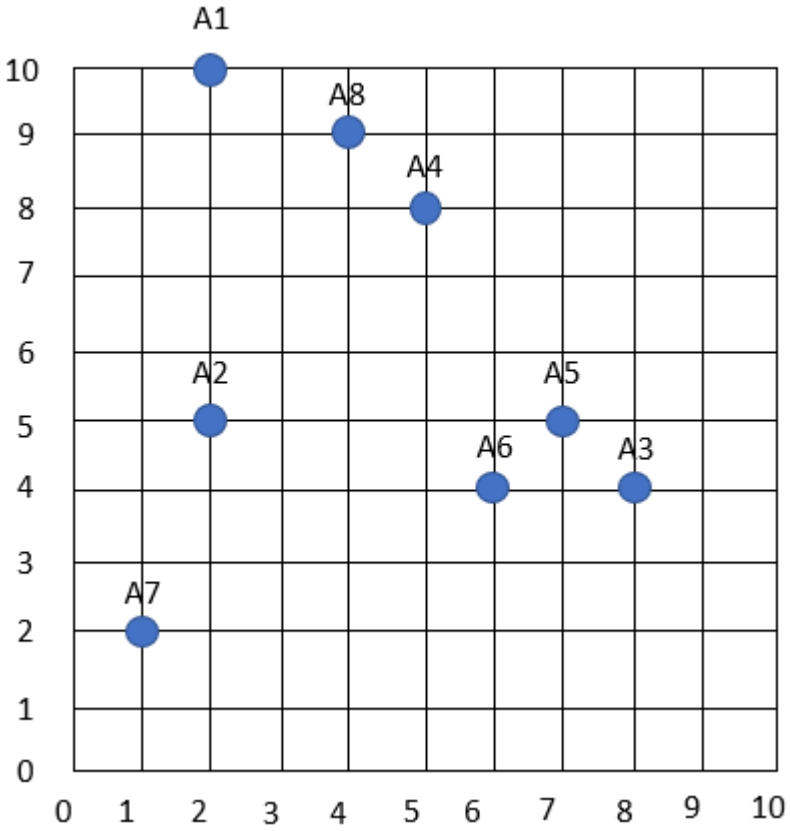
1. K-Means algorithm:

Consider the following eight points in a 2-dimensional space: {(2, 10); (2, 5); (8, 4); (5, 8); (7, 5); (6, 4); (1, 2); (4, 9)}. Use the Euclidean distance metric to measure the separation of the points.

a. Plot the data points and group them into appropriate clusters. How many clusters are required and what are contents of each of these clusters?



3 clusters are required. Cluster 1: (1.5, 3.5); Cluster 2: (7, 4.3); Cluster 3: (3.6, 9).

b. Now consider we want to divide the points into 3 initial clusters (C1, C2, C3) with centers defined as {(2, 5), (5, 8), (4, 9)} respectively.

c. What's the center of the cluster after one iteration?
   Cluster 1: (3, 3.6)
   Cluster 2: (6.6, 5.6)
   Cluster 3: (3, 9.5)

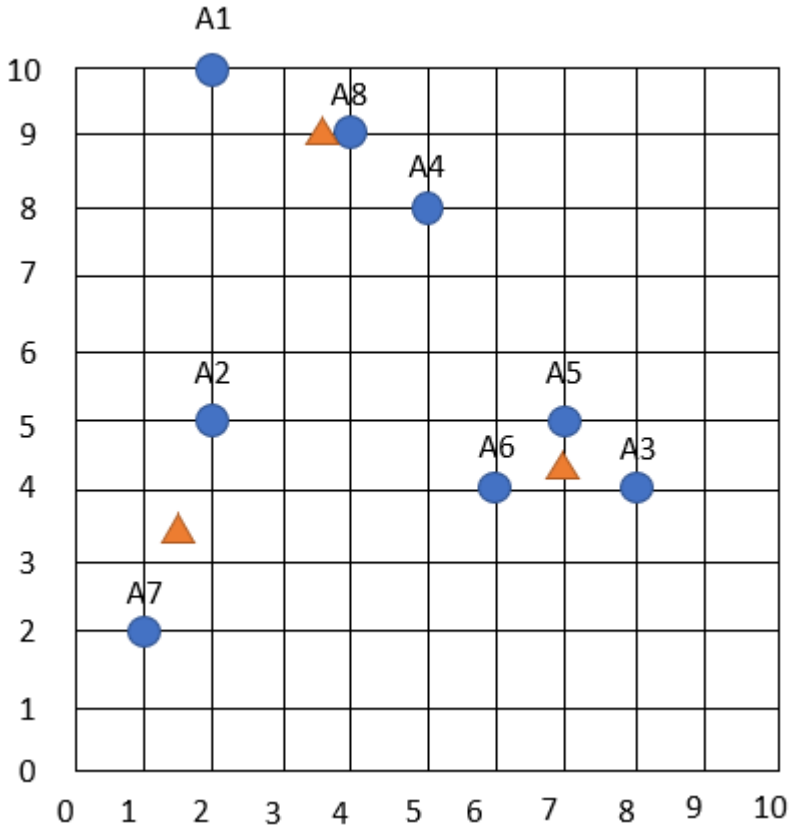
d. What's the center of the cluster after one 2nd iteration?
   Cluster 1: (1.5, 3.5)
   Cluster 2: (7, 4.3)
   Cluster 3: (3.6, 9)

e. What's the center of the cluster after one 3rd iteration?
   Cluster 1: (1.5, 3.5)
   Cluster 2: (7, 4.3)
   Cluster 3: (3.6, 9)

f. Compare the results of each iteration with your answers in part (a).

g. How many iterations are required for the clusters to converge?
   2

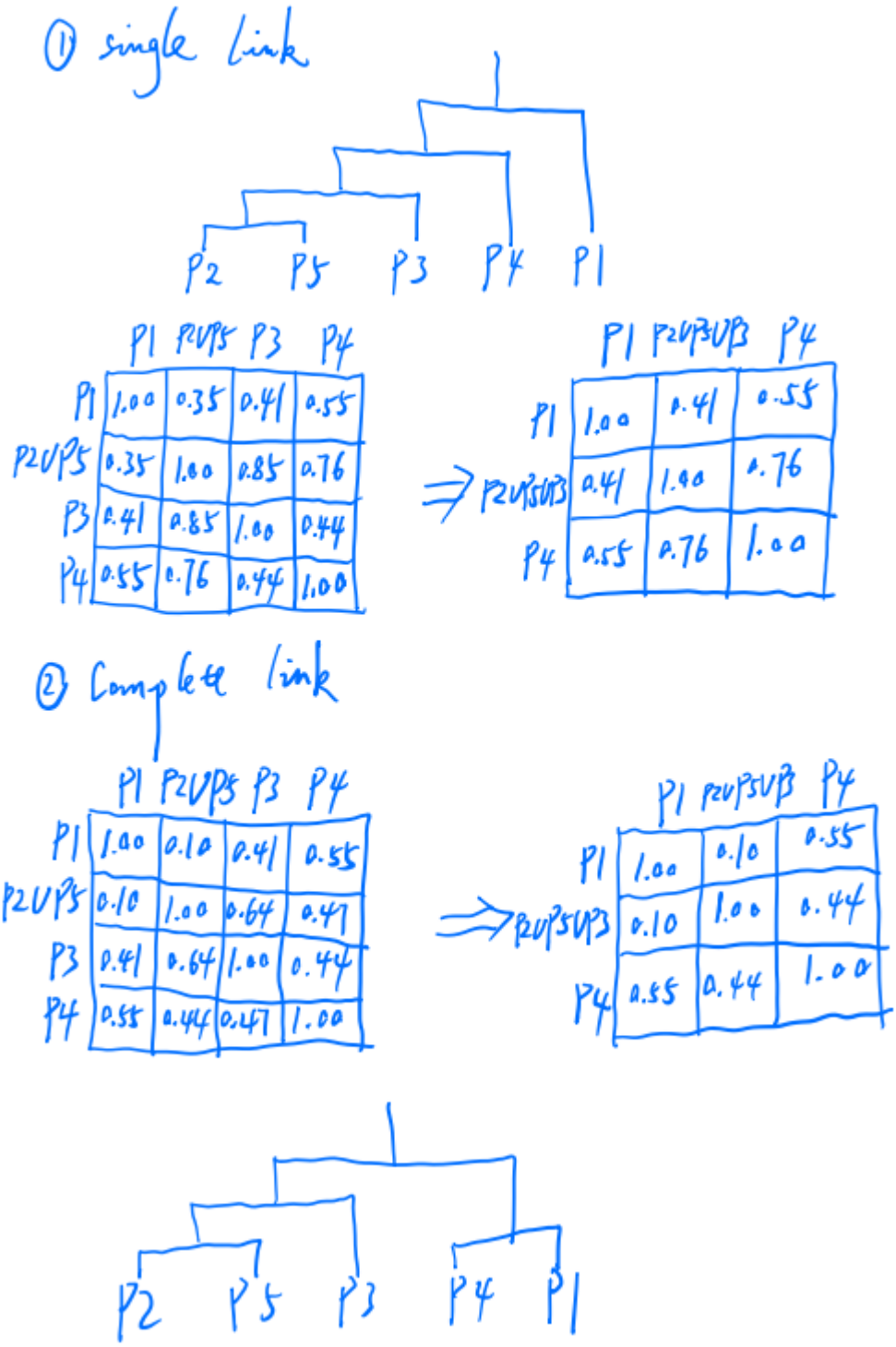
h. What are the resulting centers and resulting clusters (K=3)? Plot the final data points.
Cluster 1: (1.5, 3.5)
Cluster 2: (7, 4.3)
Cluster 3: (3.6, 9)



2. Hierarchical algorithm:

Use the similarity matrix in Table 1 to perform single and complete link hierarchical clustering. Show your results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged. (with enough explanation and calculation)

	P1	P2	P3	P4	P5
P1	1.00	0.10	0.41	0.55	0.35
P2	0.10	1.00	0.64	0.47	0.98
P3	0.41	0.64	1.00	0.44	0.85
P4	0.55	0.47	0.44	1.00	0.76
P5	0.35	0.98	0.85	0.76	1.00



3. DBSCAN algorithm

Consider the following eight point in a 2-dimensional space: {(2, 10); (2, 5); (8, 4); (5, 8); (7, 5); (6, 4); (1, 2); (4, 9)}. Suppose we use the Euclidean distance metric.

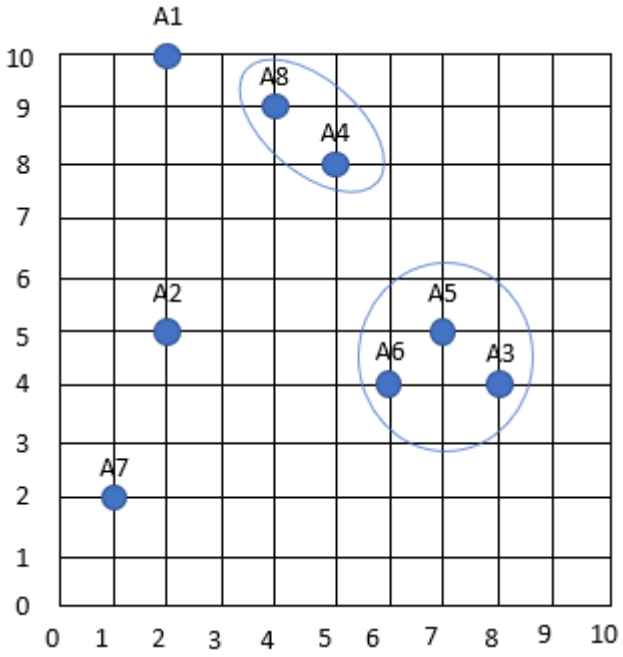
a. If Epsilon is 2 and min_samples is 2, what are the clusters that DBSCAN would discover. Plot the discovered clusters.

N_2(A1)={}; N_2(A2)={}; N_2(A3)={A5, A6}; N_2(A4)={A8}; N_2(A5)={A3, A6};

$N_2(A6)=\{A3, A5\}$; $N_2(A7)=\{\}$; $N_2(A8)=\{A4\}$.

Thus, A1, A2, and A7 are outliers, we have two clusters C1={A4, A8} and C2={A3, A5, A6}.

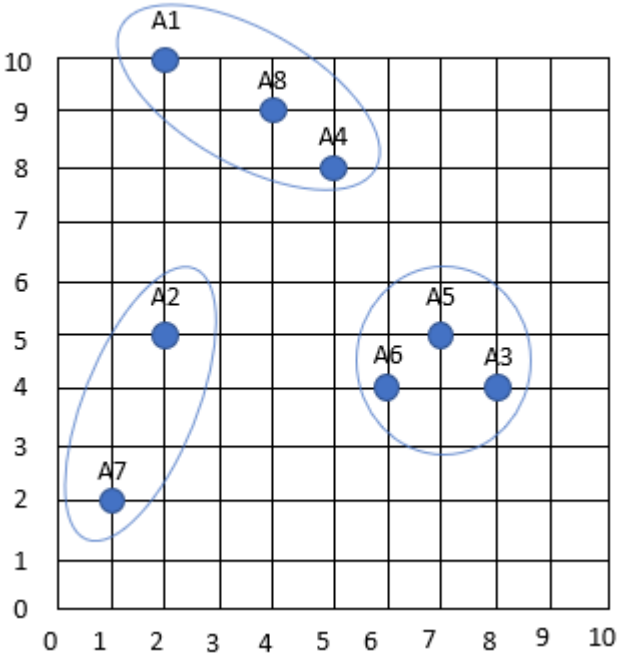
Plot:



b. What if Epsilon is increased to $\sqrt{10}$?

If Epsilon is creased to $\sqrt{10}$, then the neighborhood of some points will increase: A1 would join the cluster C1 and A2 would join with A7 to form cluster C3={A2, A7}.

Plot:



4. Explain the shortcomings of BFR algorithm and describe how CURE algorithm overcomes the shortcomings.

Answer:
The shortcomings of BFR algorithm are: (1) it makes a very strong assumption about the shape of clusters: they must be normally distributed about a centroid. (2) The mean and standard deviation for a cluster may differ for different dimensions, but the dimensions must be independent.
How CURE algorithm overcomes the shortcomings:
To avoid the problems with non-uniform sized or shaped clusters, CURE employs a hierarchical clustering algorithm that adopts a middle ground between the centroid based and all point extremes.

Part #2: BigTable and Cassandra

Q1. Compare BigTable with Cassandra.

Bigtable and Cassandra are distributed databases. They implement multidimensional key-value stores that can support tens of thousands of queries per second (QPS), storage that scales up to petabytes of data, and tolerance for node failure.

While the feature sets of these databases are similar at a high level, their underlying architectures and interaction details differ in ways that are important to understand. This document highlights the similarities and differences between the two database systems.

We can compare them in terminology, in data modeling handling, in writing and reading path taken, in the physical data layout to understand aspects of the database architecture.

Although many of the concepts used in Bigtable and Cassandra are similar, each database has slightly different naming conventions and subtle differences.

One of the core building blocks of both databases is the sorted string table (SSTable). In both architectures, SSTables are created to persist data that's used to respond to read queries. An SSTable is a simple abstraction to efficiently store large numbers of key-value pairs while optimizing for high throughput, sequential read/write workloads."

the design philosophy and key attributes of Bigtable and Cassandra.

Bigtable

Bigtable provides many of the core features described in the Bigtable: A Distributed Storage System for Structured Data paper. Bigtable separates the compute nodes, which serve client requests, from the underlying storage management. Data is stored on Colossus. The storage layer automatically replicates the data to provide durability that exceeds levels provided by standard Hadoop Distributed File System (HDFS) three-way replication.

This architecture provides consistent reads and writes within a cluster, scales up and down without any storage redistribution cost, and can rebalance workloads without modifying the cluster or schema. If any data processing node becomes impaired, the Bigtable service replaces

it transparently. Bigtable also supports asynchronous replication between geographically distributed clusters in topologies of up to four clusters in different Google Cloud zones or regions throughout the world.

In addition to gRPC and client libraries for various programming languages, Bigtable maintains compatibility with the open source Apache HBase Java client library, an alternative open source database engine implementation of the Bigtable paper.

Developers describe Cassandra as "A partitioned row store. Rows are organized into tables with a required primary key". Partitioning means that Cassandra can distribute your data across multiple machines in an application-transparent matter. Cassandra will automatically repartition as machines are added and removed from the cluster. Row store means that like relational databases, Cassandra organizes data by rows and columns. The Cassandra Query Language (CQL) is a close relative of SQL. On the other hand, Google Cloud Bigtable is detailed as "The same database that powers Google Search, Gmail and Analytics". Google Cloud Bigtable offers you a fast, fully managed, massively scalable NoSQL database service that's ideal for web, mobile, and Internet of Things applications requiring terabytes to petabytes of data. Unlike comparable market offerings, Cloud Bigtable doesn't require you to sacrifice speed, scale, or cost efficiency when your applications grow. Cloud Bigtable has been battle-tested at Google for more than 10 years—it's the database driving major applications such as Google Analytics and Gmail.

Cassandra belongs to "Databases" category of the tech stack, while Google Cloud Bigtable can be primarily classified under "NoSQL Database as a Service".

"Distributed" is the primary reason why developers consider Cassandra over the competitors, whereas "High performance" was stated as the key factor in picking Google Cloud Bigtable.

Q3. Explain the concept of tunable consistency in Cassandra.

To address this problem, Cassandra maintains tunable consistency. When performing a read or write operation a database client can specify a consistency level. The consistency level refers to the number of replicas that need to respond for a read or write operation to be considered complete.

For reading non-critical data (the number of “likes” on a social media post, for example), it’s probably not essential to have the very latest data. You can set the consistency level to ONE and Cassandra will simply retrieve a value from the closest replica. If I’m concerned about accuracy, however, I can specify a higher consistency level, like TWO, THREE, or QUORUM. If a QUORUM (essentially a majority) of replicas reply, and if the data was written with similarly strong consistency, users can be confident that they have the latest data. If there are inconsistencies between replicas when data is read, Cassandra will internally manage a process to ensure that replicas are synchronized and contain the most recent data.

The same process applies to write operations. Specifying a higher consistency level forces multiple replicas to be written before a write operation can complete. For example, if “ALL” or “THREE” are specified when updating a table with three replicas, data will need to be updated to all replicas before a write can complete.

There is a trade-off between consistency and availability here, as well. If one of the replicas is down or unreachable, the write operation will fail since Cassandra cannot meet the required consistency level. In this case, Cassandra sacrifices availability to guarantee consistency.

Q4. Define memtable.
Bigtable servers store recent updated in memory in a data structure known as memtable.
Reading from memtable is quick and if the server crashes, memtable can be constructed again using committed log.

Q5. What is SSTable? How is it different from other relational tables?
SSTable is sorted string table, used to store table data in google file system. It is different from relational database as relational database do not provide partition tolerance.

Q6. Explain CAP theorem.
For any system that share data is impossible to guarantee simultaneously the following properties:

a. Consistency
b. Availability
c. Partition tolerance

Very large scale system with partition at some pour

a. It is essential to decide between consistency and availability
b. Traditional DBMS prefer consistency over availability and partition tolerance
c. More web application put more emphases on availability

Q7. Describe difference between Tablet Server and Tablets.
A tablet is a set of consecutive rows of a table and is a pair of distribution and load balancing within BigTable
A tablet server stored and served tablets to clients. For a given tablet, a tablet server acts as a leader and other servers follow, replicates of the tablet.