rand Score and Mutual Information
silhouette_score_spectral = silhouette_score(X_scaled, spectral_labels)
rand_score_spectral = adjusted_rand_score(true_labels, spectral_labels)
mi_spectral = adjusted_mutual_info_score(true_labels, spectral_labels)

# Print Silhouette score, Rand Scores and Mutual Information
print("Spectral Clustering - Silhouette Score:", silhouette_score_spectral)
print("Spectral Clustering - Rand Score:", rand_score_spectral)
print("Spectral Clustering - Mutual Information:", mi_spectral)
print('\n')

# Scatter plots for each clustering algorithm
plt_cluster(X_scaled, true_labels, spectral_labels, title_cluster = 'Spectral␣
 ↪Clustering')
```

```
Spectral Clustering - Silhouette Score: 0.5878535006929564
Spectral Clustering - Rand Score: 1.0
Spectral Clustering - Mutual Information: 1.0
```



1.0.5 Explain the ambiguity in Silhoutte Scores.

The Silhouette Score measures how similar an object is to its own cluster compared to other clusters. It provides a measure of how well-separated the clusters are. The score ranges from -1 to +1, where:

A high positive value (close to +1) indicates that the samples are well clustered, with clear and distinct boundaries between clusters. A value near 0 indicates overlapping clusters or that the data points are very close to the decision boundary between clusters. A negative value (close to -1) suggests that the samples may have been assigned to the wrong clusters.

In the results, we have observed that both Hierarchical Clustering and Spectral Clustering have the same Silhouette Score, which is 0.5878535006929564, while DBSCAN has a slightly lower Silhouette Score of 0.5679472142355035. These scores suggest that, on average, the data points within clusters are relatively close to each other, and there is some degree of separation between clusters.

1.0.6 Explore and Compare Linkage Techniques (Optional)

Single Linkage defines the distance between two clusters as the minimum distance between any two points in each cluster. It tends to create long, string-like clusters and can be sensitive to noise.

Silhouette Score: -0.11912125769335813,
Adjusted Rand Score: 9.638785547720615e-08,
Adjusted Mutual Information Score: 7.183077256948358e-08

A negative value indicates that the clusters may be poorly defined. Very low values suggest that the single linkage method doesn't perform well on your data, possibly due to the string-like clusters it creates.

Complete Linkage Complete linkage defines the distance between two clusters as the maximum distance between any two points in each cluster. It tends to create compact, spherical clusters.

Silhouette Score: 0.5878535006929564,
Adjusted Rand Score: 1.0,
Adjusted Mutual Information Score: 1.0

A high positive value indicates well-separated clusters, which is the case here. The Rand Score and Mutual Information are both perfect, indicating that the complete linkage method produced clusters that are in perfect agreement with the ground truth or reference clusters.

Average Linkage: Average linkage defines the distance between two clusters as the average distance between all pairs of points, one from each cluster.

Silhouette Score: 0.5852277876646795,
Adjusted Rand Score: 0.9820733060105458,
Adjusted Mutual Information Score: 0.9690231356896917

The Silhouette Score is high and positive, suggesting that average linkage produces well-defined and separated clusters. The Rand Score and Mutual Information are close to 1.0, indicating that the clustering is in good agreement with the reference clusters but not perfect, as in the case of complete linkage.