

ISYE6740 Homework 1

Joel Quek
jquek7@gatech.edu

SECTION 1 - CONCEPT QUESTIONS

Question 1 - Differences between Supervised and Unsupervised Learning

Learning Mode	Benefits	Drawbacks
Supervised Learning	<ul style="list-style-type: none">Supervised learning can achieve high accuracy when the models are properly trained with a substantial amount of labelled data.	<ul style="list-style-type: none">It requires a large amount of labelled data, which can be expensive and time-consuming to acquire.
	<ul style="list-style-type: none">It provides predictable results as the model is explicitly trained to predict the output based on the input.	<ul style="list-style-type: none">There is a risk of overfitting to the training data, especially if not enough generalization is done, which can make the model perform poorly on unseen data
Unsupervised Learning	<ul style="list-style-type: none">It can discover hidden patterns or intrinsic structures in data that are not initially evident.	<ul style="list-style-type: none">The results can sometimes be unpredictable as there is no explicit guidance on what the output should look like.
	<ul style="list-style-type: none">It does not require labelled data, which can be particularly useful when labels are unavailable or difficult to obtain.	<ul style="list-style-type: none">It can be challenging to evaluate the performance of the model since there are no true labels to compare against.

Question 2 - Defining a Similarity Function for Mixed Categorical and Real-Valued Data Points

Each data point X^i has three features which can be written in a ternary form:

("City Name", "Property Type", "Price")

Feature Type	Specific Feature(s)	Method	Distance Measure
Categorical Feature	City Name Property Type	Convert these categorical features into one-hot encoded vectors. For example, "Atlanta" and "San Francisco" can be encoded into binary vectors where each vector has a dimension for every possible city and property type in the dataset.	Use Hamming distance to measure the similarity between these binary vectors. The Hamming distance counts the number of positions at which the corresponding symbols are different.
Numerical Feature	Price of Property	Since price is a numerical value, you can use the Euclidean distance to measure differences between prices.	Calculate the Euclidean distance between the price values of x^i and x^j

To combine these different types of measures (Hamming for categorical data and Euclidean for numerical data), you can use a weighted sum or another method that accounts for the different scales and types of data.

$$d(x^i, x^j) = w_1 \cdot \text{Hamming}(x^{i,cat}, x^{j,cat}) + w_2 \cdot \text{Euclidean}(x^{i,num}, x^{j,num})$$

Where, for example, $x^{i,cat}$ refers to the specific category of the i th data sample. Every Hamming Distance (resp. Euclidean Distance) calculation will calculate the same category (resp. numerical variable) from two different data indices.

Choice of weights can be a result of feature importance; one example could be real-world domain knowledge about which type of feature (categorical or numerical) is more important for the task.

Question 3 - Equivalence of Clustering Assignment Formulations

We calculate the squared Euclidean distance between a specific data point x^i and c^j ,

$$\|x^i - c^j\|^2, \forall j = 1, \dots, k$$

and take the minimum.

The difference vector $x^i - c^j$ is such that

$$x^i = (x_1^i, x_2^i, \dots, x_n^i) \text{ where } x^i \in \mathbb{R}^n$$

$$c^j = (c_1^j, c_2^j, \dots, c_n^j) \text{ where } c^j \in \mathbb{R}^n$$

$$\text{Hence, } x^i - c^j = (x_1^i - c_1^j, x_2^i - c_2^j, x_3^i - c_3^j, \dots, x_n^i - c_n^j)$$

The squared Euclidean Distance is

$$\begin{aligned} \|x^i - c^j\|^2 &= \sum_{k=1}^n (x_k^i - c_k^j)^2 \\ &= (x^i - c^j)^T (x^i - c^j) \\ &= (x^i)^T x^i - (x^i)^T c^j - (c^j)^T x^i + (c^j)^T c^j \\ &= (x^i)^T x^i - 2(c^j)^T x^i + (c^j)^T c^j \end{aligned}$$

The last line is true since $(x^i)^T c^j = (c^j)^T x^i$ and both are in \mathbb{R}^1 .

Because x^i is fixed, $(x^i)^T x^i$ is a constant for all values of $j = 1, \dots, k$

Hence we just have to minimise $-2(c^j)^T x^i + (c^j)^T c^j$ (Result (i))

We multiply the constant $\frac{1}{2}$ to *Result (i)* to get

$$\begin{aligned}
 & \frac{1}{2}[-2(c^j)^T x^i + (c^j)^T c^j] \\
 &= -(c^j)^T x^i + \frac{1}{2}(c^j)^T c^j \\
 &= \frac{1}{2}(c^j)^T c^j - (c^j)^T x^i \\
 &= (c^j)^T \left[\frac{1}{2}c^j - x^i \right]
 \end{aligned}$$

Which is the function to minimise in the second cluster assignment function in the question. Hence,

$$\pi(i) = \arg \min_{j=1, \dots, k} \left((c^j)^T \left(\frac{1}{2}c^j - x^i \right) \right)$$

Note: Multiplying *Result (i)* by $\frac{1}{2}$ is acceptable as it is a *positive* constant and will not affect the result of our minimisation attempts.

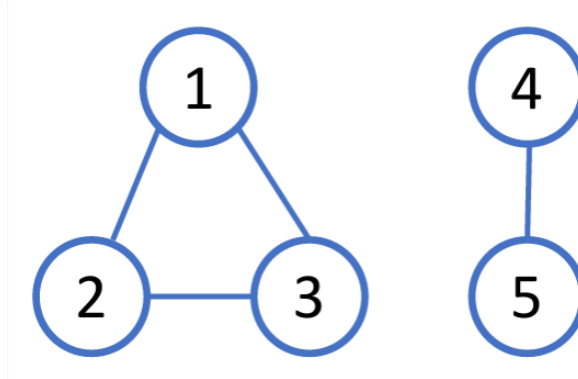
Question 4 - Impact of Initializations on K-Means Clustering Results

- In k-means, each cluster center c_j is iteratively updated to be the mean of the data points assigned to it. Thus, the initial centers influence the first round of assignments of data points to clusters, which in turn affects the updated positions of the centers and every subsequent assignment and update step. If the initial centers are not well-placed or representative of the data's structure, the algorithm might converge to a less desirable clustering.
- K-means clustering aims to minimize the sum of squares within each cluster. However, the sum of squares function is not convex except in some very specific cases, and thus k-means is prone to finding local minimum points. The initial positions of the centers, which are usually selected randomly unless specified otherwise, largely determine the local minimum that the algorithm converges to. Different initializations can lead the algorithm to different local minimum values because each initialization starts the optimization from a different point in the solution space.

Question 5 - Reason for Guaranteed Finite Termination of K-Means Algorithm

1. There is only a finite number of possible assignments of each data point as each data point can only belong to one of the k clusters. Given m data points and k clusters, these figures are discrete and finite.
2. The cluster assignment function $\pi(i) = \arg \min_{j=1, \dots, k} \|x^i - c^j\|^2$ uses Euclidean distance which is a nonnegative quantity. The algorithm either reassigns a new cluster centre or leaves it unchanged.
3. The algorithm will achieve a steady-state eventually because the cluster centers will be chosen in such a way that the next data-point would lead to a minimal sum-of-squares within each cluster.
4. If a new data point creates an increase in value of the sum-of-squares within a cluster, that solution would be rejected because it goes against the premise of k-means clustering.
5. Hence, because every cluster has a best solution, then every stage of k-means would either move towards that best solution or remain unchanged because that best solution has been achieved. Thus, the algorithm will terminate.

Question 6 - Eigenvectors of Laplacian Matrix



Vertices = {1, 2, 3, 4, 5}

Edges = {(1, 2), (2, 3), (1, 3), (4, 5)}

Adjacency Matrix

Given the edges:

- Vertices 1, 2, and 3 are fully connected to each other.
- Vertices 4 and 5 are connected to each other.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Degree Matrix

The degrees of the vertices in this configuration:

- Vertex 1 has 2 connections.
- Vertex 2 has 2 connections.
- Vertex 3 has 2 connections.
- Vertex 4 has 1 connection.
- Vertex 5 has 1 connection.

$$D = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Laplacian Matrix

$$L = D - A$$

$$L = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

To solve for the Eigenvectors of this Laplacian matrix we need to solve

$$Lv = 0$$

The Eigenvectors corresponding to zero eigenvalues (calculated using Python)

$$\begin{bmatrix} -0.57735027 \\ -0.57735027 \\ -0.57735027 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.70710678 \\ 0.70710678 \end{bmatrix}$$

- First Column: The first eigenvector $[-0.57735027, -0.57735027, -0.57735027, 0, 0]^T$ suggests that vertices 1, 2, and 3 are interconnected and form one component.
- Second Column: The second eigenvector $[0, 0, 0, 0.70710678, 0.70710678]^T$ indicates that vertices 4 and 5 are interconnected, forming another component.

For a general case if there are n columns in an eigenvector matrix of a Laplacian matrix, that means there are n distinct connected components.

Also as an extension, for each of the n columns, the indices of nonzero rows of each eigenvector represents the edges that are connected within their respective clusters.

SECTION 2 - MATH OF K-MEANS CLUSTERING

Question 1

m data points $x^i \in \mathbb{R}, i=1, \dots, m$

the Distortion Function is defined as

$$J = \sum_{i=1}^m \sum_{j=1}^k r^{ij} \|x^i - \mu^j\|^2$$

where r_{ij} is an indicator function such that

$$r_{ij} = \begin{cases} 1, & \text{if } x^i \text{ is in the } j^{\text{th}} \text{ cluster} \\ 0, & \text{otherwise} \end{cases}$$

and μ^j is the cluster centroid of cluster j

Expanding the dissimilarity function

$$\begin{aligned} \|x^i - \mu^j\|^2 &= (x^i - \mu^j)^T (x^i - \mu^j) \\ &= (x^i)^T x^i - (x^i)^T \mu^j - (\mu^j)^T x^i + (\mu^j)^T \mu^j \\ &= (x^i)^T x^i - 2(x^i)^T \mu^j + (\mu^j)^T \mu^j \end{aligned}$$

Note: Same step used in Section 1 Question 3

Once again, x^i is fixed as only μ^j are varied

Hence, $(x^i)^T x^i$ being a constant, we do not have to consider it when minimising J .

To minimise J is to minimise

$$\sum_{i=1}^m r^{ij} [-2(x^i)^T \mu^j + (\mu^j)^T \mu^j]$$

As we are taking derivative of J with respect to μ^j with r^{ij} fixed

$$\frac{\partial J}{\partial \mu^j} = \frac{\partial}{\partial \mu^j} \sum_{i=1}^m r^{ij} [-2(x^i)^T \mu^j + (\mu^j)^T \mu^j]$$

$$= \sum_{i=1}^m r^{ij} [-2x^i + 2\mu^j]$$

Minimisation occurs when $\frac{\partial J}{\partial \mu^j} = 0$ because the dissimilarity function is a convex function. In fact it will be a global minimum.

$$\sum_{i=1}^m r^{ij} [-2x^i + 2\mu^j] = 0$$

$$\sum_{i=1}^m r^{ij} (-2x^i) + \sum_{i=1}^m r^{ij} (2\mu^j) = 0$$

$$\sum_{i=1}^m 2r^{ij} \mu^j = \sum_{i=1}^m 2r^{ij} x^i$$

$$\mu^j = \frac{\sum_{i=1}^m 2r^{ij} x^i}{\sum_{i=1}^m 2r^{ij}}$$

$$\mu^j = \frac{\sum_{i=1}^m r^{ij} x^i}{\sum_{i=1}^m r^{ij}}$$

Question 2

From the previous question,

$$\mu^j = \frac{\sum_{i=1}^m r^{ij} x^i}{\sum_{i=1}^m r^{ij}}$$

But in this question the centroids μ^j are fixed

$$\text{Hence } \mu^j = \frac{\sum_{i=1}^m r^{ij} x^i}{\sum_{i=1}^m r^{ij}} \text{ is constant } \forall i \in [1, m]$$

I propose

$$r^{ij} = \begin{cases} 1, & \text{if } j = \arg \min_k \|x^i - \mu^k\|^2 \\ 0, & \text{otherwise} \end{cases}$$

Proof via Induction (m represents the number of data points)

Base Case (m=1)

Only one data point. So trivially that data point is its own cluster centroid. Therefore the assignment $r^{11} = 1$ trivially minimises J.

Inductive Step (prove $p \Rightarrow p+1$)

Assume that the optimal condition that minimises J, holds for $m = p$ where $p \geq 1$

This means that for p data points, the assignment r^{ij} minimises J when cluster centroids μ^j are fixed.

Let $r^{(p+1)j}$ be the assignment of x^{p+1} to cluster μ^j

By the proposed optimality condition

$$r^{(p+1)j} = \begin{cases} 1, & \text{if } j = \arg \min_k \|x^{p+1} - \mu^k\|^2 \\ 0, & \text{otherwise} \end{cases}$$

Consider two cases in the Inductive Step.

Case 1: $r^{(p+1)j} = 1$ for some j

This means that x^{p+1} is assigned to the closest centroid μ^j . Since $r^{pj} = 1$ is minimal, $r^{(p+1)j} = 1$ is also minimal. The assignment minimises the distortion function J for the $(p+1)^{th}$ data point.

Case 2: $r^{(p+1)j} = 0$ for all j

If $r^{(p+1)j} = 0$ for all clusters, it means that x^{p+1} is not assigned to any cluster. But x^{p+1} cannot form a new cluster either as the centroids have been fixed. This contradicts the premise of our assignment function, hence case 2 is not possible.

By Induction, we conclude that our proposed assignment variable r^{ij} will minimise the distortion function J , when the centroids μ^j are fixed.

Question 3

$$J = \sum_{i=1}^m \sum_{j=1}^k r^{ij} (x^i - \mu^j)^T \Sigma (x^i - \mu^j), \quad \Sigma \in \mathbb{R}^{n \times n}$$

Derivation of μ^j

By applying Chain rule,

$$\frac{\partial J}{\partial \mu^j} = \sum_{i=1}^m r^{ij} \Sigma (2)(x^i - \mu^j)(-1) = -2 \sum_{i=1}^m r^{ij} \Sigma (x^i - \mu^j)$$

Equating the derivative to zero, again due to the convexity of J ,

$$\frac{\partial J}{\partial \mu^j} = 0 \Rightarrow -2 \sum_{i=1}^m r^{ij} \Sigma (x^i - \mu^j) = 0$$

Solving for μ^j and factoring out -2,

$$\sum_{i=1}^m r^{ij} \Sigma x^i - \sum_{i=1}^m r^{ij} \Sigma \mu^j = 0$$

$$\sum_{i=1}^m r^{ij} \Sigma \mu^j = \sum_{i=1}^m r^{ij} \Sigma x^i$$

$$\mu^j = \frac{\sum_{i=1}^m r^{ij} \Sigma x^i}{\sum_{i=1}^m r^{ij} \Sigma}$$

Derivation of r^{ij}

$$r^{ij} = \begin{cases} 1, & \text{if } j = \arg \min_k (x^i - \mu^k)^T \Sigma (x^i - \mu^k) \\ 0, & \text{otherwise} \end{cases}$$

Can be shown using Induction like in the previous question.

SECTION 3 IMAGE COMPRESSION USING CLUSTERING

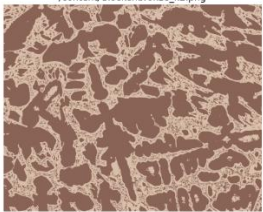


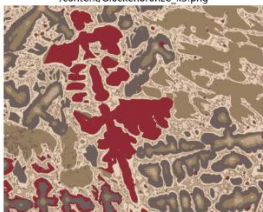


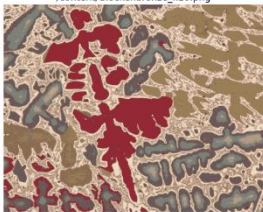


Question 1

In this question I report the Pseudocode of the Compression Algorithm and also the output images for each k-value. The full code is in the separate Jupyter Notebook. I used my GTID as the seed, for reproducibility.

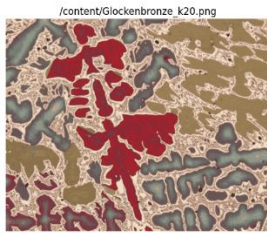
Pseudocode of Compression Algorithm

```
1  SET random seed
2
3  FUNCTION load_image(path):
4      READ and normalize image
5      RETURN pixels, shape
6
7  FUNCTION initialize_centroids(data, k):
8      RETURN k random centroids from data
9
10 FUNCTION assign_clusters(data, centroids):
11     CALCULATE distances, assign clusters
12     RETURN assignments
13
14 FUNCTION update_centroids(data, assignments, k):
15     RETURN mean of assigned points for each cluster
16
17 FUNCTION k_means(data, k, max_iters=100, tolerance=1e-4):
18     INITIALIZE centroids
19     FOR iteration in max_iters:
20         ASSIGN clusters, update centroids
21         IF centroids converged:
22             BREAK
23     RETURN assignments, centroids, iteration count
24
25 FUNCTION k_means_with_timing(data, k, max_iterations=100, tolerance=1e-4):
26     START timer
27     RUN k_means
28     STOP timer
29     RETURN assignments, centroids, iterations, time
30
31 FUNCTION reconstruction_error(pixels, centroids, assignments):
32     RECONSTRUCT image, calculate error
33     RETURN error
34
35 FUNCTION save_compressed_image(pixels, centroids, assignments, shape, path):
36     RECONSTRUCT, save, display image
37
38 SET image paths, k values
39 START total timer
40
41 FOR each image in paths:
42     LOAD image
43     INITIALIZE error, iteration lists
44
45     FOR each k in k values:
46         RUN k_means_with_timing
47         STORE error, iterations
48         SAVE compressed image
49         PRINT k-means results
50
51     PLOT elbow plot for image
52
53 STOP total timer
54 PRINT total elapsed time
```

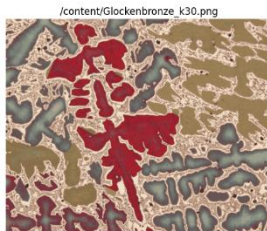
Image Outputs for Each k-value

K Value	Glockenbronze	Football	Avengers
K=2	 <p>/content/Glockenbronze_k2.png</p>	 <p>/content/football_k2.png</p>	 <p>/content/Avengers_k2.png</p>
K=5	 <p>/content/Glockenbronze_k5.png</p>	 <p>/content/football_k5.png</p>	 <p>/content/Avengers_k5.png</p>
K=10	 <p>/content/Glockenbronze_k10.png</p>	 <p>/content/football_k10.png</p>	 <p>/content/Avengers_k10.png</p>

K=20



K=30



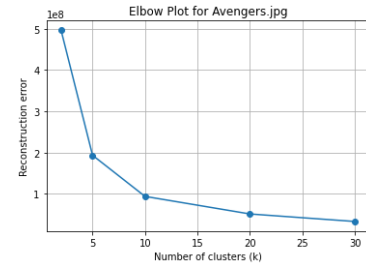
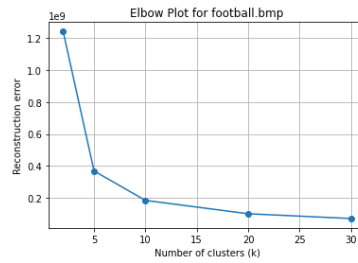
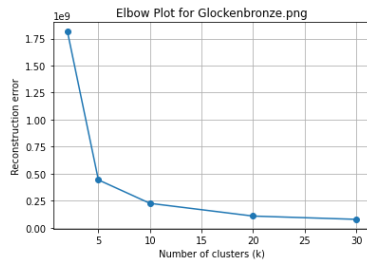
Question 2

		Time	Iterations	Reconstruction Error
Glockenbronze	K=2	5.50s	15	1814032702.28
	K=5	15.41s	31	442157631.65
	K=10	52.71s	100	225233885.38
	K=20	107.38s	100	107461951.33
	K=30	157.41s	100	77207353.48
Football	K=2	1.06s	23	1241175017.83
	K=5	3.12s	34	368469454.87
	K=10	14.31s	92	184187827.85
	K=20	27.58s	91	100432566.09
	K=30	47.19s	100	70731010.40
Avengers	K=2	0.12s	7	496551149.53
	K=5	2.16s	86	193507314.59
	K=10	2.46s	54	93900208.92
	K=20	7.84s	77	51344735.68
	K=30	15.14s	100	33156486.10

Total time taken to generate all images was 477.43s or 7.96 minutes.

Question 3

The method to finding the best k-value is by using an elbow plot.



Method

The best k value is determined using the elbow method. This method involves plotting the reconstruction error against different values of k (number of clusters) and identifying the point where the reduction in error starts to diminish significantly. This point, often referred to as the "elbow," indicates the optimal number of clusters. The elbow point balances the trade-off between minimizing the reconstruction error and not overly increasing the complexity of the model.

Analysis

For all three images, the elbow occurs when $k=5$, hence the best k value is found to be 5. This was determined by observing the elbow points in the reconstruction error plots, where the reduction in error becomes less significant beyond $k=5$. This indicates that using 5 clusters provides a good balance between error minimization and model simplicity.

SECTION 4 MNIST DATASET CLUSTERING

Questions 1 and 2

Like section 3, I used my GTID as the seed, for reproducibility.

Cluster	L2-Norm Purity Score	Manhattan Distance Purity Score
0	0.4226511071214841	0.5102136103653554
1	0.9397106109324759	0.9419340262087664
2	0.5572447447447447	0.7390857729840781
3	0.676358349990914	0.4819725864123957
4	0.9032536091359621	0.8008196721311476
5	0.8510204081632653	0.8631547969393761
6	0.4411556436765331	0.618554429263993
7	0.4041374214998153	0.4438938053097345
8	0.3418176787937474	0.3546345447121274
9	0.49987136609210187	0.3776815957847196

Summary of Results

- For clusters 0, 2, 6, and 7, the Manhattan distance metric provides higher purity scores than the L2 -norm.
- For clusters 1, 3, 4, and 5, the L2 -norm and Manhattan distance metrics provide very similar purity scores, with slight variations.
- For clusters 3, 4, and 9, the L2 -norm metric provides higher purity scores than the Manhattan distance.

Conclusion

The choice of the better metric can depend on the specific application and dataset.

In this analysis:

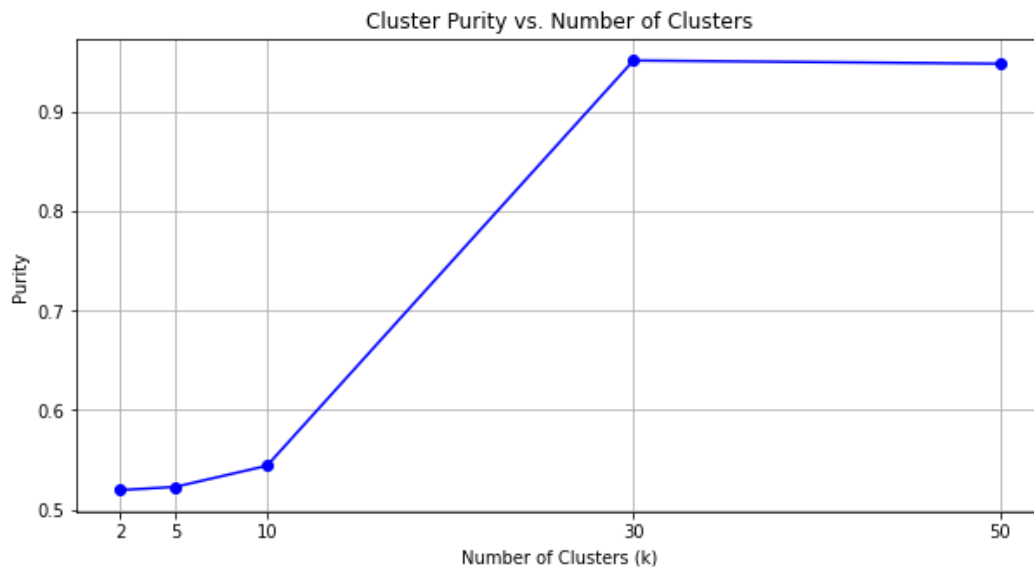
- The Manhattan distance generally provides better or comparable purity scores for most clusters compared to the L2 -norm.
- Specifically, the Manhattan distance shows clear advantages in clusters 0, 2, 6, and 7.

Based on the overall comparison, the Manhattan distance tends to give better clustering results in terms of purity scores for this MNIST dataset.

SECTION 5 POLITICAL BLOGS DATASET

The goal of this question is to verify the hypothesis whether similar blogs of similar political orientation cluster together. By finding the k-value that results in the lowest mismatch rate, we can infer how distinct the community structures are in terms of political orientation.

Question 1



Key Findings from Outputs:

1. Purity of Clusters:

- At $k=5$, the overall purity is relatively low at 52.29%, with significant variation among clusters. Individual cluster purities range from 51.90% to 100%.
- At $k=10$, overall purity slightly improves to 54.17%, with more clusters reaching maximum purity (100%).
- At $k=30$, there's a substantial increase in overall purity to 95.59%, demonstrating that finer clustering leads to more effective grouping.
- At $k=50$, overall purity remains high at 95.18%, with many clusters maintaining perfect purity.

2. Majority Labels:

- For all values of k , clusters with maximum purity (100%) show perfect homogeneity, indicating effective grouping at these cluster sizes.

3. Mismatch Rate:

- For lower values of k (5 and 10), there are still some clusters with lower purities, resulting in higher mismatch rates.
- At higher k values (30 and 50), the mismatch rate effectively decreases as most clusters approach perfect purity.

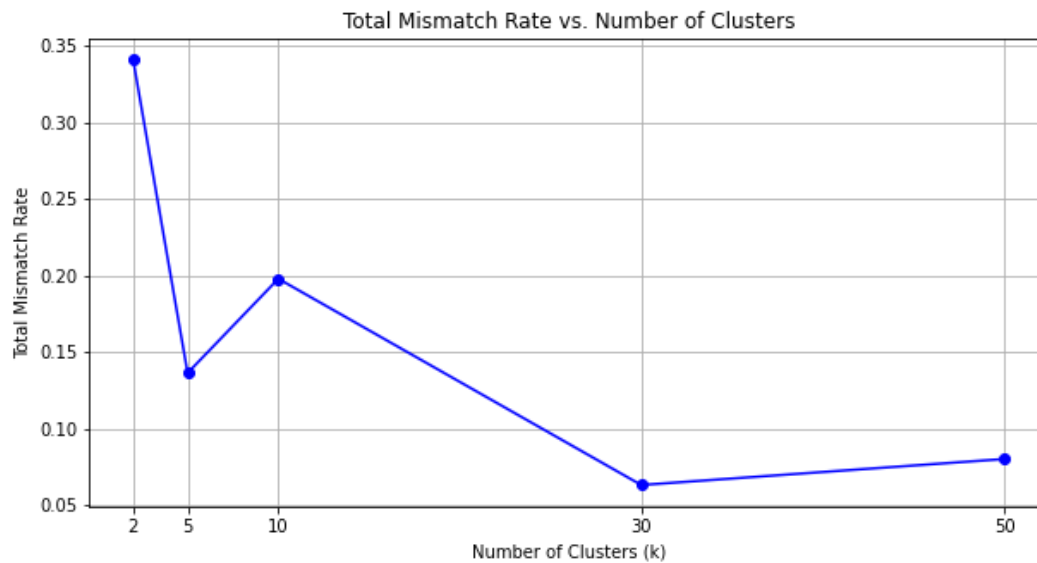
Plot Analysis:

- The plot illustrates a rapid increase in purity, notably beyond $k=10$, which stabilizes above 90% at $k=30$ and remains high at $k=50$. This trend suggests increased effectiveness of the clustering algorithm with more clusters.

Conclusions:

- Effectiveness of Spectral Clustering: Your data supports the hypothesis that blogs with similar political orientations effectively cluster together using spectral clustering, particularly at higher values of k .
- Optimal Number of Clusters: $k=30$ seems optimal for achieving high purity while maintaining manageability, although $k=50$ also provides high purity with more granularity.
- Recommendations: Given the high purity achieved at $k=30$ and $k=50$, it's beneficial to consider these settings for detailed analysis. Further, it might be worth exploring different initializations or parameters in the spectral clustering to see if the results can be optimized even further.

Question 2



Summary of Mismatch Rates and Tuning k

1. Trend in Mismatch Rates:

- The mismatch rate generally decreases as the number of clusters (k) increases.
- At k=2, the total mismatch rate is relatively high at 48.04%.
- At k=5, the rate slightly decreases to 47.71%.
- A significant drop is observed by k=10, where the rate further decreases to 45.83%.
- The lowest rates are observed at k=30 (4.41%) and k=50 (4.58%), indicating a substantial improvement in clustering accuracy with higher k values.

2. Optimal k Value:

- While the lowest mismatch rates are achieved at k=30 and k=50, choosing an optimal k should consider both precision and practicality. Very high k values may lead to overfitting or cumbersome management of too many small clusters.

- $k=30$ appears to offer a practical balance, achieving a very low mismatch rate while avoiding the complexities associated with managing too many clusters.

Implications for Network Community Structure

- Community Complexity: The reduction in mismatch rates as k increases suggests the presence of multiple distinct sub-communities within the blog network, indicating a complex structure where blogs are tightly knit based on similar political orientations.

- Practical Application: For practical applications, selecting a k around 30 might be optimal. This balances achieving low mismatch rates and maintaining manageable cluster sizes without getting too granular, which is important for practical data interpretation and usability.