**Chelsea Murray – 113481968 – Homework 4 – Data Mining Fall 2024**

1. The chat transcript can be found here: <https://chatgpt.com/share/6729a718-d024-800f-a2d2-42bd08fe6cf0>

|  |  |
| --- | --- |
| User: | Assume I am looking for information at the intelligence level of a professional in the field of data mining. Tell me about three real-world companies/organizations from three distinct sectors—software technology, telecommunications, and transportation—that have implemented data clustering algorithms in their businesses. Include detailed descriptions of how those companies/organizations have applied data clustering algorithms in their operations. |
| ChatGPT: | Certainly! Here’s an overview of how companies in software technology, telecommunications, and transportation apply data clustering algorithms to enhance their operations.  **1. Software Technology: Google**  **Application of Clustering:** Google uses data clustering extensively across its services, particularly in organizing search results, personalizing recommendations, and improving user experience. One significant area is **image recognition and search**. Google's algorithms cluster images with similar visual content, allowing the company to categorize vast amounts of images, videos, and other visual data. This clustering supports Google Photos in organizing user photos by people, places, or activities.  **How It Works:** By applying clustering algorithms (such as k-means and hierarchical clustering), Google groups similar visual data based on pixel patterns, color distributions, or shape similarities. The clustered data helps the search algorithm quickly identify relevant images for users' search queries, improving the speed and accuracy of visual search results. Moreover, clustering aids in refining Google’s recommendation systems for YouTube and Play Store by grouping users based on shared preferences, enabling the platform to suggest more personalized content.  **2. Telecommunications: AT&T**  **Application of Clustering:** AT&T uses data clustering to optimize its network performance and improve customer experience. Clustering algorithms help AT&T analyze patterns in user data across different geographic regions and identify areas experiencing high demand or recurring issues. By grouping these regions based on customer usage patterns and signal strength, AT&T can allocate network resources more effectively.  **How It Works:** Clustering algorithms such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and k-means are employed to analyze location-based data, call drop frequencies, and network traffic. These clusters reveal zones of high usage or network congestion, helping AT&T to prioritize infrastructure upgrades or reconfigure network settings dynamically. For instance, AT&T might detect clusters of high data traffic in metropolitan areas during peak hours, prompting a temporary reallocation of bandwidth to prevent network slowdowns.  **3. Transportation: Uber**  **Application of Clustering:** Uber applies clustering algorithms for **demand forecasting** and **dynamic pricing**. By clustering locations based on demand patterns, Uber can identify “hot zones” where ride requests are likely to spike, especially during events, rush hours, or holidays. Clustering is also used to optimize driver availability in high-demand areas.  **How It Works:** Uber leverages k-means clustering and density-based clustering to group similar trip data by factors like pickup locations, drop-off points, and times of high demand. This data helps Uber forecast demand patterns in specific clusters, adjusting pricing through “surge pricing” when demand in these areas exceeds driver availability. Clustering also assists Uber in mapping out ideal “standby” zones for drivers to minimize wait times and optimize routes, enhancing both driver efficiency and customer satisfaction. |

[response once u are not sleep deprived]

1. Given the table:



2.1) Here is the application of bisecting K-means to cluster the data using K=4, number of trials=2, and choosing the cluster with the largest SSE for splitting at each step.

**Cluster Step 1: Initialize cluster containing all points.**

The first cluster contains every point in the table.

|  |  |
| --- | --- |
| **Cluster** | **Contents** |
| A | {(4.38, -15.96), (0.69, 2.5), (4.69, 36.75), (4.7, -7.83), (4.38, 34.43), (0.47, -2.5), (4.07, 33.8), (5.01, -16.59)} |

**Cluster Step 2: Split into two on the “all points” cluster and find the bisection with the smallest SSE.**

At the second step, the cluster is bisected twice (according to the number of trials specified). The first trial will sort the first half of cluster A into one bisection and the second half into another bisection (ex. if A contains {1, 2, 3, 4}, then its bisected clusters would contain {1, 2} and {3, 4}); the second trial will alternate adding points from cluster A to one cluster vs. the other (ex. if A contains {1, 2, 3, 4}, then its bisected clusters would contain {1, 3} and {2, 4}).

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Contents** | **SSE** |
| AA | {(4.38, -15.96), (0.69, 2.5), (4.69, 36.75), (4.7, -7.83)} | 1624.5636 |
| AB | {(4.38, 34.43), (0.47, -2.5), (4.07, 33.8), (5.01, -16.59)} | 2018.2172 |
| Total SSE | | 3642.7808 |
|  |  |  |
| AC | {(4.38, -15.96), (4.69, 36.75), (4.38, 34.43), (4.07, 33.8)} | 1952.201 |
| AD | {(0.69, 2.5), (4.7, -7.83), (0.47, -2.5), (5.01, -16.59)} | 214.6486 |
| Total SSE | | 2166.8496 |

Here are the SSE calculations for each cluster:

AA: Midpoint/Centroid = (mean(x), mean(y)) = (3.615, 3.865)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (4.38, -15.96) | (3.615-4.38)2+(3.865-(-15.96))2 = 0.5852+393.0306 = 393.6158 | 393.6158 |
| (0.69, 2.5) | (3.615-0.69)2+(3.865-2.5)2 = 8.5556+1.8632 = 10.4188 | 393.6158+10.4188 = 404.0346 |
| (4.69, 36.75) | (3.615-4.69)2+(3.865-36.75)2 = 1.1556+1081.4232 = 1082.5788 | 404.0346+1082.5788 = 1486.6134 |
| (4.7, -7.83) | (3.615-4.7)2+(3.865-(-7.83))2 = 1.1772+136.7730 = 137.9502 | 1486.6134+137.9502 = 1624.5636 |

AB: Midpoint/Centroid = (mean(x), mean(y)) = (3.4825, 12.285)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (4.38, 34.43) | (3.4825-4.38)2+(12.285-34.43)2 = 0.8055+490.4010 = 491.2065 | 491.2065 |
| (0.47, -2.5) | (3.4825-0.47)2+(12.285-(-2.5))2 = 9.0752+218.5962 = 227.6714 | 491.2065+227.6714 = 718.8779 |
| (4.07, 33.8) | (3.4825-4.07)2+(12.285-33.8)2 = 0.3452+462.8952 = 463.2404 | 718.8779+463.2404 = 1182.1183 |
| (5.01, -16.59) | (3.4825-5.01)2+(12.285-(-16.59))2 = 2.3333+833.7656 = 836.0989 | 1182.1183+836.0989 = 2018.2172 |

AC: Midpoint/Centroid = (mean(x), mean(y)) = (4.38, 22.255)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (4.38, -15.96) | (4.38-4.38)2+(22.255-(-15.96))2 = 0+1460.3862 = 1460.3862 | 1460.3862 |
| (4.69, 36.75) | (4.38-4.69)2+(22.255-36.75)2 = 0.0961+210.1050 = 210.2011 | 1460.3862+210.2011 = 1670.5873 |
| (4.38, 34.43) | (4.38-4.38)2+(22.255-34.43)2 = 0+148.2306 = 148.2306 | 1670.5873+148.2306 = 1818.8179 |
| (4.07, 33.8) | (4.38-4.07)2+(22.255-33.8)2 = 0.0961+133.2870 = 133.3831 | 1818.8179+133.3831 = 1952.201 |

AD: Midpoint/Centroid = (mean(x), mean(y)) = (2.7175, -6.105)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (0.69, 2.5) | (2.7175-4.38)2+(-6.105-2.5)2 = 2.7639+74.0460 = 76.8099 | 76.8099 |
| (4.7, -7.83) | (2.7175-0.69)2+(-6.105-(-7.83))2 = 4.1108+2.9756 = 7.0864 | 76.8099+7.0864 = 83.8963 |
| (0.47, -2.5) | (2.7175-4.69)2+(-6.105-(-2.5))2 = 3.8908+12.9960 = 16.8868 | 83.8963+16.8868 = 100.7831 |
| (5.01, -16.59) | (2.7175-4.7)2+(-6.105-(-16.59))2 = 3.9303+109.9352 = 113.8655 | 100.7831+113.8655 = 214.6486 |

Because the total SSE for the bisection of A into AC and AD is lower than that of AA and AB, this is the bisection that will be “chosen” for this iteration.

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Contents** | **SSE** |
| AC | {(4.38, -15.96), (4.69, 36.75), (4.38, 34.43), (4.07, 33.8)} | 1952.201 |
| AD | {(0.69, 2.5), (4.7, -7.83), (0.47, -2.5), (5.01, -16.59)} | 214.6486 |

AC has the higher SSE of the two clusters, so it will be bisected next.

**Cluster Step 3: Split into two on the highest-SSE cluster.**

AD remains untouched since its SSE was lower. Bisecting AC with two trials in the same fashion as the first bisection results in the following clusters:

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Contents** | **SSE** |
| AD | {(0.69, 2.5), (4.7, -7.83), (0.47, -2.5), (5.01, -16.59)} | 214.6486 |
|  | | |
| ACA | {(4.38, -15.96), (4.69, 36.75)} | 1389.2200 |
| ACB | {(4.38, 34.43), (4.07, 33.8)} | 0.2464 |
| Total SSE | | 1389.4664 |
|  | | |
| ACC | {(4.38, -15.96), (4.38, 34.43)} | 1269.576 |
| ACD | {(4.69, 36.75), (4.07, 33.8)} | 4.5434 |
| Total SSE | | 1274.1194 |

ACA: Midpoint/Centroid = (mean(x), mean(y)) = (4.535, 10.395)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (4.38, -15.96) | (4.535-4.38)2+(10.395-(-15.96))2 = 0.0240+694.5860 = 694.6100 | 694.6100 |
| (4.69, 36.75) | (4.535-4.69)2+(10.395-36.75)2 = 0.0240+694.5860 = 694.6100 | 694.6100+694.6100 = 1389.2200 |

ACB: Midpoint/Centroid = (mean(x), mean(y)) = (4.225, 34.115)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (4.38, 34.43) | (4.225-4.38)2+(34.115-34.43)2 = 0.0240+0.0992 = 0.1232 | 0.1232 |
| (4.07, 33.8) | (4.225-4.07)2+(34.115-33.8)2 = 0.0240+0.0992 = 0.1232 | 0.1232+0.1232 = 0.2464 |

ACC: Midpoint/Centroid = (mean(x), mean(y)) = (4.38, 9.235)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (4.38, -15.96) | (4.38-4.38)2+(9.235-(-15.96))2 = 0+634.7880 = 634.7880 | 634.7880 |
| (4.38, 34.43) | (4.38-4.38)2+(9.235-34.43)2 = 0+634.7880 = 634.7880 | 634.7880+634.7880 = 1269.576 |

ACD: Midpoint/Centroid = (mean(x), mean(y)) = (4.38, 35.275)

|  |  |  |
| --- | --- | --- |
| **Point** | **Eucl. Dist. Squared** | **Running Sum** |
| (4.69, 36.75) | (4.38-4.69)2+(35.275-36.75)2 = 0.0961+2.1756 = 2.2717 | 2.2717 |
| (4.07, 33.8) | (4.38-4.07)2+(35.275-33.8)2 = 0.0961+2.1756 = 2.2717 | 2.2717+2.2717 = 4.5434 |

The ACC/ACD set has the lowest SSE, meaning those will be the clusters retained from this set of trials.

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Contents** | **SSE** |
| AD | {(0.69, 2.5), (4.7, -7.83), (0.47, -2.5), (5.01, -16.59)} | 214.6486 |
| ACC | {(4.38, -15.96), (4.38, 34.43)} | 1269.576 |
| ACD | {(4.69, 36.75), (4.07, 33.8)} | 4.5434 |

The highest SSE is cluster ACC, so that will be bisected next.

**Cluster Step 4: Split into two on the highest-SSE cluster.**

There is only one way to bisect a cluster of 2, so the cluster set is as follows:

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Contents** | **SSE** |
| AD | {(0.69, 2.5), (4.7, -7.83), (0.47, -2.5), (5.01, -16.59)} | 214.6486 |
| ACD | {(4.69, 36.75), (4.07, 33.8)} | 4.5434 |
| ACCA | {(4.38, -15.96)} | 0 |
| ACCB | {(4.38, 34.43)} | 0 |
| Total SSE | | 219.192 |

With K=4 clusters, the algorithm is complete and this is the final clustering and total SSE of each cluster.  
  
Here is the resulting graph (generated with Desmos):

The bisection methods may have been more effective if the spread of y-values were taken into consideration prior to attempting to split the data, as it appears the grouping of dissimilar y-values may have contributed to less-accurate clustering.

