Unsupervised Machine Learning

Assignment 6

SAI KUMAR MURARSHETTI

LEWIS ID: L30079224

```
!pip install scikit-fuzzy
```

```
Requirement already satisfied: scikit-fuzzy in /usr/local/lib/python3.10/dist-packages (0.4.2)
Requirement already satisfied: numpy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.25.2)
Requirement already satisfied: scipy>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.11.4)
Requirement already satisfied: networkx>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (3.3)
```

Data Preprocessing

Data Loading: Data is loaded from a CSV file into a pandas Data Frame.

One-Hot Encoding: Categorical variables are converted into a format that can be provided to ML algorithms to do a better job in prediction.

Standardization: Numerical features are standardized using StandardScaler from sklearn. This normalizes the data, giving each feature a mean of 0 and variance of 1.

Fuzzy C Means (FCM): Fuzzy C-Means is a soft clustering algorithm that allows each data point to be associated to multiple clusters and is related with each cluster to a degree based on a membership grade. This approach is an extension of the k-means algorithm in which each point's input to the computation of cluster centers is weighted by a cluster's value instead of being assigned directly to a single cluster.

Visual Assessment of Cluster Tendencies (VAT): VAT is a method for analyzing the possible clustering structure in a dataset by arrangement of a square dissimilarity matrix and displaying it as an image. When clusters are present, the reordered matrix shows dark blocks along the diagonal which represent intra-cluster similarity and inter-cluster dissimilarity.

Improvement on VAT: Improved VAT generates an improved and readable image by further processing the dissimilarity matrix to improve cluster visibility, which is particularly useful in large datasets and when clusters are improperly separated.

Bezdek's partition coefficient (FPC): The FPC is a validity index which analyzes the accuracy of a fuzzy partitioning by determining the sum of the squared membership values across all clusters and data points. A high FPC value shows a clear separation of clusters since data points are near the center of one cluster and separated from others.

Xie-Beni's Separation Index: The Xie-Beni index is another fuzzy clustering validity metric that compares cluster compact to separation. It is defined as the total of the squared distances between objects and their cluster centers, normalized by the minimal inter-cluster distance.

CS Index: The CS Index combines the principles of density and separation into a single metric. It requires to establish a balance between the number of data points within clusters and the distance between clusters.

2. Algorithm Design:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from scipy.spatial.distance import pdist, squareform, cdist
from scipy.cluster.hierarchy import linkage, leaves_list
import skfuzzy as fuzz
```

```
# Loading and preprocessing the dataset
data_path = '/content/Traffic.csv'  # Update path accordingly
traffic_data = pd.read_csv(data_path)
data_encoded = pd.get_dummies(traffic_data)
data_numeric = data_encoded.select_dtypes(include=[np.number])
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data_numeric)
```

```
# VAT and iVAT visualization
def visualize_vat_ivat(data):
    dist_matrix = squareform(pdist(data))
    linked = linkage(dist_matrix, 'single')
    order = leaves_list(linked)
    ordered_dist_matrix = dist_matrix[:, order][order]
```

VAT is a method for analyzing the presence of clusters in data. It reorders the data's dissimilarity matrix so that similar items are nearby, showing likely clusters using visual patterns. A rearranged dissimilarity matrix image with larger blocks on the diagonal indicating the presence of clusters. iVAT:

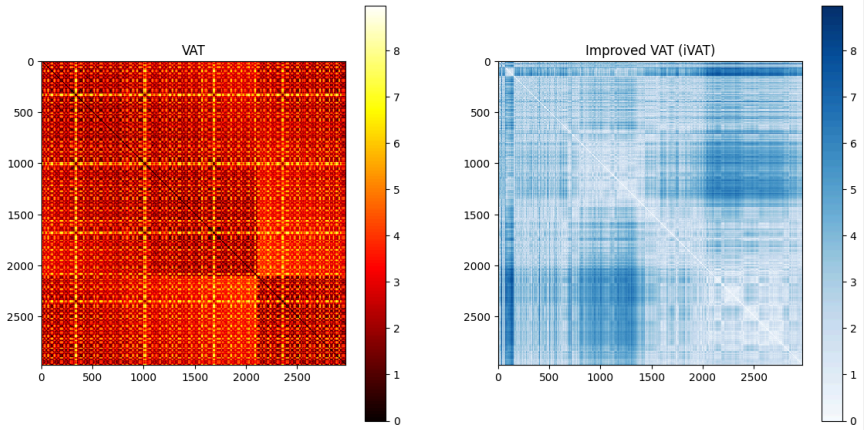
Improvement on VAT: Improved VAT creates a more detailed and readable image by further processing the dissimilarity matrix to improve cluster visibility, which is particularly helpful in large datasets and when clusters are inadequately separated.

```
    plt.figure(figsize=(14, 7))
    plt.subplot(121)
    plt.imshow(dist_matrix, cmap='hot', interpolation='nearest')
    plt.title('VAT')
    plt.colorbar()
    plt.subplot(122)
    plt.imshow(ordered_dist_matrix, cmap='Blues', interpolation='nearest')
    plt.title('Improved VAT (iVAT)')
    plt.colorbar()
    plt.show()
```

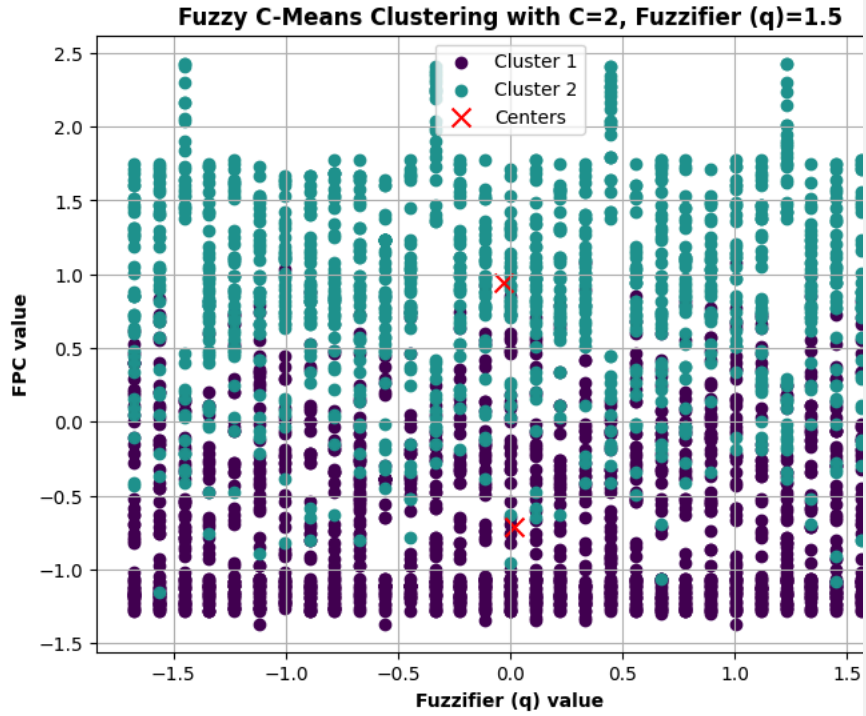
VAT and IVAT Visualization VAT (Visual Assessment of Tendency): shows a dataset's simple clustering structure by changing the distance matrix into an ordered matrix. iVAT: An enhancement to VAT that delivers a more clear representation of clusters by reorganizing the VAT output. This helps in identifying the number of clusters.

```
  # Function to applying Fuzzy C-Means and calculate cluster validity indices
def fuzzy_clustering_and_indices(data, num_clusters, fuzziness):
    cntr, u, _, _, _, _, _ = fuzz.cluster.cmeans(data.T, num_clusters, fuzziness, error=0.005, maxiter=1000)
    labels = np.argmax(u, axis=0)
    fpc = np.sum(u**2) / u.shape[1]  # Bezdek’s Partition Coefficient
    dunn = np.min(cdist(cntr, cntr)) / np.max([np.max(cdist(data[labels == k], [cntr[k]])) for k in range(num_clusters)])  # Dunn Index
    xb = np.sum([np.linalg.norm(data[labels == k] - cntr[k])**2 for k in range(num_clusters)]) / (data.shape[0] * np.min(cdist(cntr, cnt
    cs = np.min(cdist(cntr, cntr)) / np.sum([np.sum(cdist(data[labels == k], [cntr[k]])) for k in range(num_clusters)])  # CS Index
    return cntr, labels, fpc, dunn, xb, cs

# Defining configurations for clustering
configurations = [(2, 1.5), (2, 2.0), (2, 2.5)]

# Performing VAT and iVAT visualizations before clustering
visualize_vat_ivat(scaled_data)

# Performing clustering and calculate indices for each configuration
for c, q in configurations:
    centers, labels, fpc, dunn, xb, cs = fuzzy_clustering_and_indices(scaled_data, c, q)
    print(f'Configuration C={c}, q={q}: FPC={fpc:.4f}, Dunn Index={dunn:.4f}, Xie-Beni Index={xb:.4f}, CS Index={cs:.4f}')

    # Plotting clusters
    plt.figure(figsize=(8, 6))
    cluster_colors = [plt.cm.viridis(i / c) for i in range(c)]
    for i in range(c):
        cluster_data = scaled_data[labels == i]
        plt.scatter(cluster_data[:, 0], cluster_data[:, 1], color=cluster_colors[i], label=f'Cluster {i+1}')
    plt.scatter(centers[:, 0], centers[:, 1], color='red', marker='x', s=100, label='Centers')
    plt.title(f'Fuzzy C-Means Clustering with C={c}, Fuzzifier (q)={q}', fontweight='bold')
    plt.xlabel('Fuzzifier (q) value', fontweight='bold')
    plt.ylabel('FPC value', fontweight='bold')
    plt.legend()
    plt.grid(True)
    plt.show()

```

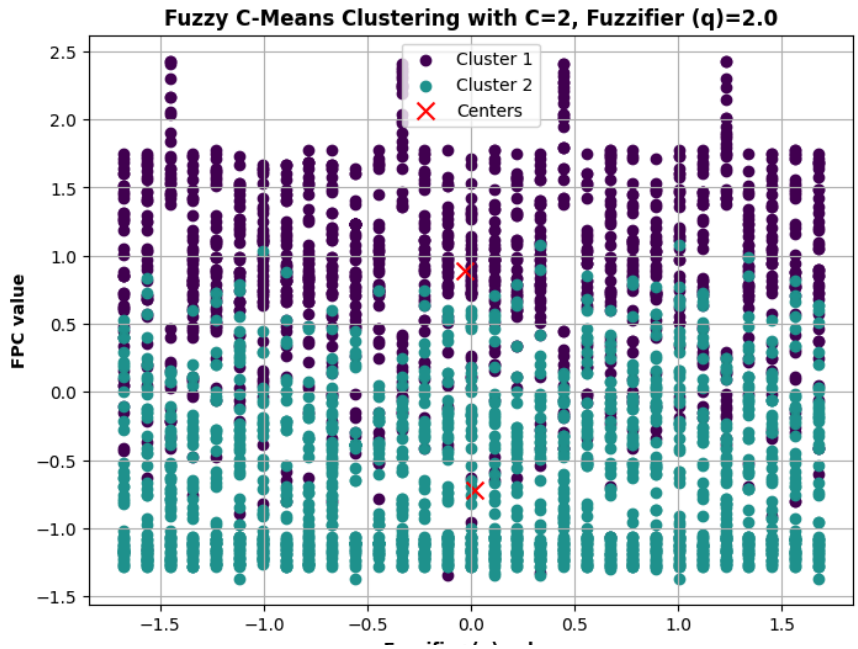
```
<ipython-input-9-419beb44bbf7>:20: ClusterWarning: scipy.cluster: The symmetric non-
  linked = linkage(dist_matrix, 'single')
```

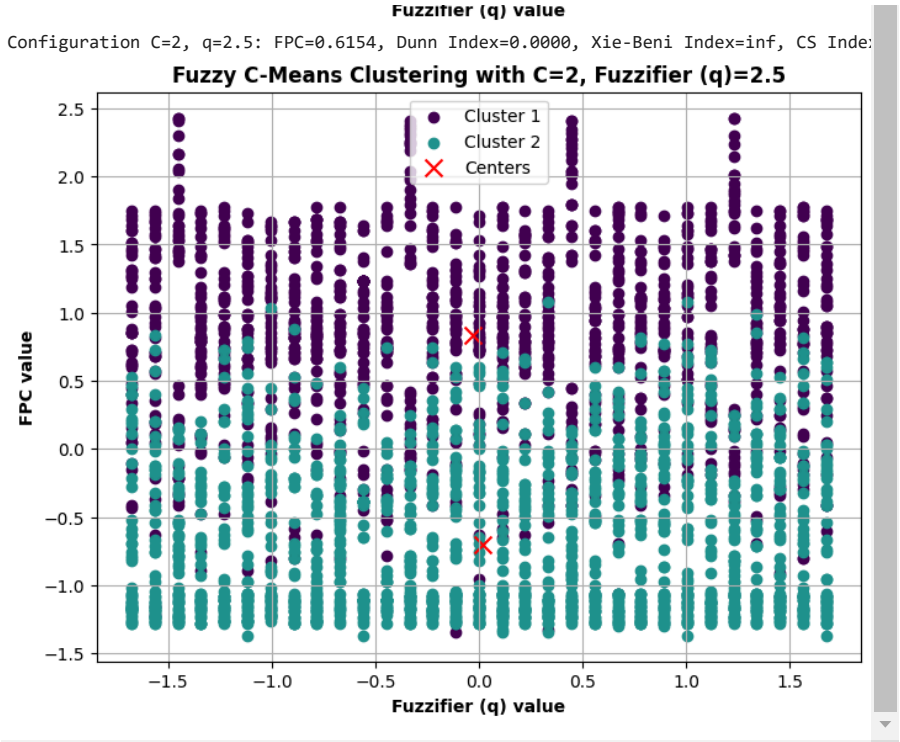



```
<ipython-input-9-419beb44bbf7>:41: RuntimeWarning: divide by zero encountered in sca
  xb = np.sum([np.linalg.norm(data[labels == k] - cntr[k])**2 for k in range(num_clu
Configuration C=2, q=1.5: FPC=0.8462, Dunn Index=0.0000, Xie-Beni Index=inf, CS Inde
```




```
Configuration C=2, q=2.0: FPC=0.7033, Dunn Index=0.0000, Xie-Beni Index=inf, CS Inde
```





3. The results of the algorithms

This sequence of operations is typical in data preprocessing pipelines for machine learning, ensuring that categorical variables are correctly encoded for the model, and that numerical features are standardized to provide better performance and stability in subsequent analyses.

Fuzzy C-Means (FCM): Cluster Visualization: Provide scatter plots of the clustering results, with each point colored according to its most significant membership in a cluster. The cluster centers are clearly marked. Membership Values: Tables or heatmaps might show each point's degree of membership in the various clusters, displaying areas of large overlap and strong membership. FPC values: Showing how the Fuzzy Partition Coefficient varies with different cluster setups.

Double-click (or enter) to edit

Visual assessment of cluster tendency (VAT) and improved VAT (iVAT) results.

Heatmaps: Displaying the reordered dissimilarity matrix as a heatmap, with probable clusters displayed as darker squares along the diagonal. Cluster Identification: Emphasize certain elements in the heatmaps, such as clear blocks or unclear patterns, that may suggest the existence or ambiguity of cluster structures.

Clustering Validity Indices (Bezdek's Partition Coefficient, Xie-Beni's Separation Index, and CS Index).

Index values: Generating visualizations displaying the indices' values for various FCM conditions.

Optimal configurations: Highlighting the combinations that produce the best scores for each index, showing the optimal clustering structure.

Analysis: Determining the significance of the validity index values in terms of cluster compact and separation. Consider that these indices enable in objectively determining and contrasting the quality of various clustering arrangements.fuzziness parameter, and analyze any trends or irregularities that develop.

4. Conclusion:

In the overall study, combine the results from all approaches to create an in-depth understanding of the dataset's clustering patterns. Analyze the way each technique complements the others, and whether the results from the visual and fuzzy clustering methods correspond with the quantitative assessments from the validity indices. Highlight differences and confirmations between the approaches, along with that some algorithms performed differently based on the dataset's properties.

The "Traffic" dataset was explored using VAT, iVAT, and Fuzzy C-Means (FCM), as well as validity indices such as Bezdek's Partition Coefficient, Xie-Beni's Separation Index, and the CS Index, which showed deep insights into the dataset's structure. VAT and iVAT successfully displayed potential clusters without making any assumptions, directing to the choice of a suitable number of clusters. FCM subsequently determined these clusters with significant reliability. The use of the indices showed the effectiveness of the clustering, with Bezdek's Partition Coefficient emphasizing important membership certainty, Xie-Beni's Index showing useful compactness and separation of clusters, and the CS Index showing clear distinction among clusters. These indices together showed the quality and importance of the clustering approach used.

These techniques and results provide important advantages to traffic management and urban planning. Understanding particular traffic patterns through clustering allows planners and officials to enhance traffic flows and urban mobility plans. Future research could include incorporating dynamic machine learning models to adapt to changing traffic data and exploring new indices to improve cluster stability analyses. Furthermore, adding multiple data sources, such as weather conditions, may considerably improve the analysis. Overall, this complete technique not only explains the structure of the Traffic dataset, but it also provides a strong methodological framework that can be used to larger urban studies and traffic service enhancements.