Understanding Total Variation Within Each Cluster

Total variation within a cluster is a measure of how spread out the data points are from the cluster's centroid. In other words, it quantifies the **variance** or **dispersion** of the data points in each cluster. To calculate total variation, we typically calculate the **sum of squared distances** between each data point and the centroid of its assigned cluster.

The formula for the total variation (within-cluster sum of squares, WCSS) for a cluster is:

$$WCSS = \sum_{i=1}^{n} (d_i - C)^2$$

Where:

- d_i is the i-th data point in the cluster.
- C is the centroid of the cluster.
- n is the number of data points in the cluster.

To calculate the total variation across all clusters, you sum the WCSS for each cluster:

$$\text{Total Variation} = \sum_{k=1}^{K} WCSS_k$$

Where K is the total number of clusters.

Steps to Calculate and Visualize Total Variation (WCSS)

Let's use the same example dataset to calculate and visualize the total variation for each cluster.

Step 1: Calculate WCSS for Each Cluster

Using the previous cluster assignments, we have:

- **Cluster 1** ($C_1 = (1.5, 1)$) with points: $(1, 1)$, $(2, 1)$
- **Cluster 2** ($C_2 = (4.5, 3.5)$) with points: $(4, 3)$, $(5, 4)$

Cluster 1:

1. Point $(1, 1)$:
   - Distance from centroid $C_1 = (1.5, 1)$:

$$\text{Distance} = \sqrt{(1 - 1.5)^2 + (1 - 1)^2} = 0.5$$

   - Squared distance: $0.5^2 = 0.25$
2. Point $(2, 1)$:
   - Distance from centroid $C_1 = (1.5, 1)$:

$$\text{Distance} = \sqrt{(2-1.5)^2 + (1-1)^2} = 0.5$$

- Squared distance: $0.5^2 = 0.25$

WCSS for Cluster 1:

$$WCSS_1 = 0.25 + 0.25 = 0.5$$

Cluster 2:

1. Point $(4, 3)$:
   - Distance from centroid $C_2 = (4.5, 3.5)$:

$$\text{Distance} = \sqrt{(4-4.5)^2 + (3-3.5)^2} = 0.707$$

   - Squared distance: $0.707^2 = 0.5$
2. Point $(5, 4)$:
   - Distance from centroid $C_2 = (4.5, 3.5)$:

$$\text{Distance} = \sqrt{(5-4.5)^2 + (4-3.5)^2} = 0.707$$

   - Squared distance: $0.707^2 = 0.5$

WCSS for Cluster 2:

$$WCSS_2 = 0.5 + 0.5 = 1.0$$

Step 2: Calculate Total Variation (WCSS across all clusters)

Now, let's compute the total variation:

$$\text{Total Variation} = WCSS_1 + WCSS_2 = 0.5 + 1.0 = 1.5$$

Step 3: Check if Variation is Minimum (Convergence)

- The total variation will **reduce** as the centroids converge towards the optimal positions.
- If the variation doesn't change after a few iterations, it indicates that the clustering process has converged, and further recalculations won't improve the results.

To determine whether the variation is zero or minimized:

- **Zero variation** would imply that all points in a cluster are **identical**, which is extremely rare.
- The **minimum variation** means the points are as close as possible to the centroid, minimizing the distance within each cluster.

Step 4: Visualize the Variation of Each Cluster

You can visualize the variation within each cluster using a scatter plot and by plotting the **sum of squared distances**.

python Copy code

```python
import numpy as np import matplotlib.pyplot as plt # Define the points and centroids
cluster1_points = np.array([[1, 1], [2, 1]]) cluster2_points = np.array([[4, 3], [5, 4]]) C1 =
np.array([1.5, 1]) C2 = np.array([4.5, 3.5]) # Calculate the squared distances for each cluster
wcss_1 = np.sum(np.square(np.linalg.norm(cluster1_points - C1, axis=1))) wcss_2 =
np.sum(np.square(np.linalg.norm(cluster2_points - C2, axis=1))) # Plot the points and centroids
plt.figure(figsize=(8, 6)) plt.scatter(cluster1_points[:, 0], cluster1_points[:, 1], color='blue',
label='Cluster 1') plt.scatter(cluster2_points[:, 0], cluster2_points[:, 1], color='green',
label='Cluster 2') plt.scatter(C1[0], C1[1], color='red', marker='x', label='Centroid 1')
plt.scatter(C2[0], C2[1], color='orange', marker='x', label='Centroid 2') # Annotate the plot
plt.text(C1[0], C1[1], f'C1 (WCSS: {wcss_1:.2f})', fontsize=12, color='red') plt.text(C2[0], C2[1],
f'C2 (WCSS: {wcss_2:.2f})', fontsize=12, color='orange') # Labels and title plt.xlabel('X-axis')
plt.ylabel('Y-axis') plt.title('Cluster Visualization with Centroids') plt.legend() plt.show()
```

Visualizing WCSS:

To visualize the **within-cluster sum of squares (WCSS)**, you can plot the WCSS values for each cluster as a bar graph:

python Copy code

```python
# Plot WCSS clusters = ['Cluster 1', 'Cluster 2'] wcss_values = [wcss_1, wcss_2] plt.bar(clusters,
wcss_values, color=['blue', 'green']) plt.title('WCSS for Each Cluster') plt.xlabel('Clusters')
plt.ylabel('WCSS') plt.show()
```

This will give you a clear visual representation of how much variation exists within each cluster, with lower WCSS indicating tighter, more compact clusters.

Conclusion

- **Zero variation**: Would only occur if all points within a cluster are the same (rare).
- **Minimum variation**: When the total variation doesn't change significantly between iterations, and the clusters are well-separated.
- **WCSS**: A key indicator to track the tightness of your clusters during the K-means algorithm, and it can be visualized for better understanding of cluster dispersion.