Unsupervised Machine Learning

Assignment 4

SAI KUMAR MURARSHETTI

LEWIS ID: L30079224

1. Technical Description of Techniques Used.

Visual Assessment of Clustering tendency (VAT):

VAT is a technique for visually assessing when clusters exist in a dataset and determining the number of clusters through analysis of an ordered dissimilarity images. The method involves creating a combination dissimilarity matrix for all data points and then rearranging it to highlight the cluster structure.

iVAT: Improved Visual Assessment of Tendency iVAT is an upgraded version of VAT that provides better and more readable visuals, particularly when data has different densities or contains noise. iVAT modifies the way the dissimilarity matrix is processed, frequently adding methods for dealing with noise and outliers more effectively. The final result is a clearer image that makes it easier to recognize clusters and their separations, which is particularly important in larger datasets.

Data Manipulation and Analysis Tools:

Pandas: Used for loading and preprocessing data.

NumPy: Employed for numerical operations on arrays.

SciPy: Utilized for generating and manipulating the dissimilarity matrix and for hierarchical clustering to order the matrix.

Matplotlib: Used for visualization of the dissimilarity matrices.

```
# Import necessary libraries
import pandas as pd
import numpy as np
from scipy.spatial.distance import pdist, squareform
from scipy.cluster.hierarchy import linkage, leaves_list
import matplotlib.pyplot as plt

```

Import data from Traffic.csv.

```
# Loading the dataset
data = pd.read_csv('Traffic.csv')
```

```
data.head()
```

	Time	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation
0	12:00:00 AM	10	Tuesday	31	0	4	4	39	low
1	12:15:00 AM	10	Tuesday	49	0	3	3	55	low
2	12:30:00 AM	10	Tuesday	46	0	3	6	55	low
3	12:45:00 AM	10	Tuesday	51	0	2	5	58	low
4	1:00:00 AM	10	Tuesday	57	6	15	16	94	normal

Next steps: Generate code with `data` View recommended plots

The dataset contains information about traffic counts by vehicle type, recorded over different times and dates. Here's the structure of the dataset:

1. Time: The specific time of the record.
2. Date: The date of the record.
3. Day of the week: The day of the week.
4. CarCount: The number of cars counted.
5. BikeCount: The number of bikes counted.

6. BusCount: The number of buses counted.
7. TruckCount: The number of trucks counted.
8. Total: Total count of vehicles.
9. Traffic Situation: Qualitative assessment of traffic (low, normal, high)

2. Design of the Algorithms

Preprocessing Data: Convert categorical variables to numeric using one-hot encoding. Exclude non-numeric columns such as 'Time'.

The objective of one-hot encoding is to encode categorical information into a format that machine learning algorithms can use to predict more accurately. This function turns categorical variable(s) into numerous binary columns, each one of which represents the category's presence (1) or absence (0). The number of binary columns equals the number of categories identified in the original column.

```
# Handling categorical variables using one-hot encoding
data_encoded = pd.get_dummies(data)

#  Using numeric data
data_numeric = data_encoded.select_dtypes(include=[np.number])
```

data_numeric: A numeric DataFrame or 2D NumPy array where each row represents an observation and each column represents a feature.

metric='euclidean': Specifies that the Euclidean distance metric should be used. The Euclidean distance is the "ordinary" straight-line distance between two points in Euclidean space.

```
# Computing the dissimilarity (distance) matrix
distances = pdist(data_numeric, metric='euclidean')
dist_matrix = squareform(distances)
```

```
# Performing hierarchical clustering to reorder the distance matrix
linked = linkage(distances, method='single')
order = leaves_list(linked)
ordered_dist_matrix = dist_matrix[:, order][order]
```

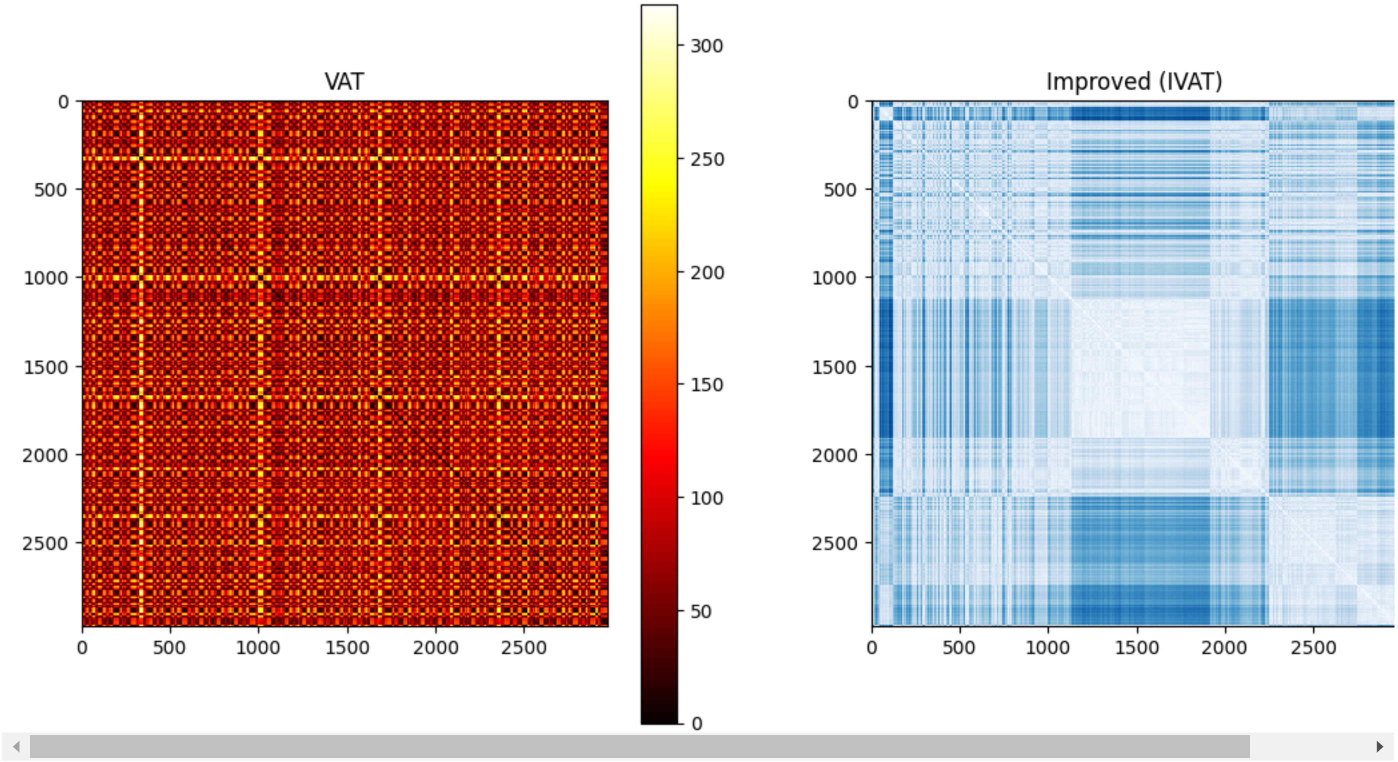
3. The results of the algorithms:

For applying VAT and iVAT, we will use the numeric features related to vehicle counts (CarCount, BikeCount, BusCount, TruckCount, Total) as they are most relevant for clustering analysis. We'll disregard qualitative and temporal aspects like time, date and traffic situation for this analysis.

```
# Visualize the original and ordered dissimilarity matrices
plt.figure(figsize=(14, 7))
# Original VAT
plt.subplot(121)
plt.imshow(dist_matrix, cmap='hot', interpolation='nearest')
plt.title('VAT')
plt.colorbar()

# Ordered iVAT
plt.subplot(122)
plt.imshow(ordered_dist_matrix, cmap='Blues', interpolation='nearest')
plt.title('Improved (IVAT)')
plt.colorbar()
plt.show()

```



4. Conclusion:

The VAT and iVAT visualizations obtained from the 'traffic.csv' dataset indicate a complex clustering structure with no identifiable, well-defined groups. The VAT image shows the presence of several, smaller potential clusters with changing intensities rather than clear-cut, separate clusters. This complex appears in the iVAT image, as the greater brightness fails to recognize massive, black blocks along the diagonal, indicating strong cluster boundaries. Instead, we see an understated pattern with no apparent distinction between clusters, suggesting a dataset with potentially overlapping subgroups or a high level of similarity in traffic patterns. The absence of apparent grouping indicates that the traffic conditions recorded in the dataset are difficult to split into different categories, reflecting the sensitive nature of traffic flow and its influencing factors.

As a result, a more complex analytical method is required to understand the dataset's structure, whether including complex clustering techniques capable of handling overlapping groups and noise and time-series analysis to capture trends in time. These results show that instead of simply relying on essential segmentation, practical traffic management applications need complex solutions that can address the complex, ongoing variations in traffic conditions.