Q1. Present the clustering of Example 7.2 if we use the minimum of the distances between any two points as the distance between two clusters.

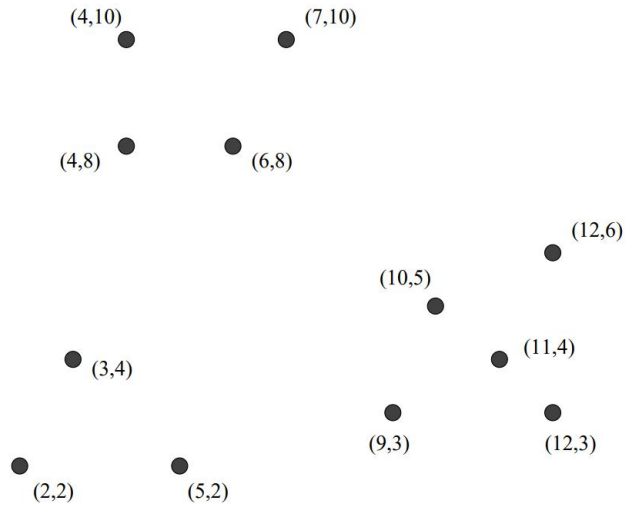


Figure 7.2: Twelve points to be clustered hierarchically

Hierarchical Clustering

Key operation: *Repeatedly combine* two nearest clusters

- **(1) How to represent a cluster of many points?**
  - **Key problem:** As you merge clusters, how do you represent the "location" of each cluster, to tell which pair of clusters is closest?
  - **Euclidean case:** each cluster has a ***centroid*** = average of its (data)points
- **(2) How to determine "*nearness*" of clusters?**
  - Measure cluster distances by distances of centroids

Initially, each point is in a cluster by itself and is the centroid of that cluster.

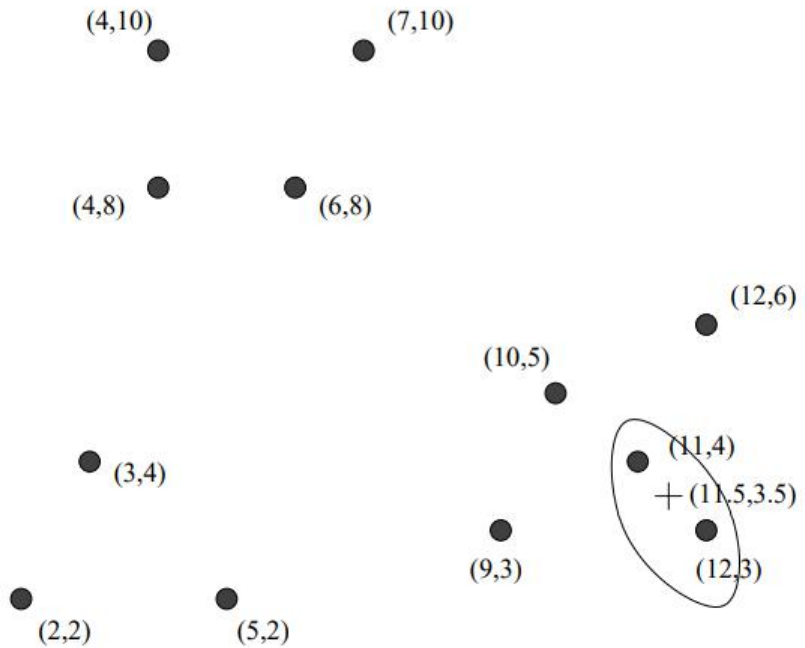


Figure 7.3: Combining the first two points into a cluster

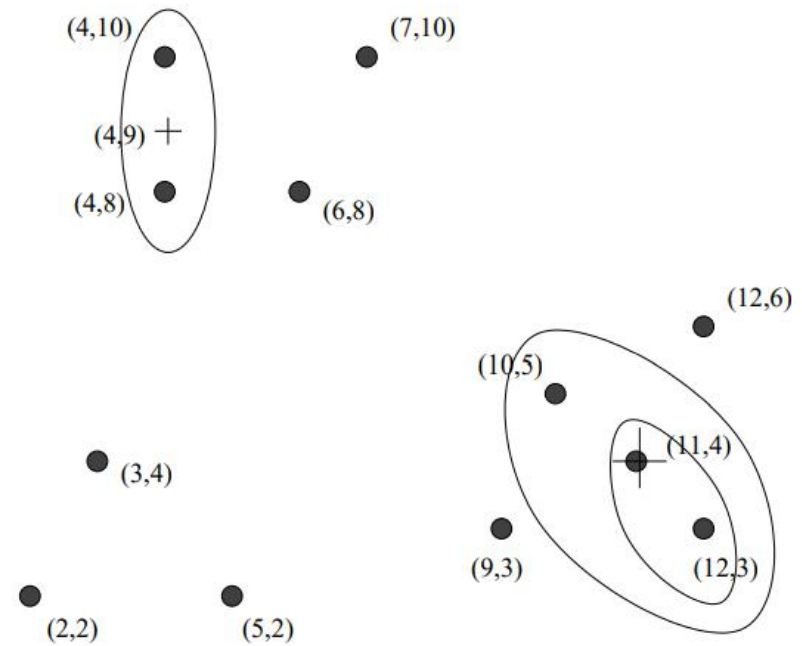


Figure 7.4: Clustering after two additional steps

Q1. Present the clustering of Example 7.2 if we use the minimum of the distances between any two points as the distance between two clusters.

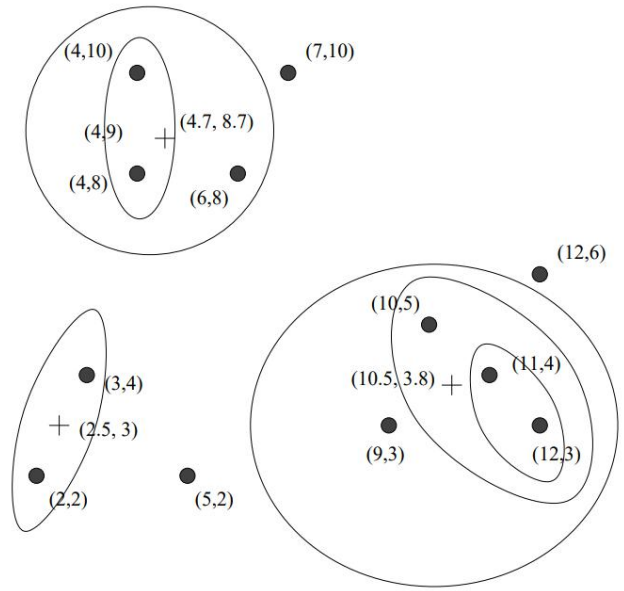
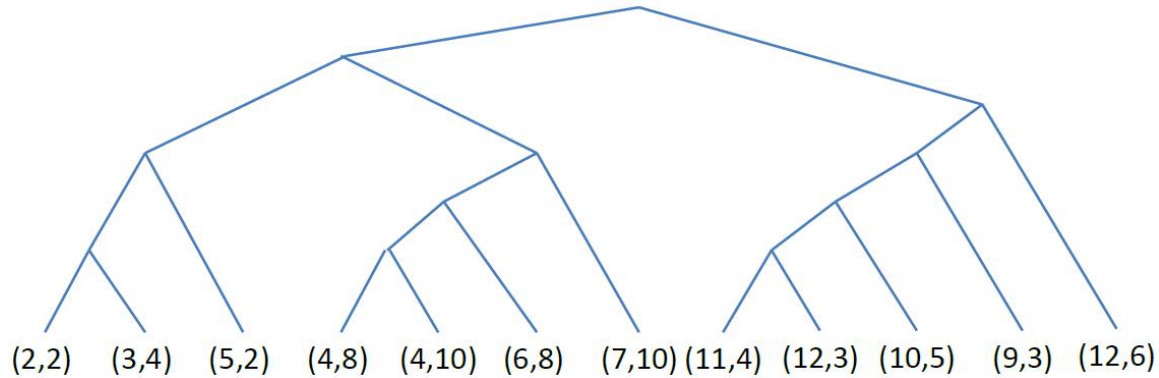


Figure 7.5: Three more steps of the hierarchical clustering



Q2.Let us reconsider the twelve points of Fig. 7.2. In the worst case, our initial choice of a point is near the center, say (6,8).

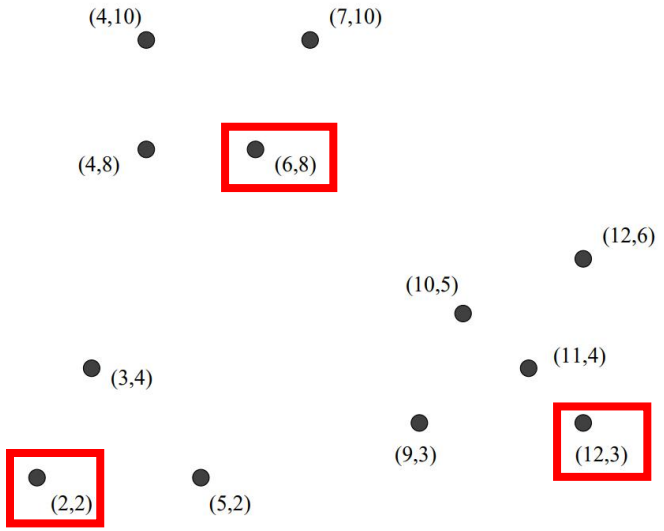


Figure 7.2: Twelve points to be clustered hierarchically

k–means Algorithm(s)

- Assumes Euclidean space/distance
- Start by picking ***k***, the number of clusters
- Initialize clusters by picking one point per cluster
  - **Example:** Pick one point at random, then ***k*-1** other points, each as far away as possible from the previous points
    - OK, as long as there are no *outliers* (points that are far from any reasonable cluster)

A. Find the other 2 starting point for the clustering by seeking the point with the largest minimum distance to the selected starting point(s).

The furthest point from (6,8) is (12,3), so that point is chosen next. Among the remaining ten points, the one whose minimum distance to either (6,8) or (12,3) is a maximum is (2,2). That point has distance $\sqrt{52} = 7.21$ from (6,8) and distance $\sqrt{101} = 10.05$ to (12,3); thus its "score" is 7.21. You can check easily that any other point is less than distance 7.21 from at least one of (6,8) and (12.3). Our selection of three starting points is thus (6,8), (12,3), and (2,2). Notice that these three belong to different clusters.

B. Compute the representation of the cluster as in the BFR Algorithm. That is, compute N, SUM, and SUMSQ.

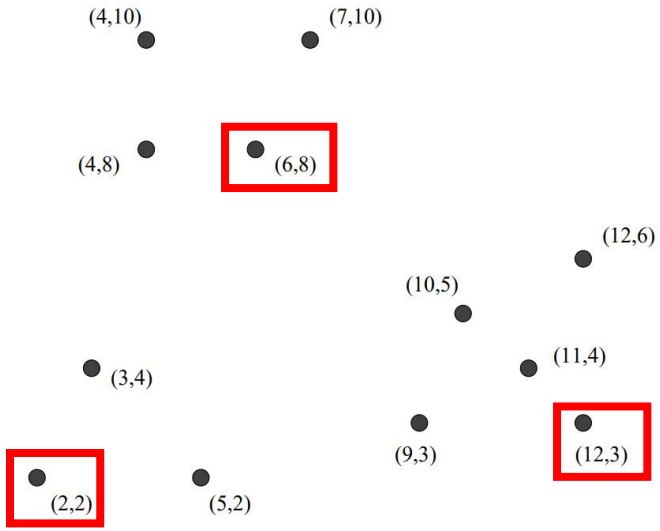


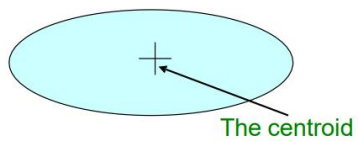
Figure 7.2: Twelve points to be clustered hierarchically

cluster	points	N	SUM	SUMSQ
1	(4, 8) (4, 10) (6, 8) (7, 10)	4	(21, 36)	(117, 328)
2	(9, 3) (10, 5) (11, 4) (12, 3) (12, 6)	5	(54, 21)	(590, 95)
3	(2, 2) (3, 4) (5,2)	3	(10, 8)	(38, 24)

Summarizing Sets of Points

For each cluster, the discard set (DS) is summarized by:

- The number of points, ***N***
- The vector ***SUM***, whose i^{th} component is the sum of the coordinates of the points in the i^{th} dimension
- The vector ***SUMSQ***: i^{th} component = sum of squares of coordinates in i^{th} dimension



C. Compute the variance and standard deviation of each cluster in each of the two dimensions.

cluster	points	N	SUM	SUMSQ
1	(4, 8) (4, 10) (6, 8) (7, 10)	4	(21, 36)	(117, 328)
2	(9, 3) (10, 5) (11, 4) (12, 3) (12, 6)	5	(54, 21)	(590, 95)
3	(2, 2) (3, 4) (5,2)	3	(10, 8)	(38, 24)

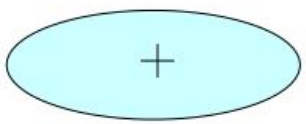
Summarizing Points: Comments

- **2*d* + 1** values represent any size cluster
  - ***d*** = number of dimensions
- Average in **each dimension** (***the centroid***) can be Calculated as $\mathbf{SUM_i / N}$
  - $\mathbf{SUM_i}$ = i^{th} component of SUM
- Variance of a cluster's discard set in dimension *i* is:

  $$\mathbf{(SUMSQ_i / N) - (SUM_i / N)^2}$$

  - And standard deviation is the square root of that
- **Next step: Actual clustering**

Note: Dropping the "axis-aligned" clusters assumption would require storing full covariance matrix to summarize the cluster. So, instead of **SUMSQ** being a ***d***-dim vector, it would be a ***d x d*** matrix, which is too big!



The variance in the i-th dimension is $SUMSQ_i/N - (SUM_i/N)^2$. The standard deviation in each dimension is the square root of the variance.

	cluster 1		cluster 2		cluster 3	
	x	y	x	y	x	y
variance	$\frac{27}{16}$	1	$\frac{34}{25}$	$\frac{34}{25}$	$\frac{14}{9}$	$\frac{8}{9}$
standard deviation	$\frac{3}{4}\sqrt{3}$	1	$\frac{\sqrt{34}}{5}$	$\frac{\sqrt{34}}{5}$	$\frac{\sqrt{14}}{3}$	$\frac{2}{3}\sqrt{2}$

Q3 Suppose a cluster of three-dimensional points has standard deviations of 2, 3, and 5, in the three dimensions, in that order. Compute the Mahalanobis distance between the origin (0, 0, 0) and the point (1,−3, 4).

Mahalanobis Distance

Normalized Euclidean distance from centroid

- For point $(\boldsymbol{x_1}, \ldots, \boldsymbol{x_d})$ and centroid $(\boldsymbol{c_1}, \ldots, \boldsymbol{c_d})$
  1. Normalize in each dimension: $\boldsymbol{y_i} = (\boldsymbol{x_i} - \boldsymbol{c_i}) / \boldsymbol{\sigma_i}$
  2. Take sum of the squares of the $\boldsymbol{y_i}$
  3. Take the square root

$$d(x, c) = \sqrt{\sum_{i=1}^{d} \left(\frac{x_i - c_i}{\sigma_i}\right)^2}$$

σ_i … standard deviation of points in the cluster in the i^{th} dimension

Then the Mahalanobis distance between p(1,-3,4) and c(0,0,0) is

$$\sqrt{\sum_{i=1}^{3} \left(\frac{p_i - c_i}{\sigma_i}\right)^2} = \sqrt{\left(\frac{1}{2}\right)^2 + \left(-\frac{3}{3}\right)^2 + \left(\frac{4}{5}\right)^2} = \frac{3}{10}\sqrt{21} \approx 1.37$$

Q4.The following figure gives a matrix with six rows.

Element	S1	S2	S3	S4
0	0	1	0	1
1	0	1	0	0
2	1	0	0	1
3	0	0	1	0
4	0	0	1	1
5	1	0	0	0

A. Compute the permutations by using the following three hash functions respectively:

$h1(x) = (2x + 1) \bmod 6$
$h2(x) = (3x + 2) \bmod 6$
$h3(x) = (5x + 2) \bmod 6$

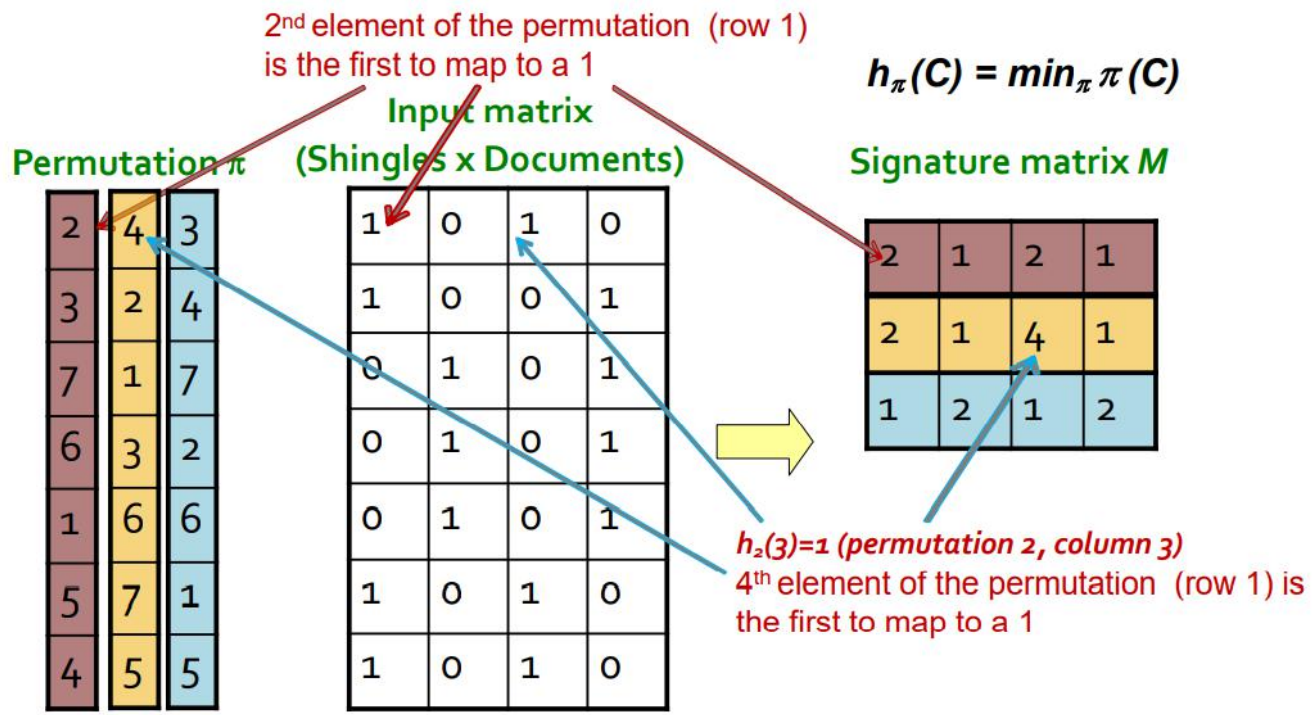
Indicate which of these permutations is true permutation and explain why?

Element	S1	S2	S3	S4	2x+1 mod 6	3x+2 mod 6	5x+2 mod 6
0	0	1	0	1	1	2	2
1	0	1	0	0	3	5	1
2	1	0	0	1	5	2	0
3	0	0	1	0	1	5	5
4	0	0	1	1	3	2	4
5	1	0	0	0	5	5	3

h3 is the true permutation because each index appears only once.

B. Compute the Minhash signature for each column if we use the three permutations in A.

Min-Hashing Example



Implementation Trick

```
for each row r do begin
   for each hash function h_i do
       compute h_i(r);
   for each column c
      if c has 1 in row r
         for each hash function h_i do
            if h_i < M(i, c) then
               M(i, c) := h_i(r);
end;
```

2x+1 mod 6	3x+2 mod 6	5x+2 mod 6
1	2	2
3	5	1
5	2	0
1	5	5
3	2	4
5	5	3

S1	S2	S3	S4
0	1	0	1
0	1	0	0
1	0	0	1
0	0	1	0
0	0	1	1
1	0	0	0

The final minhash signature matrix is:

S1	S2	S3	S4
5	1	1	1
2	2	2	2
0	1	4	0

C. How close are the estimated Jaccard similarities for the six pairs of columns to the true Jaccard similarities?

Similarity for Signatures

- We know: **Pr[$h_\pi(C_1) = h_\pi(C_2)$] = $sim(C_1, C_2)$**
- Now generalize to multiple hash functions
- The *similarity of two signatures* is the fraction of the hash functions in which they agree
- Thus, the expected similarity of two signatures equals the Jaccard similarity of the columns or sets that the signatures represent
  - And the longer the signatures, the smaller will be the expected error

Jaccard similarity: $sim(D_1, D_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$

sig

S1	S2	S3	S4
5	1	1	1
2	2	2	2
0	1	4	0

col

Element	S1	S2	S3	S4
0	0	1	0	1
1	0	1	0	0
2	1	0	0	1
3	0	0	1	0
4	0	0	1	1
5	1	0	0	0

similarities	1-2	1-3	1-4	2-3	2-4	3-4
col/col	0	0	0.25	0	0.25	0.25
sig/sig	0.33	0.33	0.67	0.67	0.67	0.67