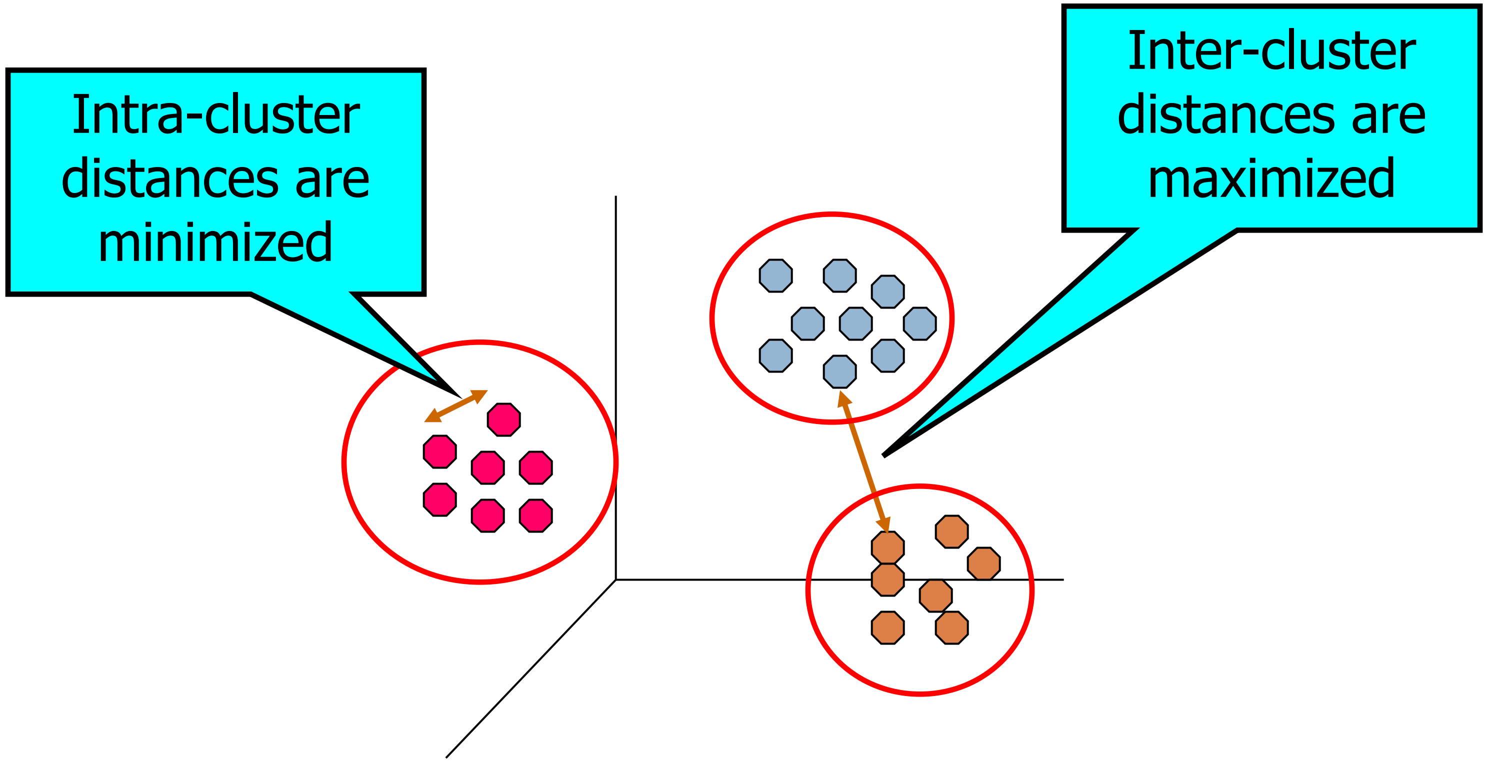
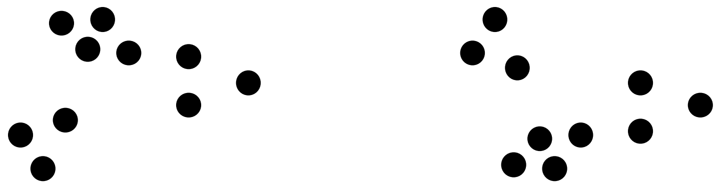
CLUSTERING

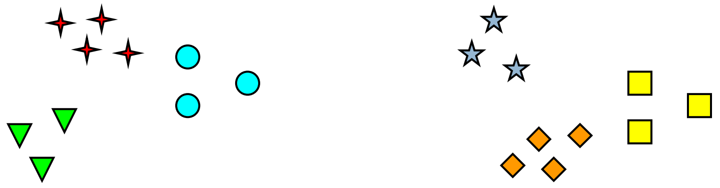
What is Cluster Analysis?

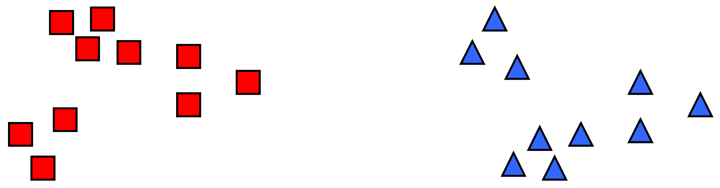
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups (understanding & summarization)

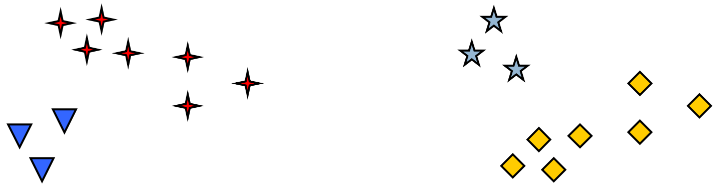

Intra-cluster distances are minimized
Inter-cluster distances are maximized

Notion of a Cluster can be Ambiguous


How many clusters?


Six Clusters


Two Clusters


Four Clusters

Types of Clusterings

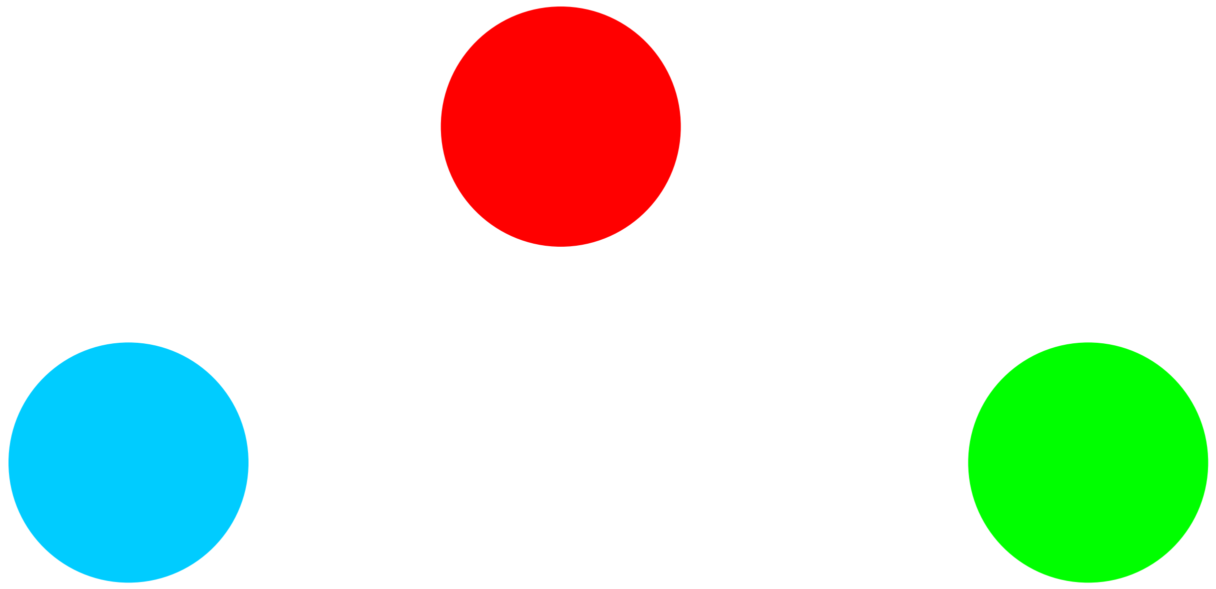
- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or 'border' points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics

Types of Clusters: Well-Separated

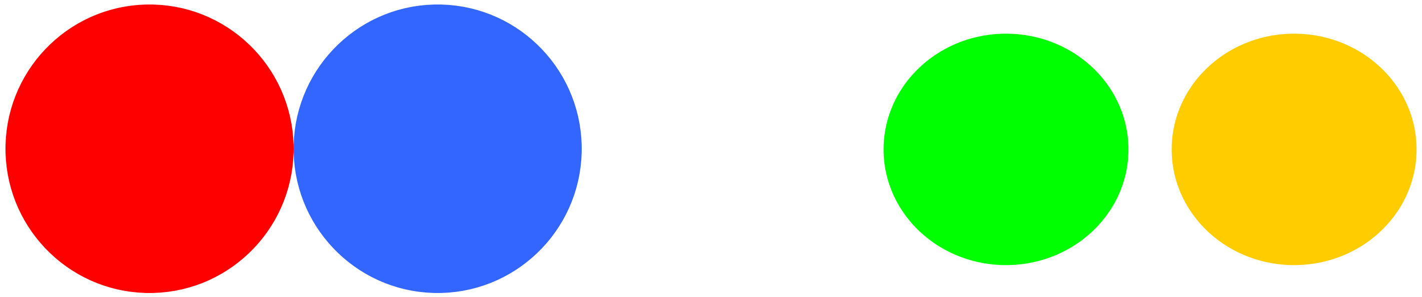
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Center-Based

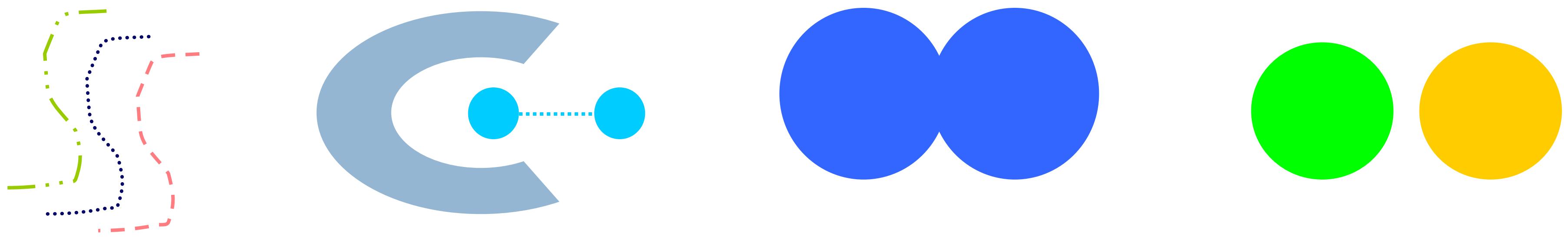
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster



4 center-based clusters

Types of Clusters: Contiguity-Based

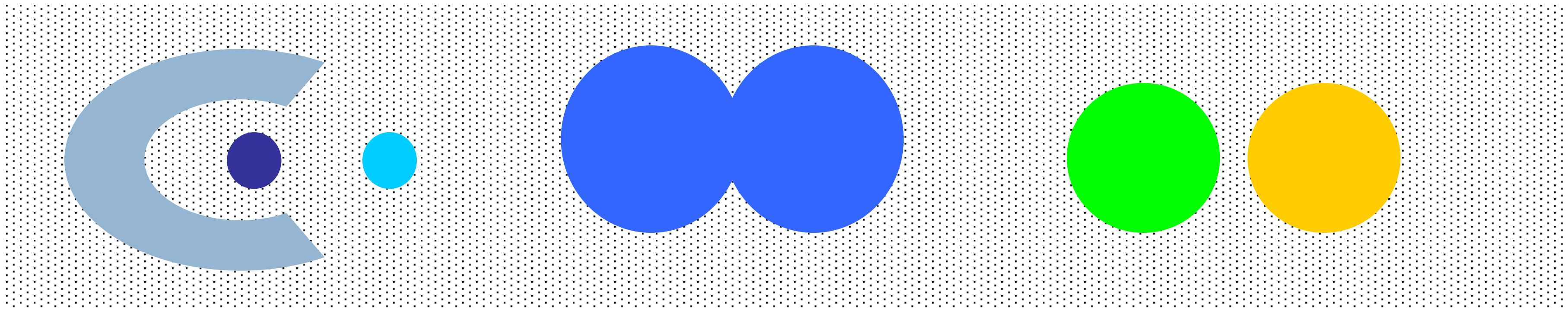
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular and when noise and outliers are present.



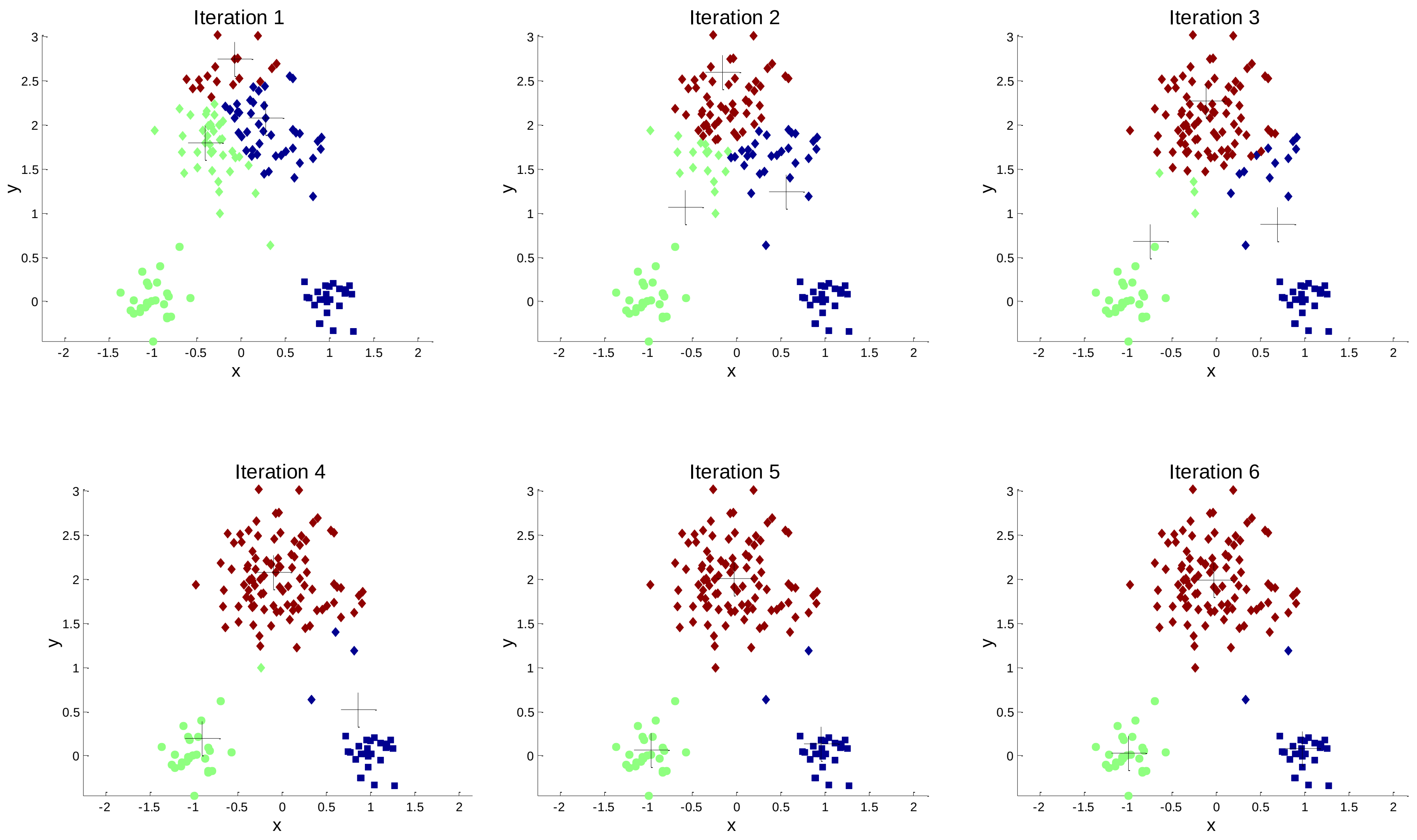
6 density-based clusters

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

```
1: Select $K$ points as the initial centroids.
2: **repeat**
3:     Form $K$ clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.
5: **until** The centroids don't change
```

K-means


Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
x
y
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3

K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’

K-means Clustering – Details

- centroid can vary, depending on the proximity measure for the data and the goal of the clustering
- goal of the clustering is typically expressed by an objective function that depends on the proximities of the points to one another or to the cluster centroids
- e.g., minimize the squared distance of each point to its closest centroid

Evaluating K-means Clusters

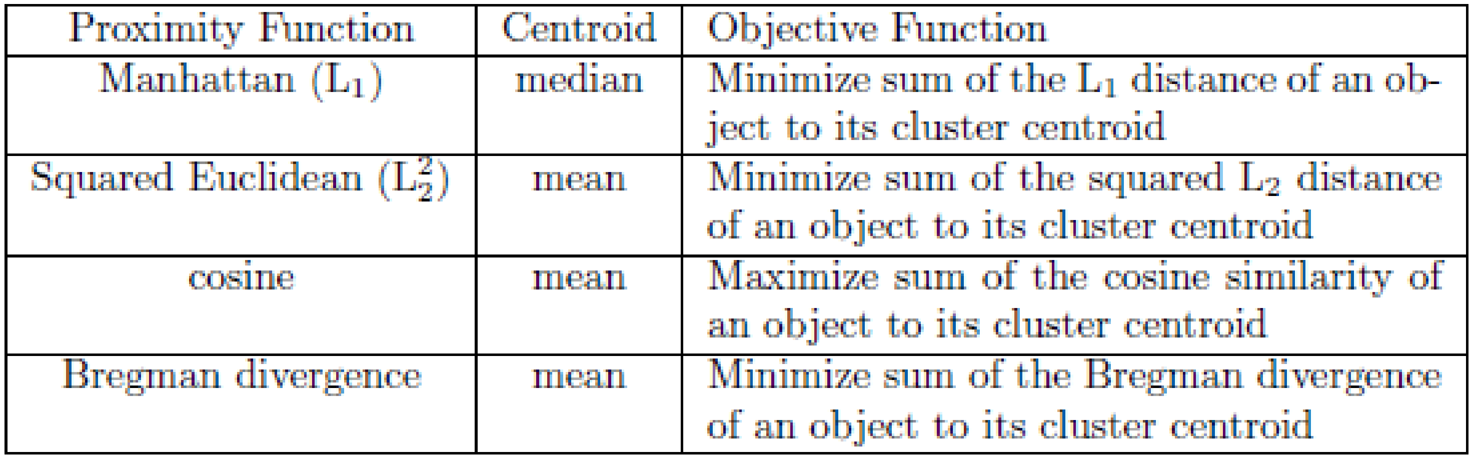
- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - *x* is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error

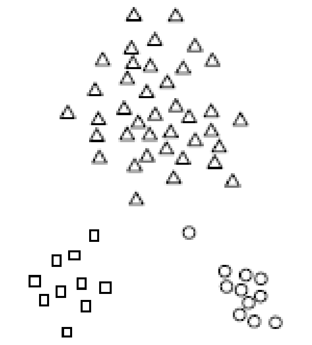
Common choices

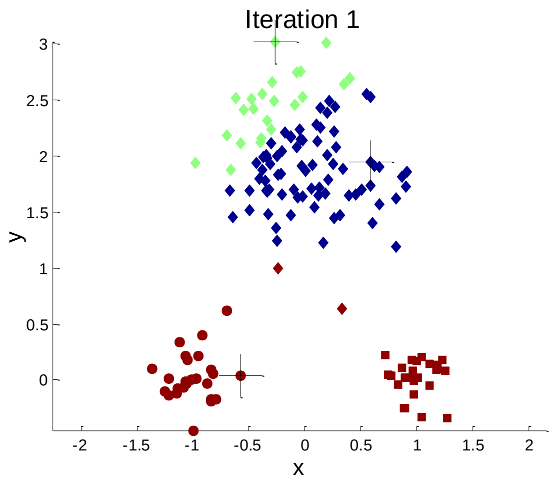
Table 8.2. K-means: Common choices for proximity, centroids, and objective functions.

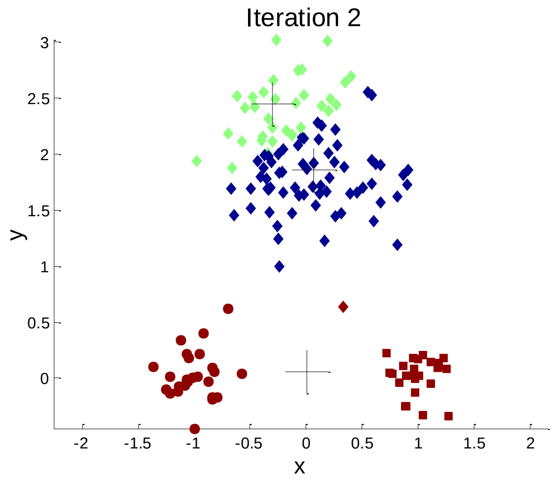


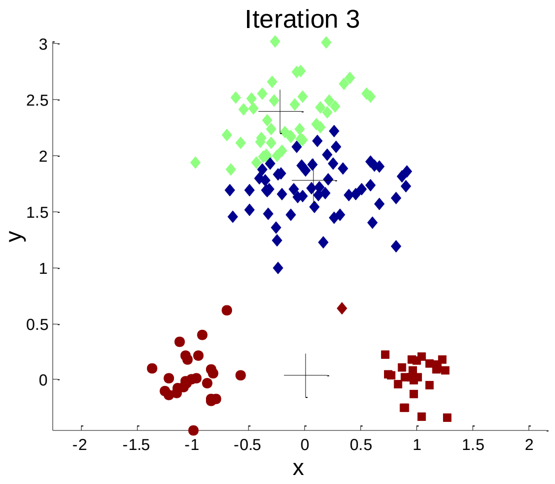
Proximity Function	Centroid	Objective Function
Manhattan (L_1)	median	Minimize sum of the L_1 distance of an object to its cluster centroid
Squared Euclidean (L_2^2)	mean	Minimize sum of the squared L_2 distance of an object to its cluster centroid
cosine	mean	Maximize sum of the cosine similarity of an object to its cluster centroid
Bregman divergence	mean	Minimize sum of the Bregman divergence of an object to its cluster centroid

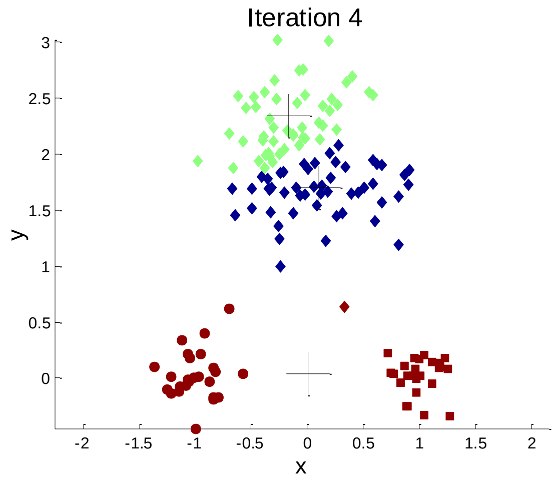
Importance of Choosing Initial Centroids …

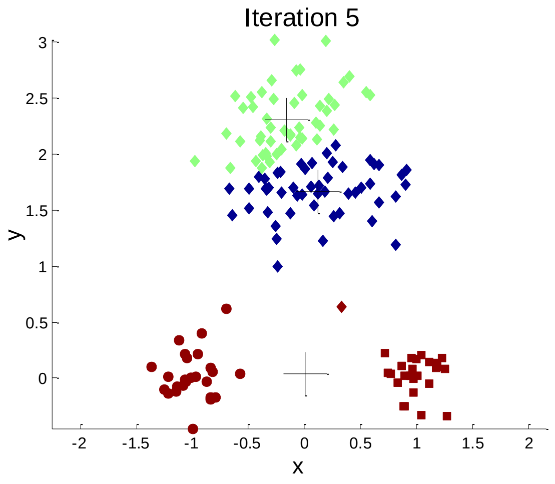

(a) Optimal clustering.











Solutions to Initial Centroids Problem

- perform multiple runs, each with a different set of randomly chosen initial centroids, and then select the set of clusters with the minimum SSE
- Sample and use hierarchical clustering to determine initial centroids
- Select the first point at random
- For each successive initial centroid, select the point that is farthest from any of the initial centroids already selected
- Outliers & expensive

Handling Empty Clusters

- Basic K-means algorithm can yield empty clusters
- Several strategies
  - Choose the point that contributes most to SSE
  - Choose a point from the cluster with the highest SSE

Pre-processing and Post-processing

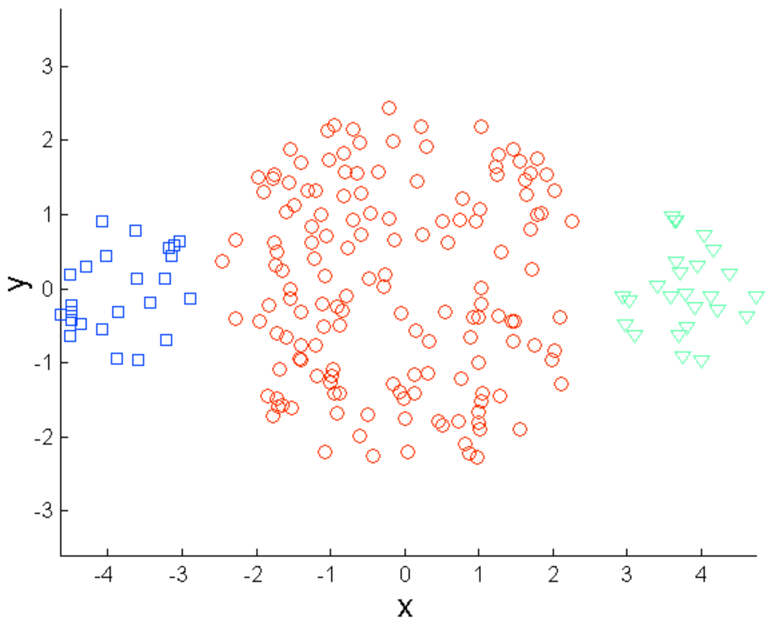
- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

Limitations of K-means

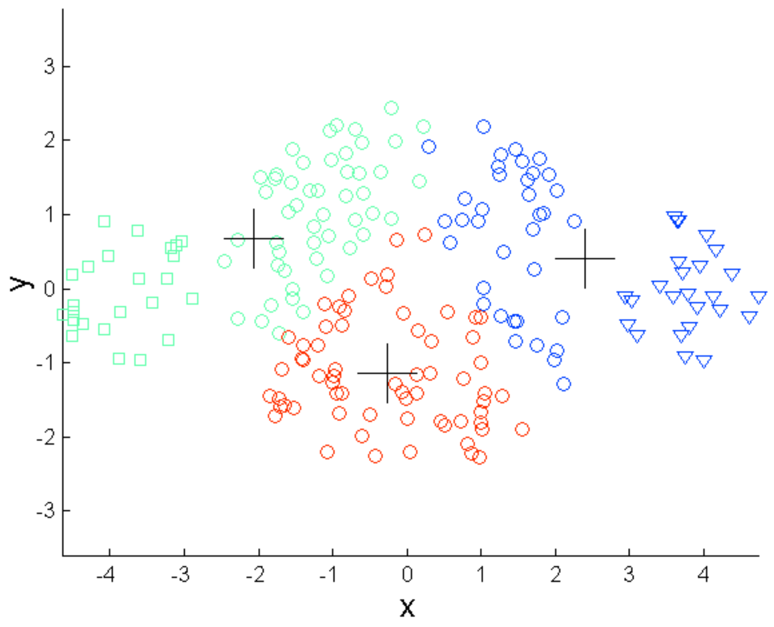
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

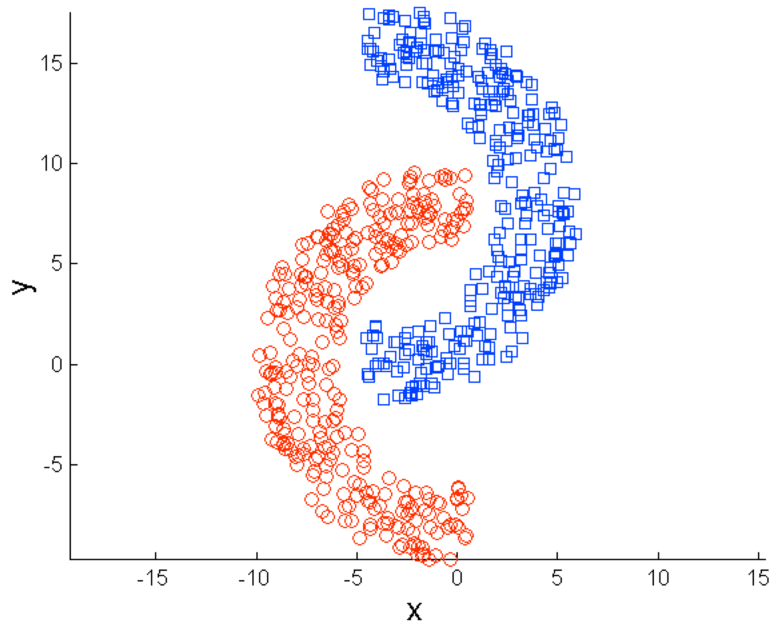


Original Points

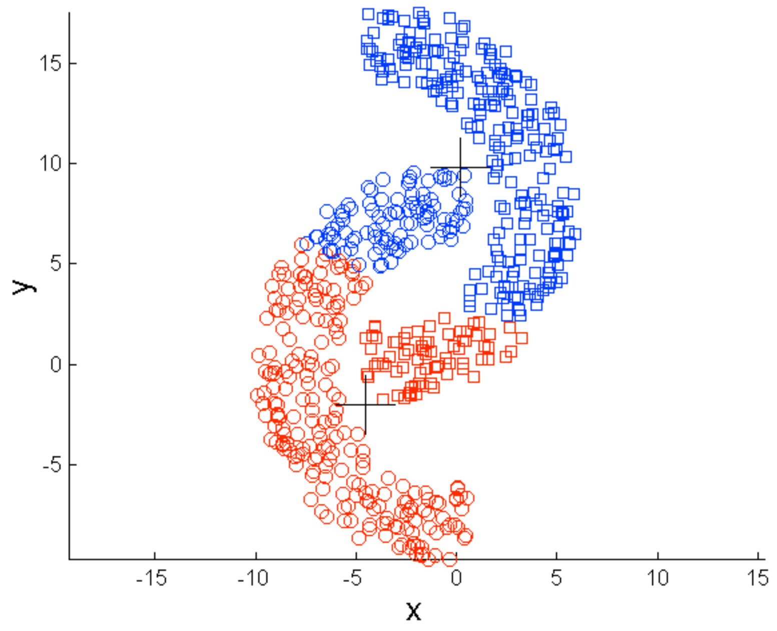


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points

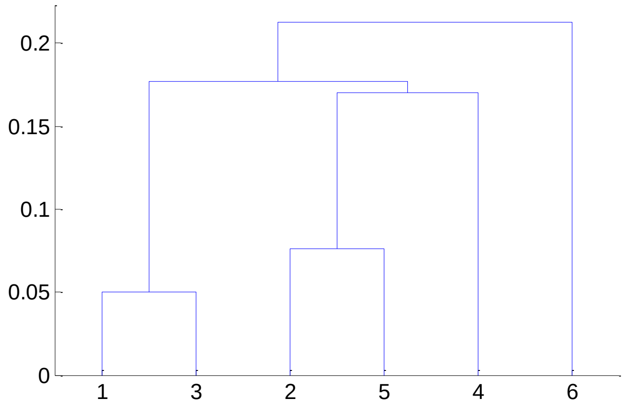


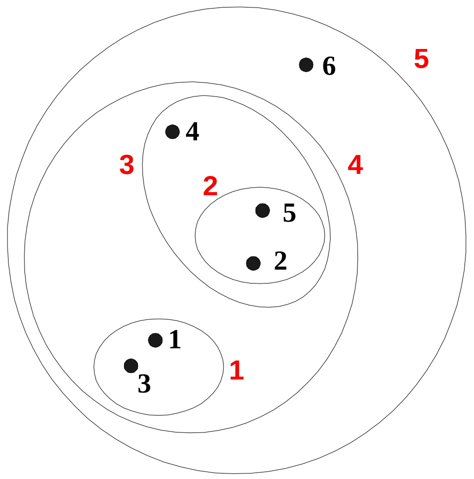
K-means (2 Clusters)

Hierarchical Clustering

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique

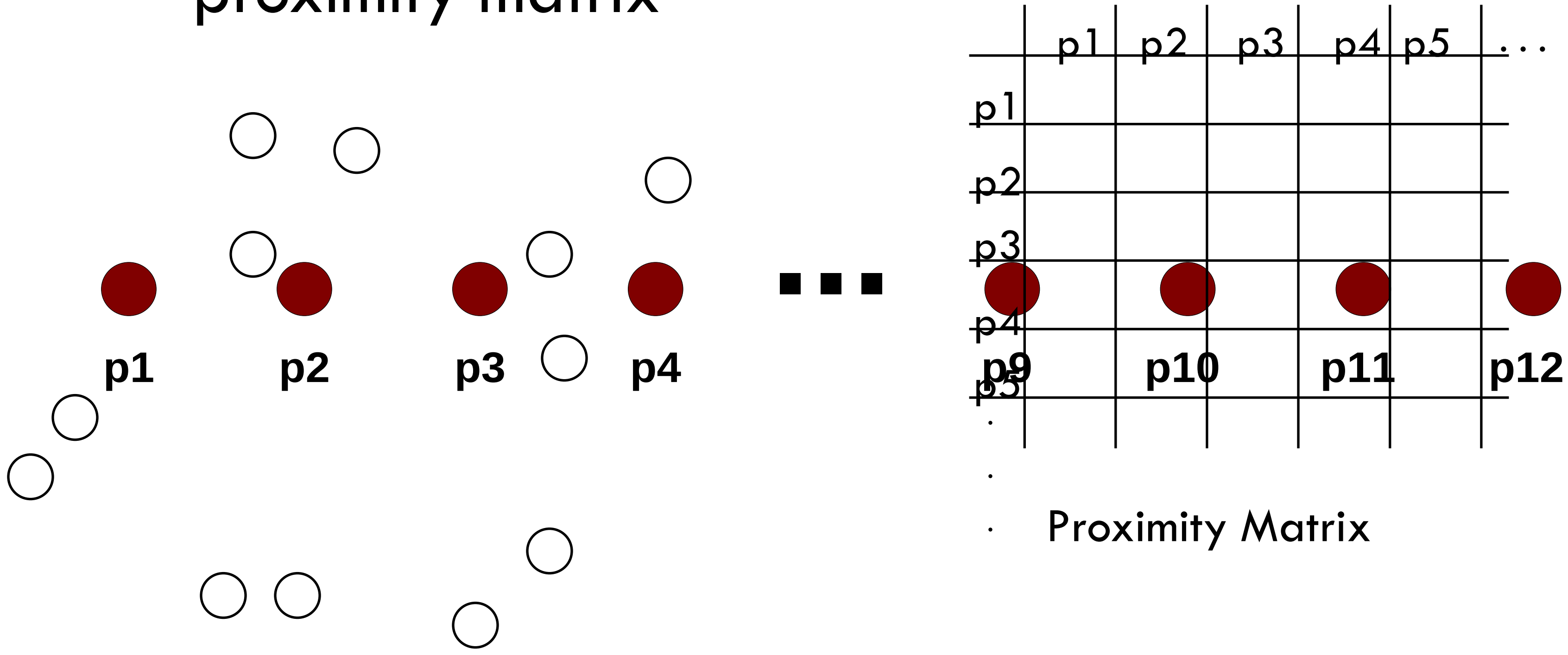
- Basic algorithm is straightforward

```
1. Compute the proximity matrix
2. Let each data point be a cluster
3. Repeat
4.         Merge the two closest clusters
5.         Update the proximity matrix
6. Until only a single cluster remains
```

- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

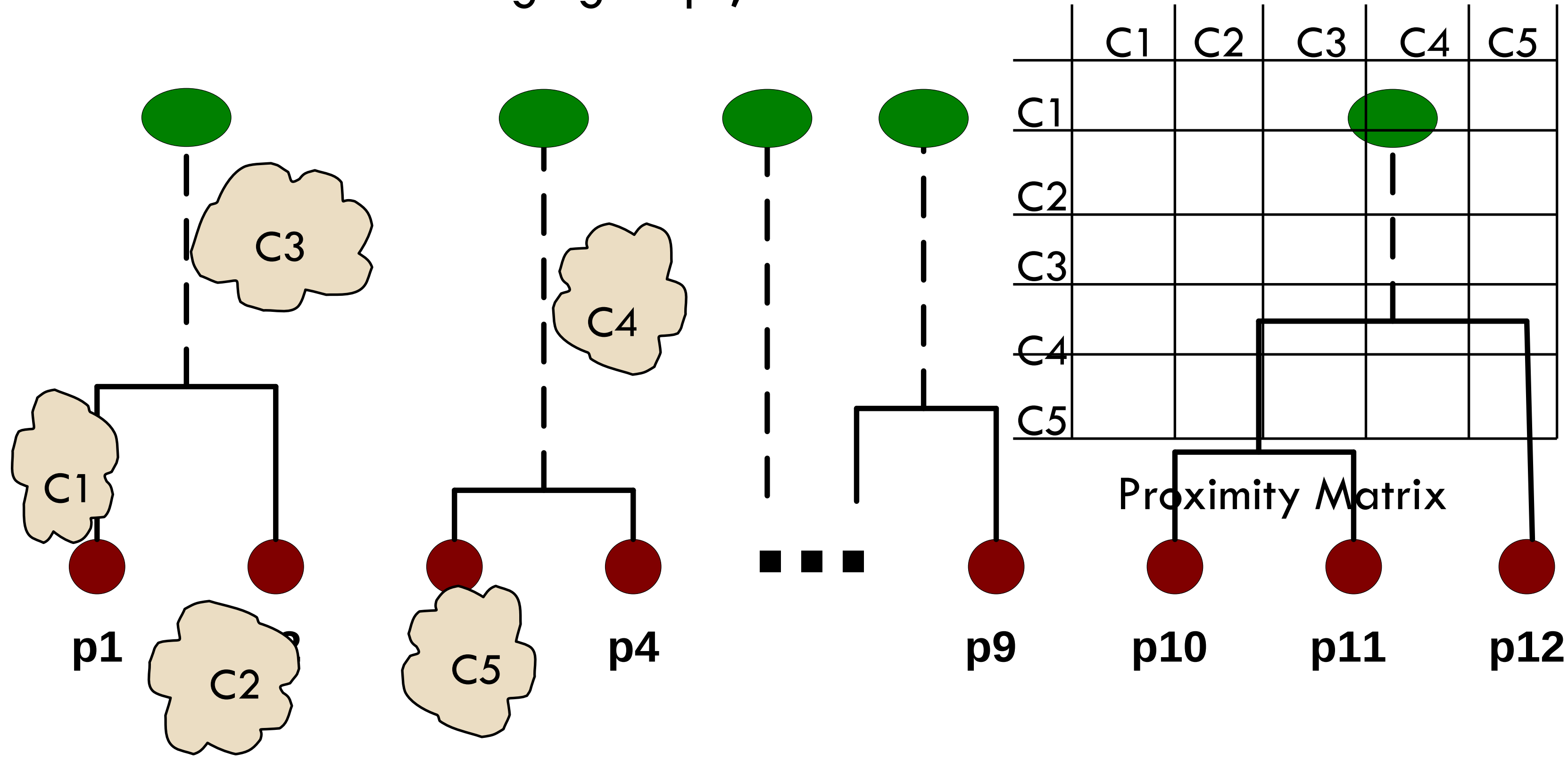
Starting Situation

- Start with clusters of individual points and a proximity matrix



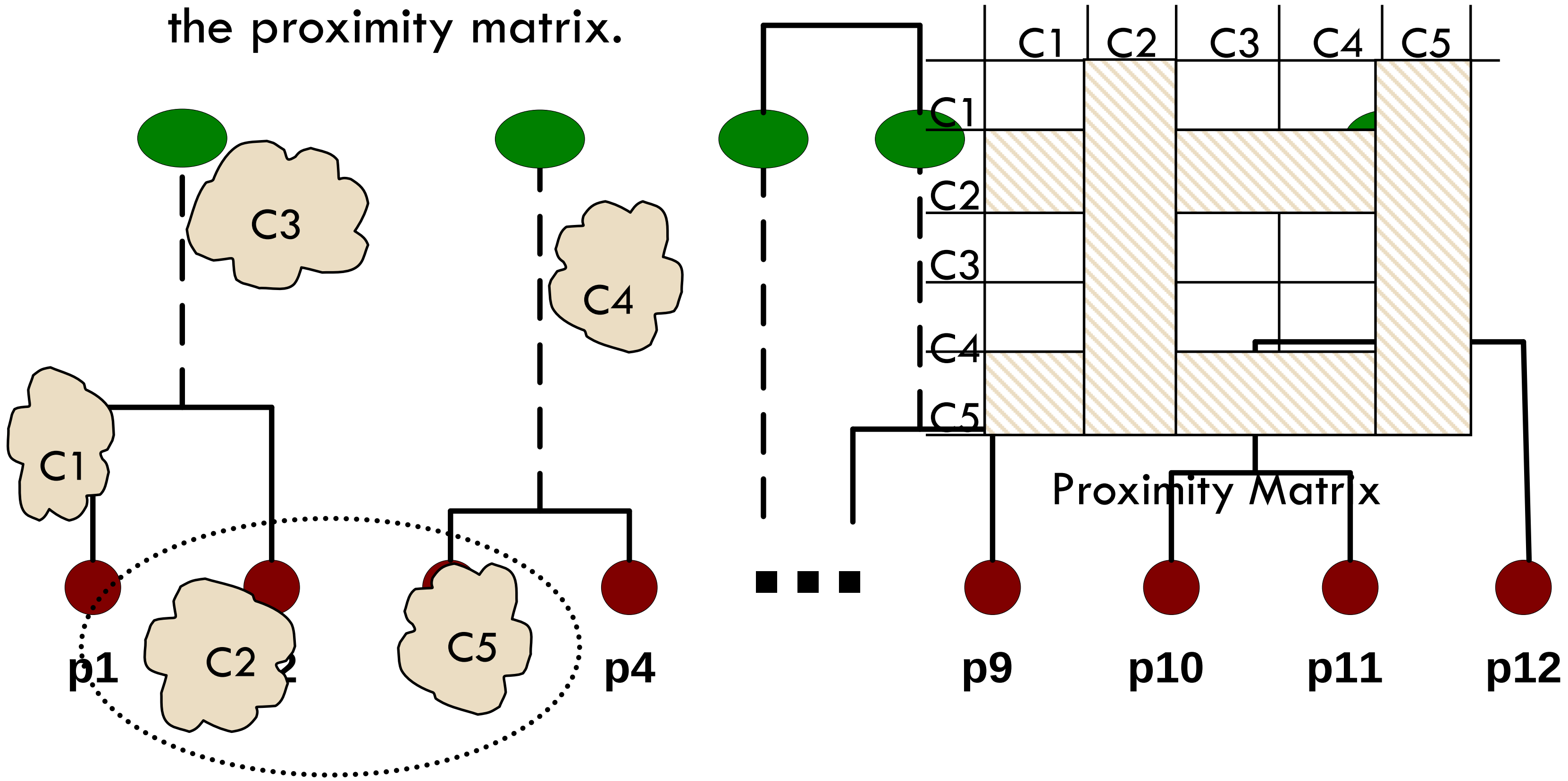
Intermediate Situation

- After some merging steps, we have some clusters



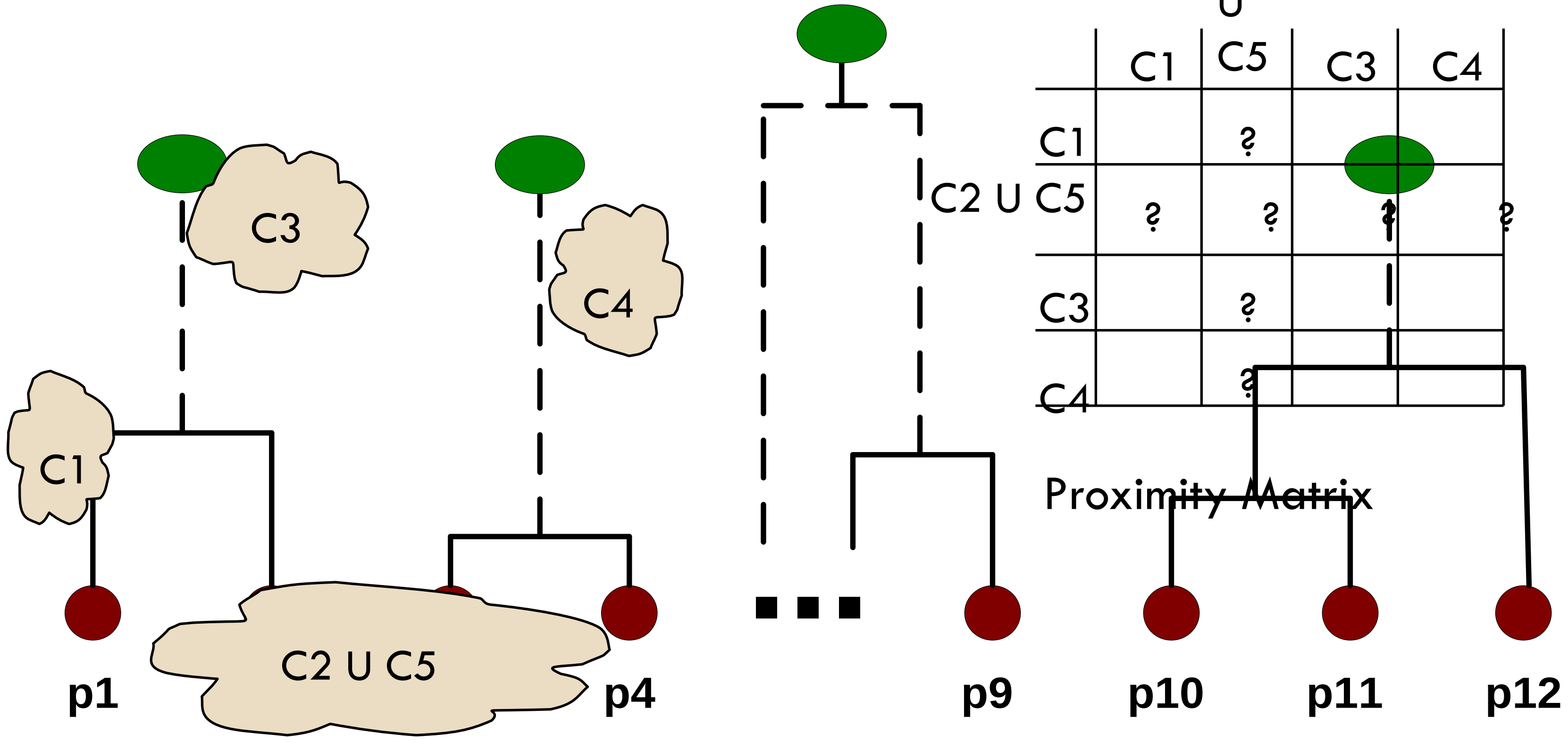
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



After Merging

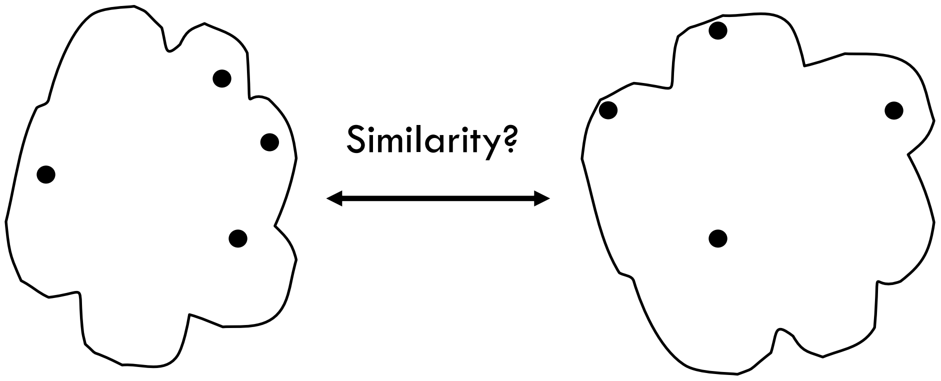
- The question is “How do we update the proximity matrix?”



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix

How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

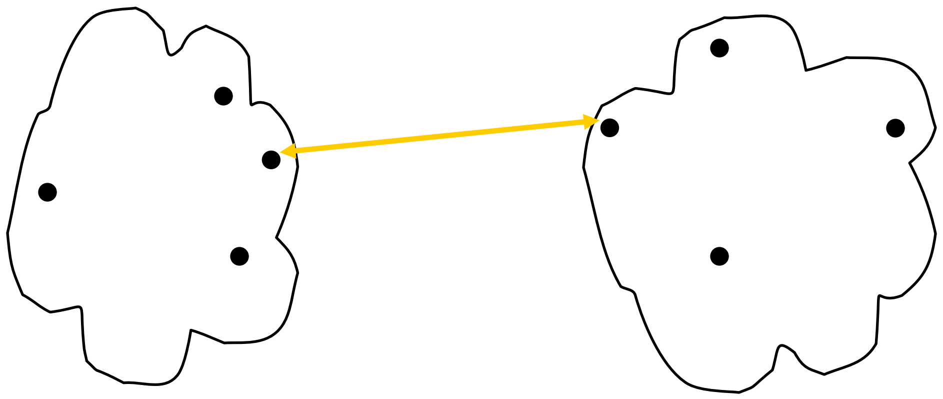
.

.

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to Define Inter-Cluster Similarity

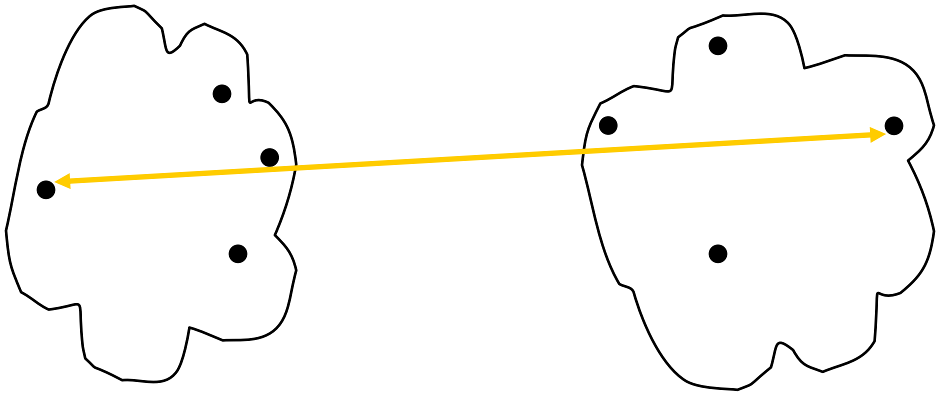


- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity

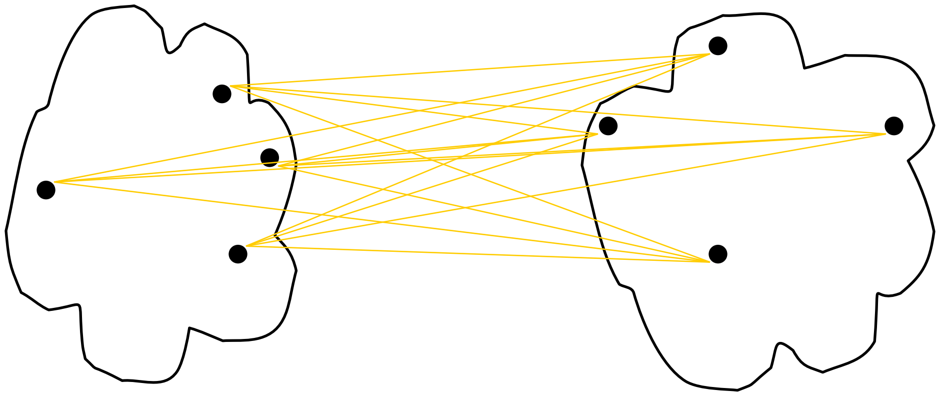


- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

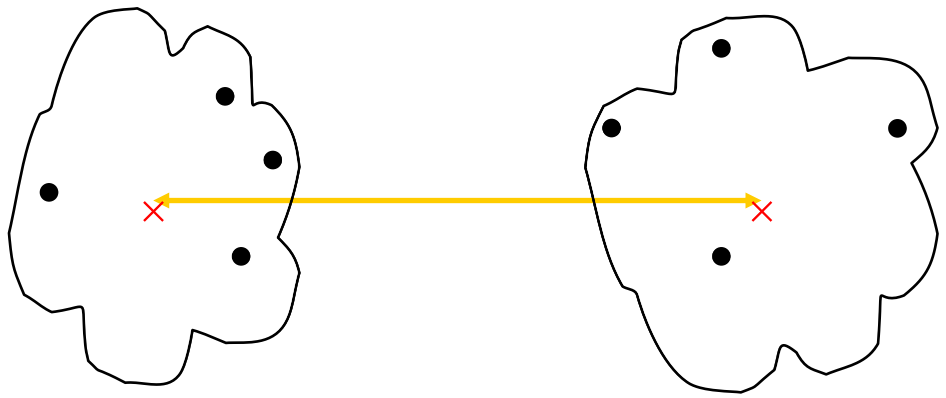
.

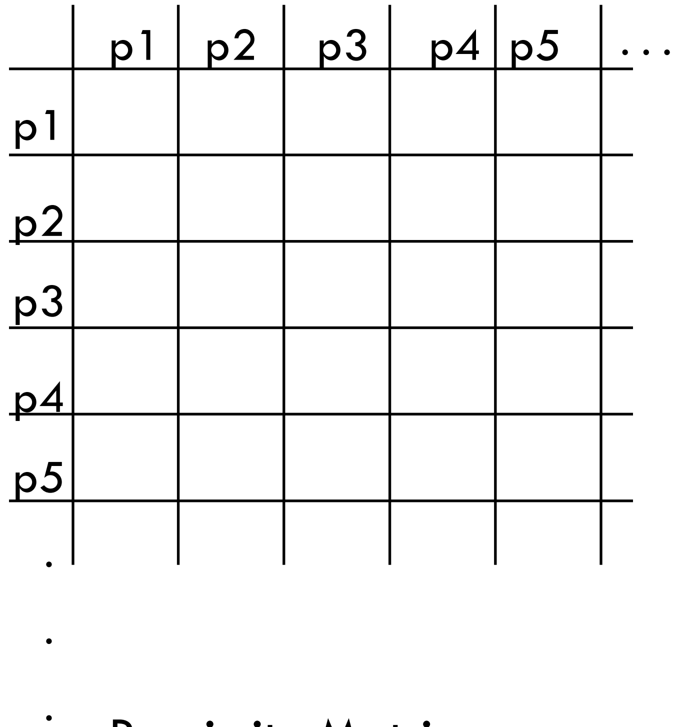
.

.

Proximity Matrix

How to Define Inter-Cluster Similarity





Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

Example: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
- Determined by one pair of points, i.e., by one link in the proximity graph

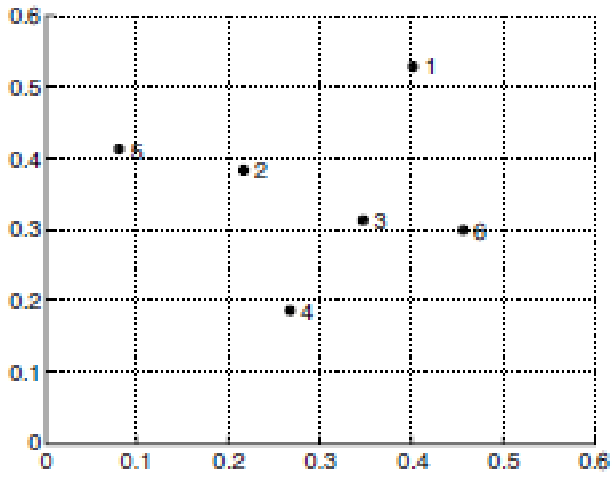


Figure 8.15. Set of 6 two-dimensional points.

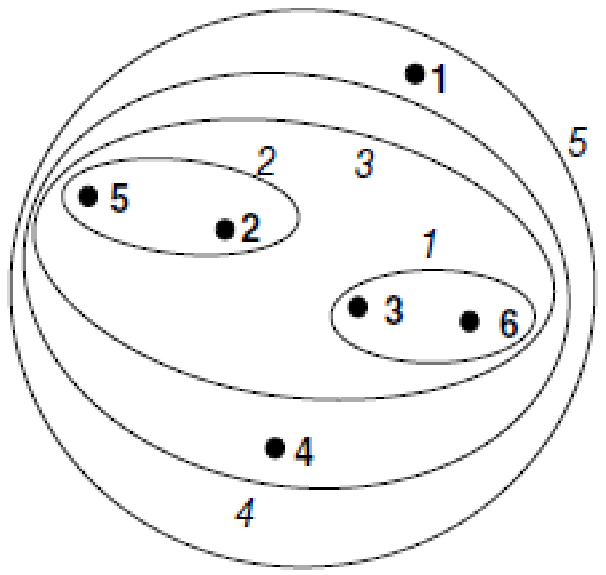
Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Table 8.3. xy coordinates of 6 points.

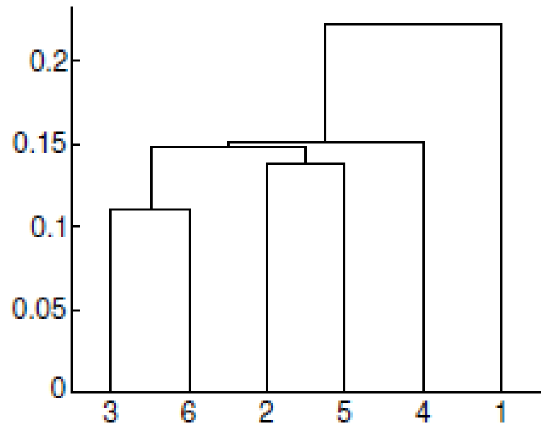
Example: MIN or Single Link

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.



(a) Single link clustering.



(b) Single link dendrogram.

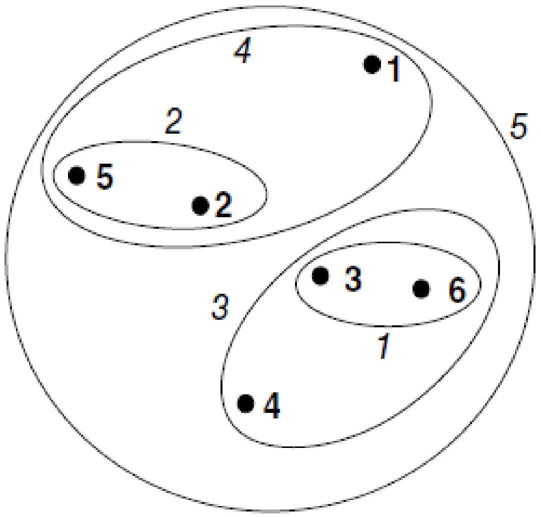
Example: MAX or Complete Linkage

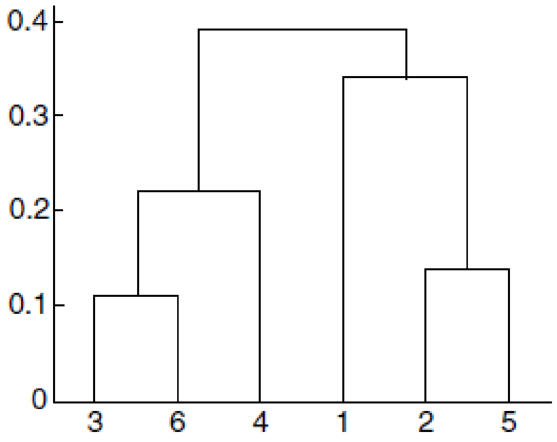
- ✓ Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
- ✓ Determined by all pairs of points in the two clusters

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4 Euclidean distance matrix for 6 points

$$\begin{aligned}
dist(\{3,6\},\{4\}) &= \max(dist(3,4), dist(6,4)) \\
&= \max(0.15, 0.22) \\
&= 0.22. \\
dist(\{3,6\},\{2,5\}) &= \max(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\
&= \max(0.15, 0.25, 0.28, 0.39) \\
&= 0.39. \\
dist(\{3,6\},\{1\}) &= \max(dist(3,1), dist(6,1)) \\
&= \max(0.22, 0.23) \\
&= 0.23.
\end{aligned}$$





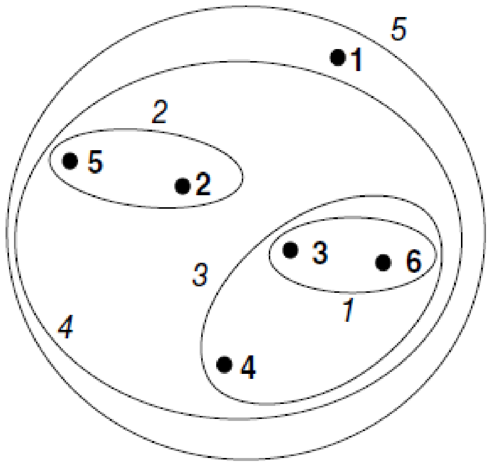
Group Average

proximity of two clusters is defined as the average pairwise proximity among all pairs of points in the different clusters.

$$proximity(C_i, C_j) = \frac{\sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_j}} proximity(\mathbf{x}, \mathbf{y})}{m_i * m_j}$$

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.



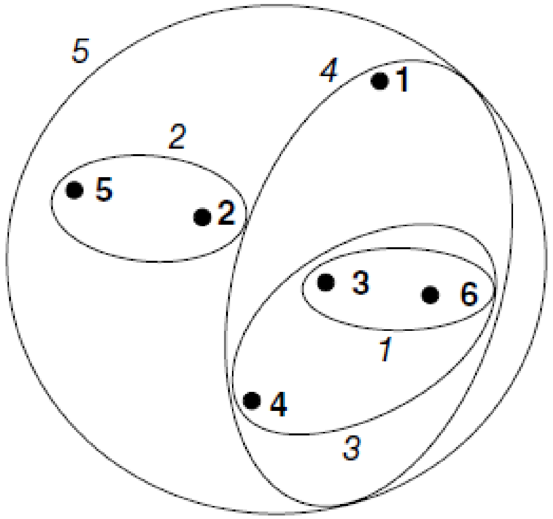
$$\begin{aligned} dist(\{3,6,4\},\{1\}) &= (0.22 + 0.37 + 0.23)/(3 * 1) \\ &= 0.28 \\ dist(\{2,5\},\{1\}) &= (0.2357 + 0.3421)/(2 * 1) \\ &= 0.2889 \\ dist(\{3,6,4\},\{2,5\}) &= (0.15 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29)/(6 * 2) \\ &= 0.26 \end{aligned}$$

Cluster Similarity: Ward's Method

- Proximity between two clusters is based on the increase in squared error when two clusters are merged
  - used to initialize K-means

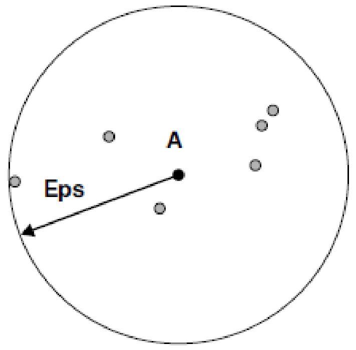
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.



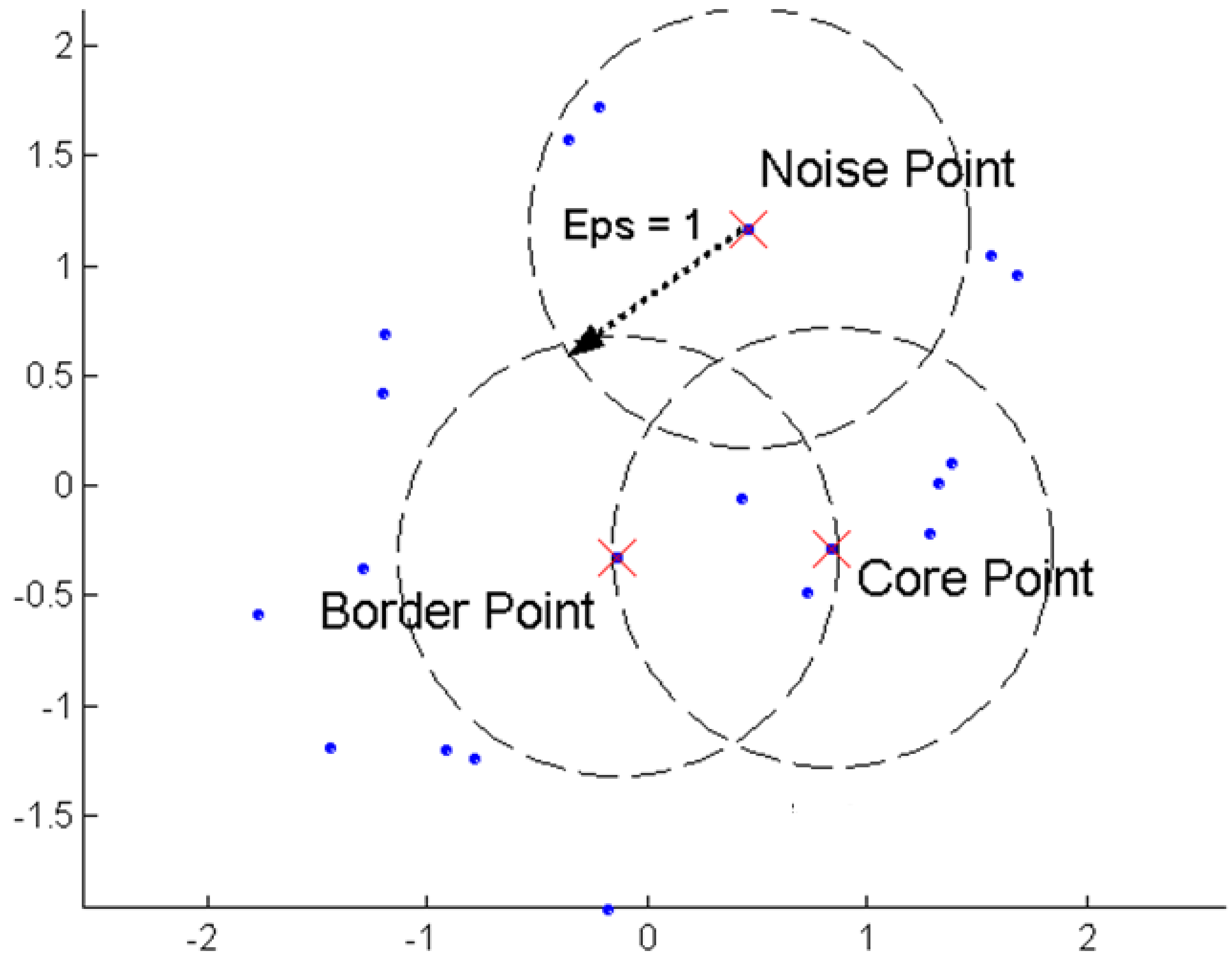
Density based Clustering:DBSCAN

DBSCAN

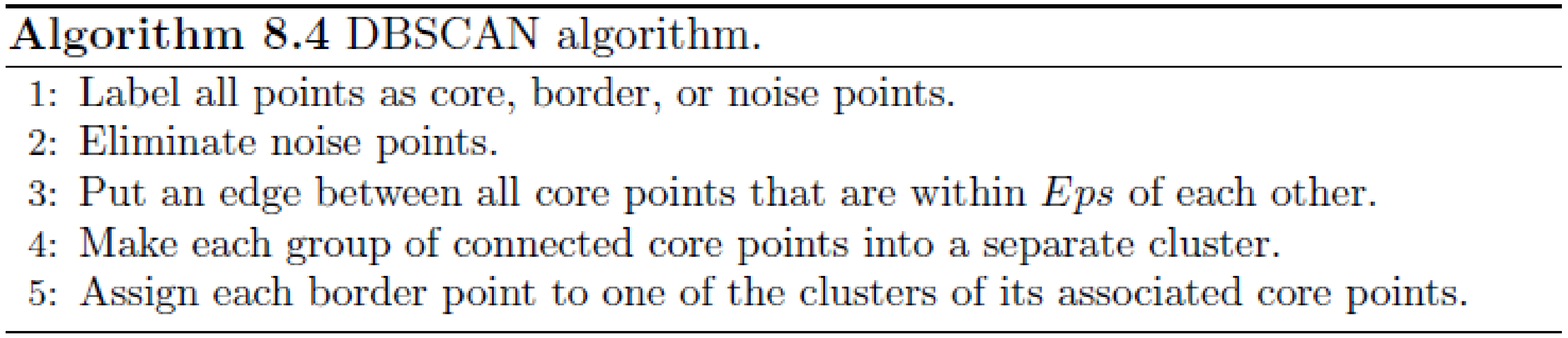


- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points



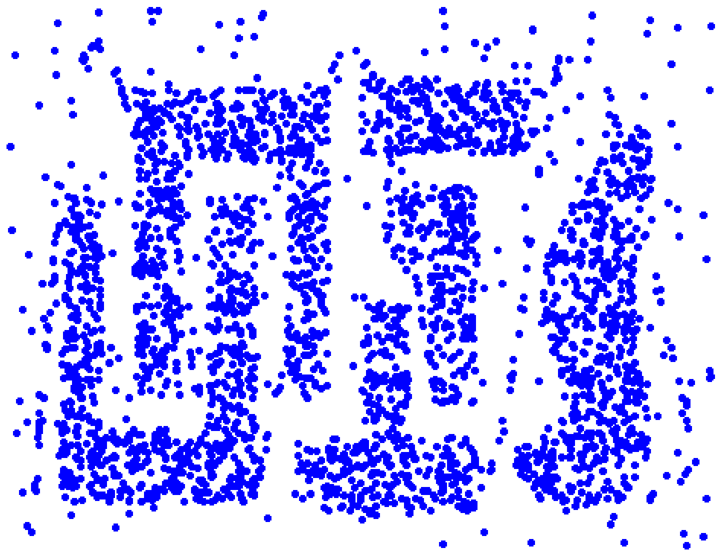
DBSCAN



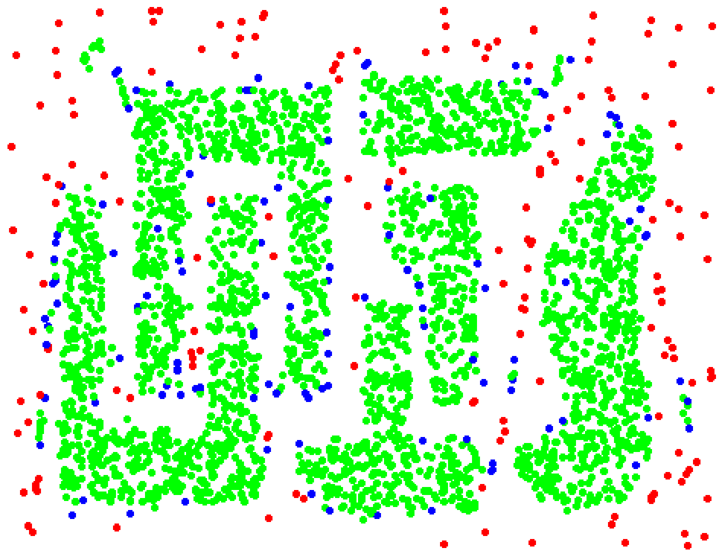
Algorithm 8.4 DBSCAN algorithm.

```
1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.
```

DBSCAN: Core, Border and Noise Points



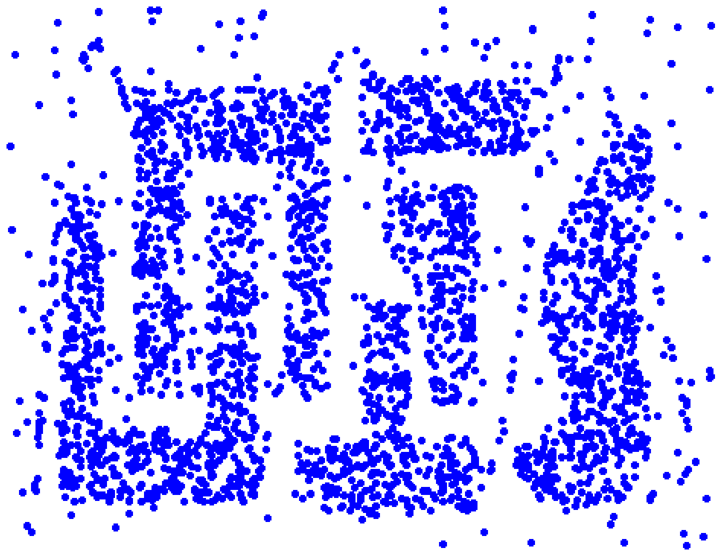
Original Points

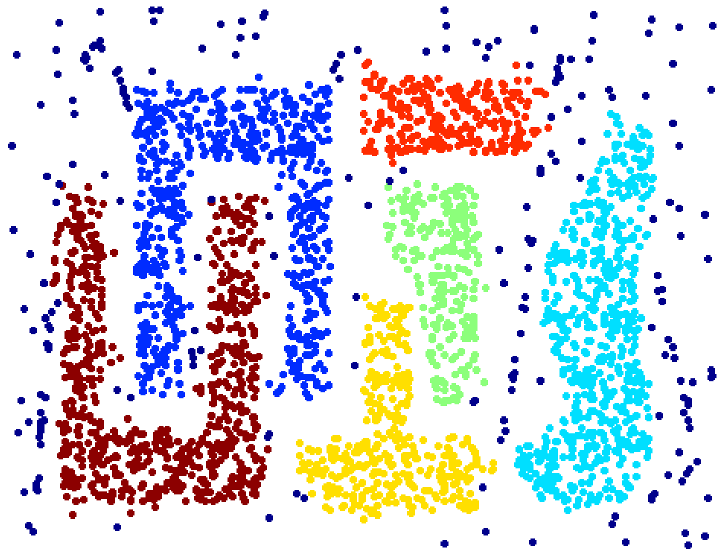


Point types: core, border and noise

Eps = 10, MinPts = 4

DBSCAN


Original Points


Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

DBSCAN: Determining EPS and MinPts

how to determine the parameters *Eps* and *MinPts*

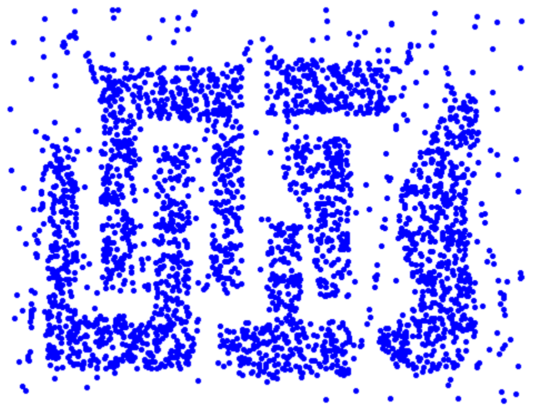
MinPts:

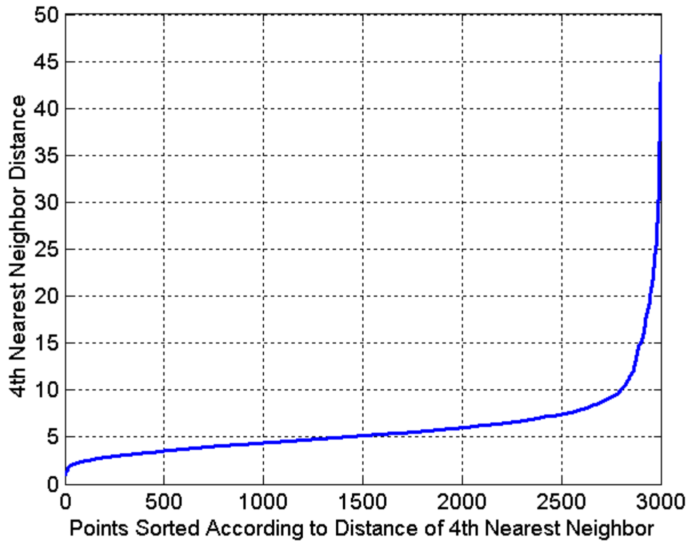
- ❖ MinPts=K too small, noise or outliers will be incorrectly labeled as clusters
- ❖ k is too large, small clusters are likely to be labeled as noise (k = 4)

Eps:

- ❖ look at the behavior of the distance from a point to its kth nearest neighbor(k-dist)
- ❖ Points belong to some cluster, the value of k-dist small if k is not larger than the cluster size
- ❖ points not in a cluster, such as noise points, the *k*-dist relatively large
- ❖ compute the *k*-dist for all the data points for some *k*
- ❖ sort them in increasing order, and then plot the sorted values
- ❖ a sharp change at the value of *k*-dist

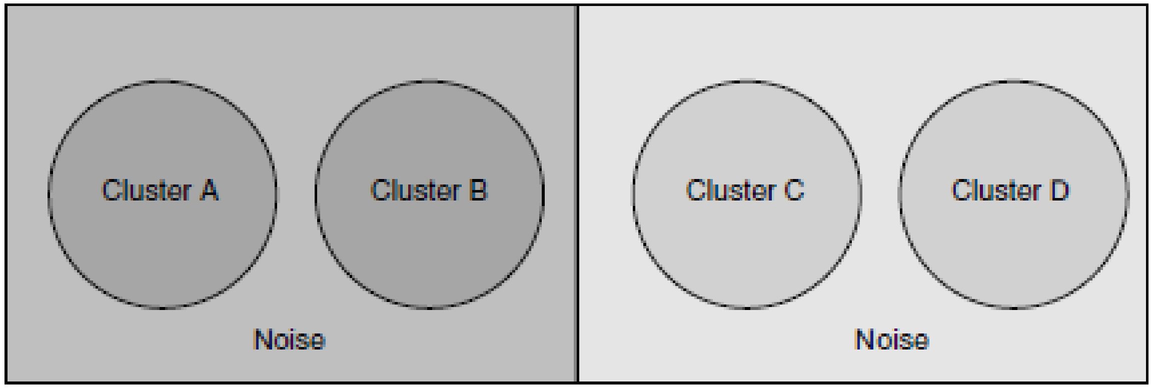
DBSCAN: Determining EPS and MinPts





Clusters of Varying Density

DBSCAN can have trouble with density if the density of clusters varies widely



- ➢ *Eps* threshold is low enough that DBSCAN finds C and *D* as clusters, then *A* and *B* and the points surrounding them will become a single cluster

- ➢ *Eps threshold high enough that DBSCAN finds A and B as separate clusters, and the points surrounding them are marked as noise, then C and D and the points surrounding them will also be marked as noise*

Cluster Validity

Cluster Validity

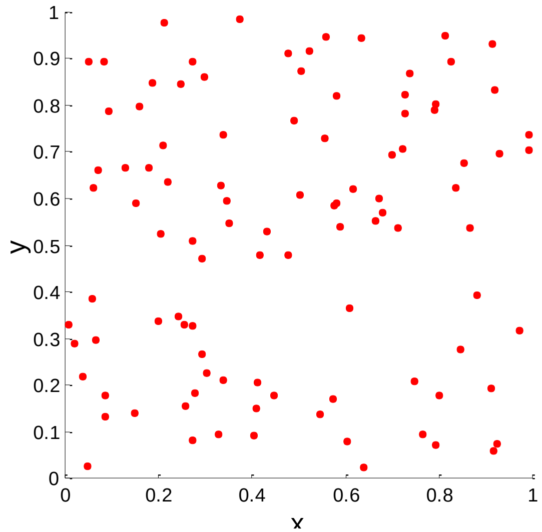
- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall

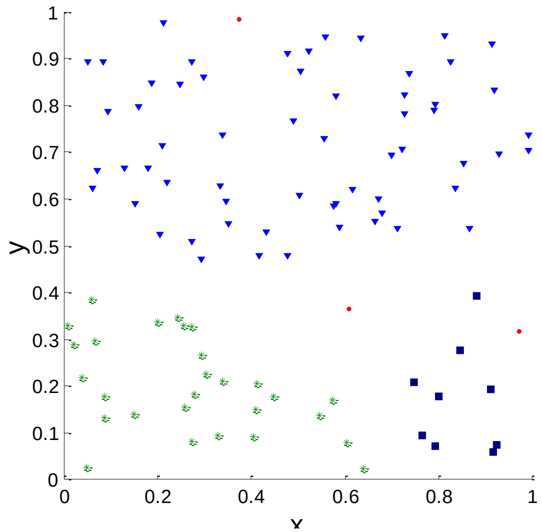
- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Clusters found in Random Data

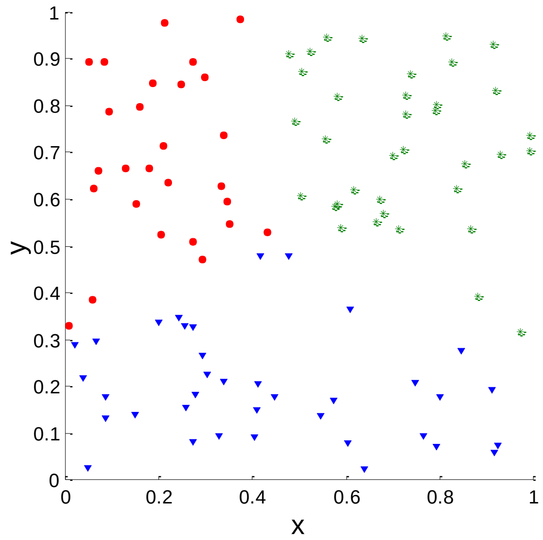
Random Points

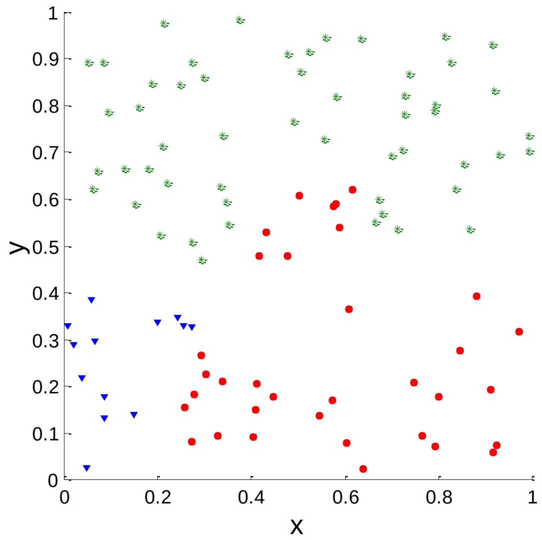




DBSCAN

K-means





Complete Link

Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - Unsupervised(Internal Index): Used to measure the goodness of a clustering structure *without* respect to external information.
  - Supervised(External Index): Used to measure the extent to which cluster labels match externally supplied class labels.
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function
- criteria

Unsupervised Measures: Cohesion and Separation

$$\boxed{overall\ validity = \sum_{i=1}^{K} w_i\ validity(C_i)}$$

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

Unsupervised Measures: Cohesion and Separation

Graph- based View

proximity graph has data objects as nodes, a link between each pair of data objects, and a weight assigned to each link that is the proximity between the two data objects connected by the link

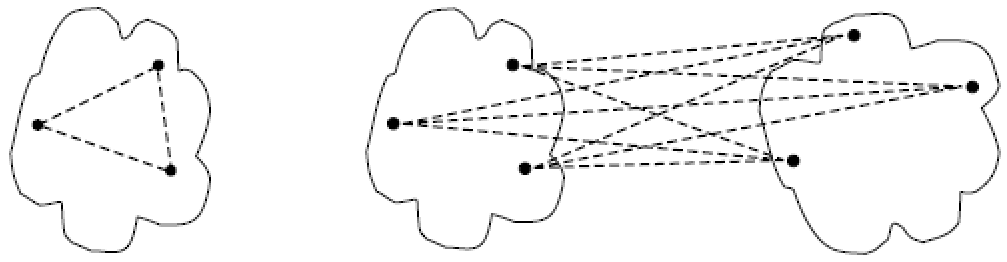


Figure 8.27. Graph-based view of cluster cohesion and separation.

$$cohesion(C_i) = \sum_{\substack{\mathbf{x} \in C_i \\ \mathbf{y} \in C_i}} proximity(\mathbf{x}, \mathbf{y})$$

$$separation(C_i, C_j) = \sum_{\substack{\mathbf{x} \in C_i \\ \mathbf{y} \in C_j}} proximity(\mathbf{x}, \mathbf{y})$$

Unsupervised Measures: Cohesion and Separation

Center-Based View

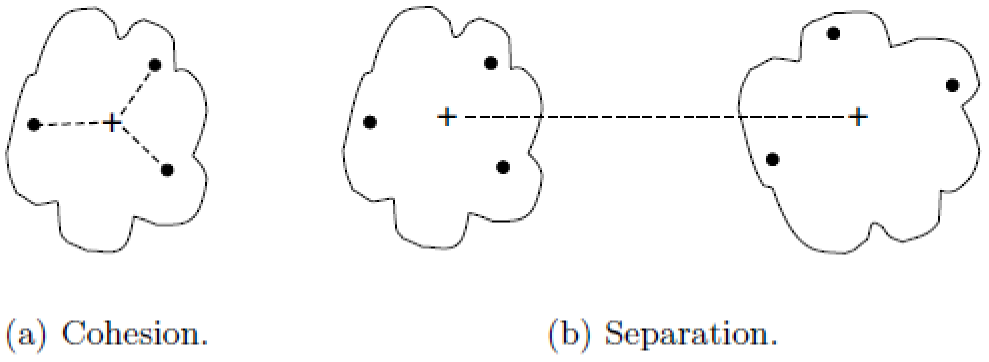


Figure 8.28. Prototype-based view of cluster cohesion and separation.

$$cohesion(C_i) = \sum_{\mathbf{x} \in C_i} proximity(\mathbf{x}, \mathbf{c}_i)$$

$$\begin{aligned} separation(C_i, C_j) &= proximity(\mathbf{c}_i, \mathbf{c}_j) \\ separation(C_i) &= proximity(\mathbf{c}_i, \mathbf{c}) \end{aligned}$$

Unsupervised Measures: Cohesion and Separation

In some cases, there is also a strong relationship between cohesion and separation

$$\text{TSS} = \sum_{i=1}^{K} \sum_{x \in C_i} (x - c)^2$$

- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$\boldsymbol{WSS} = \sum_{\boldsymbol{i}} \sum_{\boldsymbol{x} \in \boldsymbol{C}_i} (\boldsymbol{x} - \boldsymbol{m_i})^2$$

  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

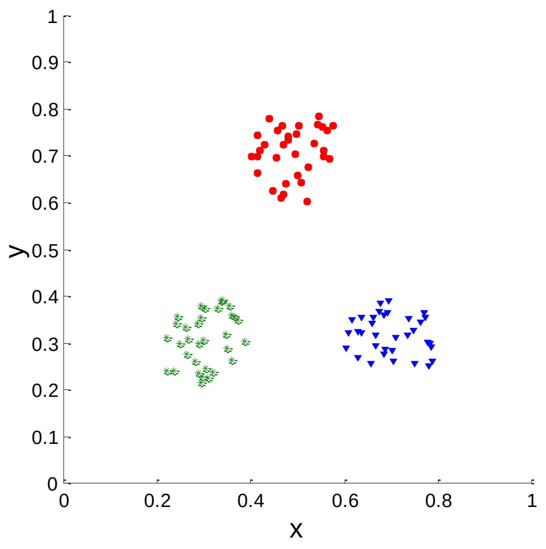
    - Where $|C_i|$ is the size of cluster i

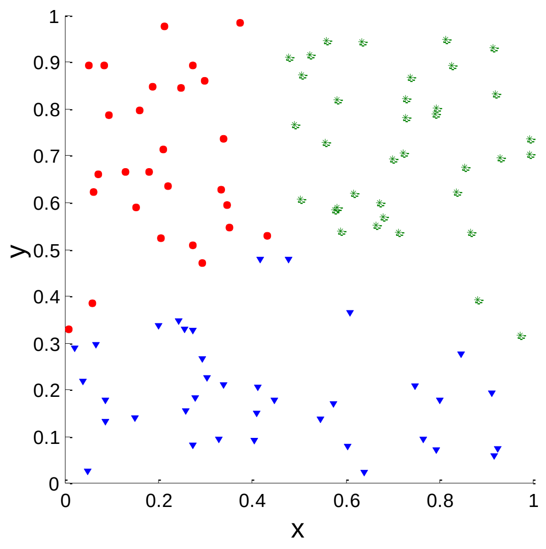
Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - “Incidence” Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density based clusters.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



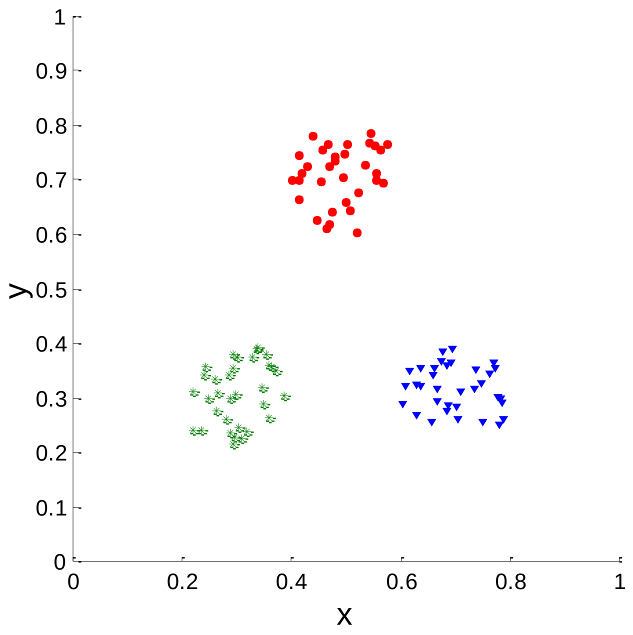


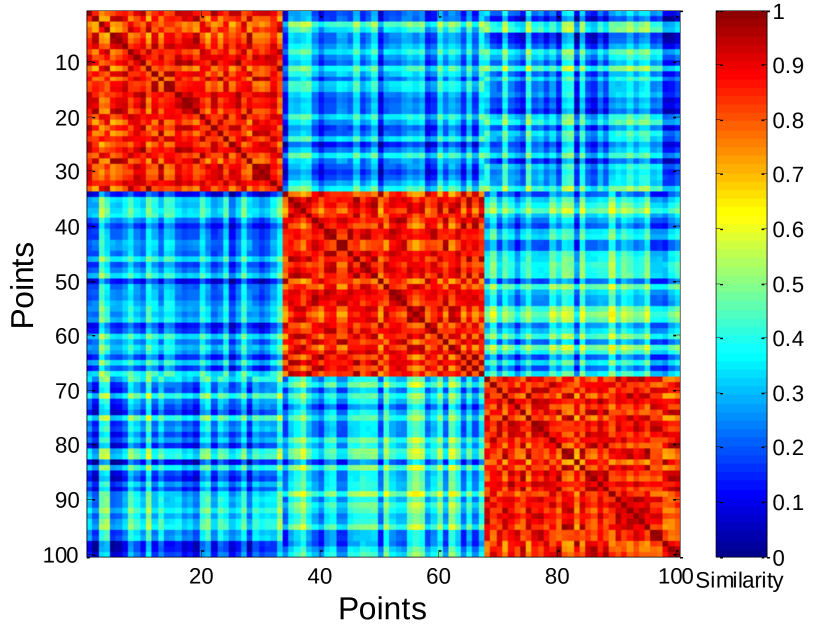
Corr = 0.9235

Corr = 0.5810

Using Similarity Matrix for Cluster Validation

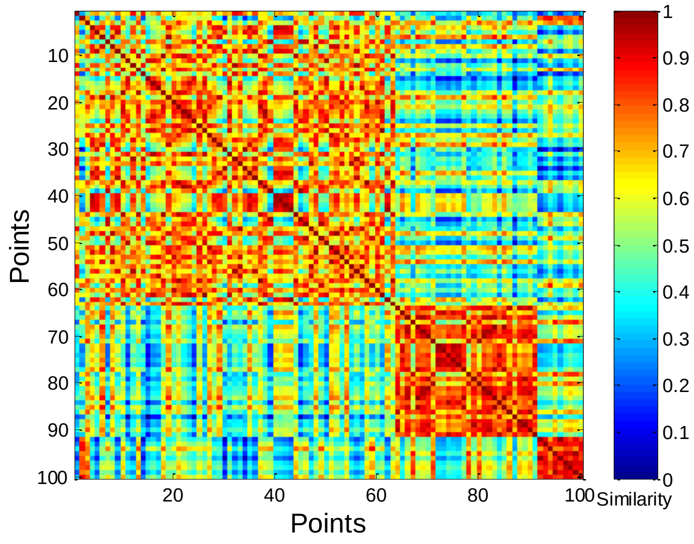
- Order the similarity matrix with respect to cluster labels and inspect visually.

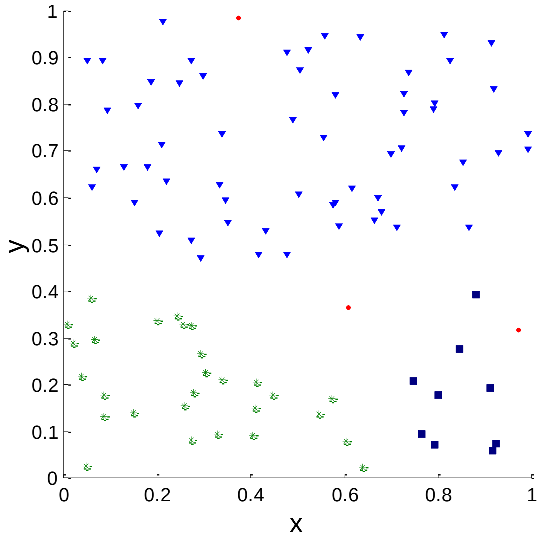




Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

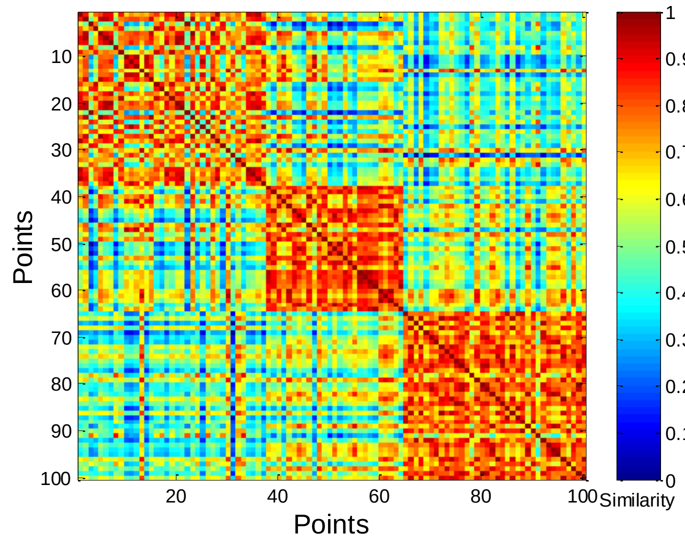


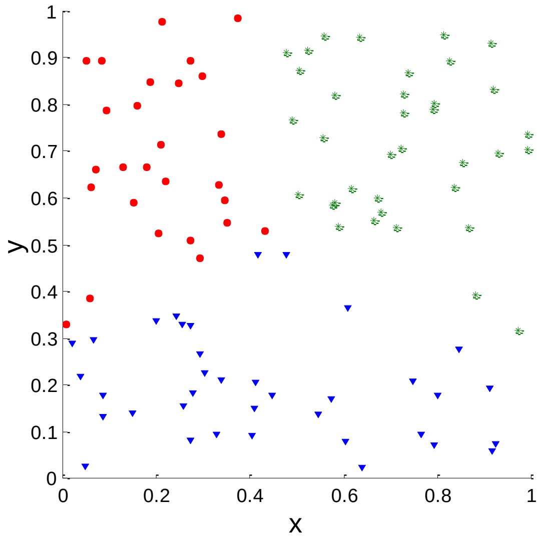


DBSCAN

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

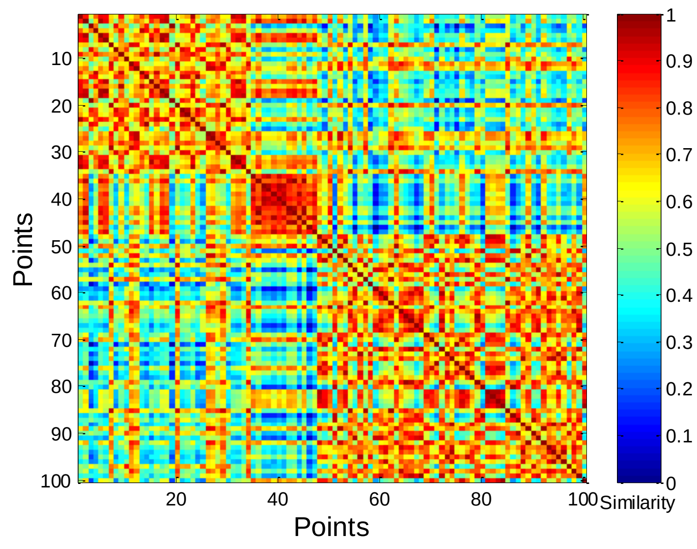


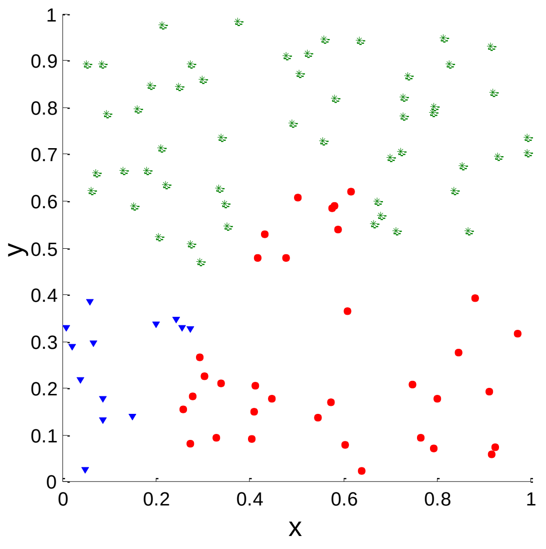


K-means

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp





Complete Link

Fuzzy Clustering

Fuzzy Clustering

- ❖ *data objects are distributed in well-separated groups, disjoint clusters seems like an ideal approach.*
- ❖ *in most cases, the objects in a data set cannot be partitioned into well-separated clusters*
- ❖ *an object that lies near the boundary of two clusters*
- ❖ assign a weight to each object and each cluster
- ❖ w_{ij} is the weight with which object $\mathbf{x}_i$ belongs to cluster C_j
  - ✓ *Fuzzy Approach*
  - ✓ *Probabilistic Approach*

Fuzzy clustering

$\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$

A collection of fuzzy clusters, C1, C2, . . ., *Ck* is a subset of all possible fuzzy subsets of

fuzzy psuedo-partition

1. All the weights for a given point, $\mathbf{x}_i$, add up to 1.
   $$\sum_{j=1}^{k} w_{ij} = 1$$

2. Each cluster, C_j, contains, with non-zero weight, at least one point, but does not contain, with a weight of one, all of the points.
   $$0 < \sum_{i=1}^{m} w_{ij} < m$$

Fuzzy c-means

Algorithm 9.1 Basic fuzzy c-means algorithm.

```
1: Select an initial fuzzy pseudo-partition, i.e., assign values to all the $w_{ij}$.
2: repeat
3:     Compute the centroid of each cluster using the fuzzy pseudo-partition.
4:     Recompute the fuzzy pseudo-partition, i.e., the $w_{ij}$.
5: until The centroids don't change.
   (Alternative stopping conditions are "if the change in the error is below a specified
   threshold" or "if the absolute change in any $w_{ij}$ is below a given threshold.")
```

Fuzzy c-means

- ✓ As with K-means, FCM can be interpreted as attempting to minimize the sum of the squared error (SSE)
- ✓ FCM is based on a fuzzy version of SSE
- ✓ K-means regarded as a special case of FCM

$$\mathrm{SSE}(C_1, C_2, \ldots, C_k) = \sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^{p}\, dist(\mathbf{x}_i, \mathbf{c}_j)^2$$

To minimize objective function, repeat the following:

Fix $\boldsymbol{c_j}$ and determine w_{ij}

Fix w_{ij} and recompute $\boldsymbol{c}$

Fuzzy c-means

Computing Centroids

$$c_j = \sum_{i=1}^{m} w_{ij}^p \mathbf{x}_i / \sum_{i=1}^{m} w_{ij}^p$$

Updating the Fuzzy Pseudo-partition

weight update formula can be derived by minimizing the SSE subject to the constraint that the weights sum to 1

$$w_{ij} = (1/dist(\mathbf{x}_i, \mathbf{c}_j)^2)^{\frac{1}{p-1}} \Big/ \sum_{q=1}^{k} (1/dist(\mathbf{x}_i, \mathbf{c}_q)^2)^{\frac{1}{p-1}}$$

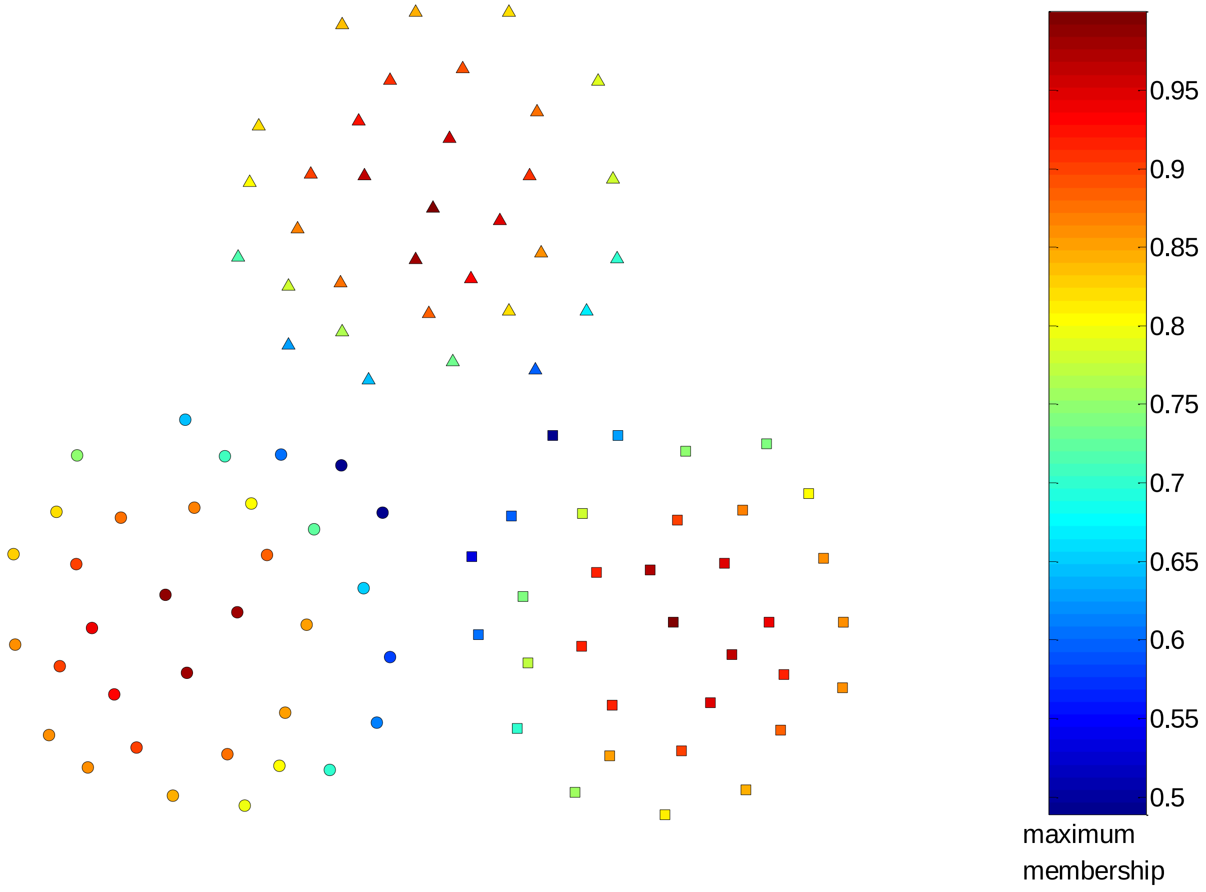
$$w_{ij} = 1/dist(\mathbf{x}_i, \mathbf{c}_j)^2 \Big/ \sum_{q=1}^{k} 1/dist(\mathbf{x}_i, \mathbf{c}_q)^2$$

Weights and p

$$w_{ij} = \left(1/dist(\mathbf{x}_i, \mathbf{c}_j)^2\right)^{\frac{1}{p-1}} \Big/ \sum_{q=1}^{k} \left(1/dist(\mathbf{x}_i, \mathbf{c}_q)^2\right)^{\frac{1}{p-1}}$$

- as p goes to infinity, the exponent tends to 0 and weights tend to the value 1/k.
- As *p* goes to 1, the membership weight goes to 1 for the closest cluster and to 0 for all the other clusters.

Fuzzy K-means Applied to Sample Data



Clustering Using Mixture Models

clustering based on statistical models

- assume that data has been generated as a result of a statistical process
- describe the data by finding the statistical model that best fits the data
- deciding on a statistical model for the data
- Estimating the parameters of that model from the data
- one distribution corresponds to a cluster
- parameters of each distribution a description of that cluster
- **mixture models,** which model the data by using a number of statistical distributions

Mixture Models

Mixture models view the data as a set of observations from a mixture of different probability distributions

1. Given several distributions, usually of the same type, but with different parameters
2. randomly select one of these distributions
3. Generate an object from it
4. Repeat the process *m* times, where *m* is the number of objects.

assume that there are *K* distributions and *m* objects, *jth* distribution have parameters *θj*

$$\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \qquad \Theta = \{\theta_1, \ldots, \theta_K\}$$

$$prob(\mathbf{x}_i|\Theta) = \sum_{j=1}^{K} w_j p_j(\mathbf{x}_i|\theta_j)$$

Gaussian Mixture

✓We cannot use one single Gaussian distribution to model real data sets.

✓A linear combination of Gaussians can give rise to very complex densities densities.

✓By using a sufficient number of Gaussians, and by adjusting their parameters as well as the coefficients in the linear combination, almost any continuous density can be approximated.

Gaussian Mixture

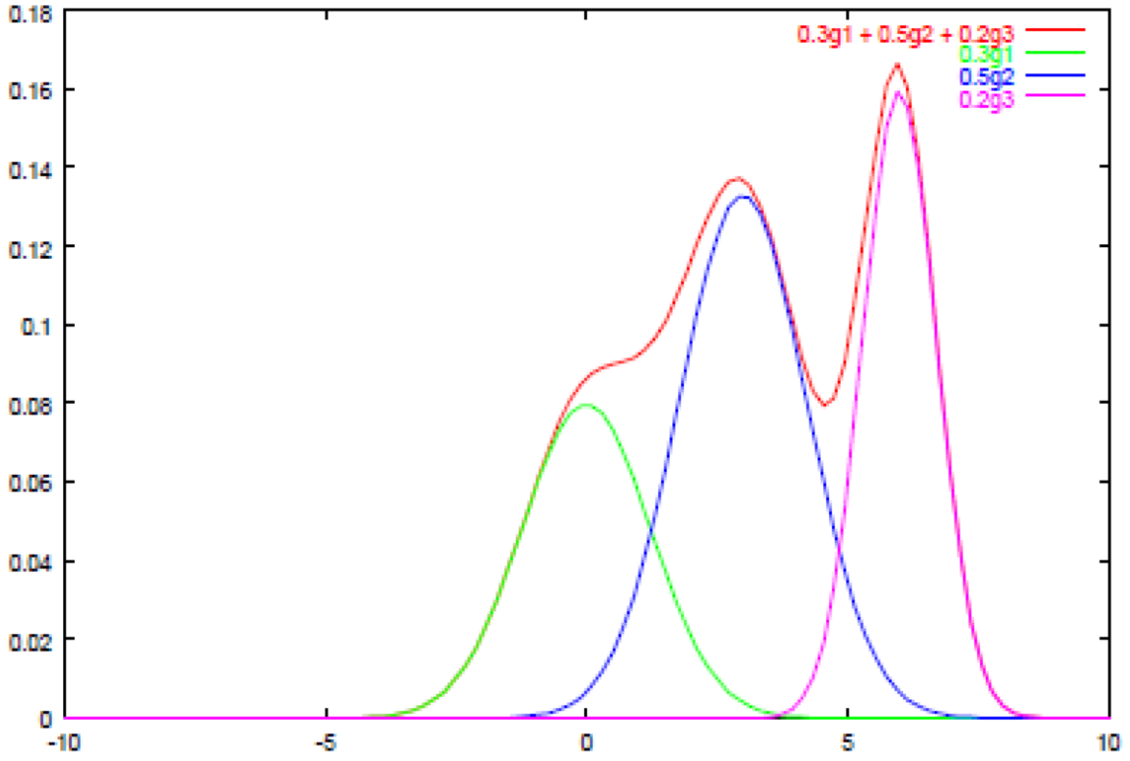
✓Let’s consider a random process x_i that follows a Gaussian mixture distribution with M component

$$x_i \sim GM(w, \mu, \sigma)$$

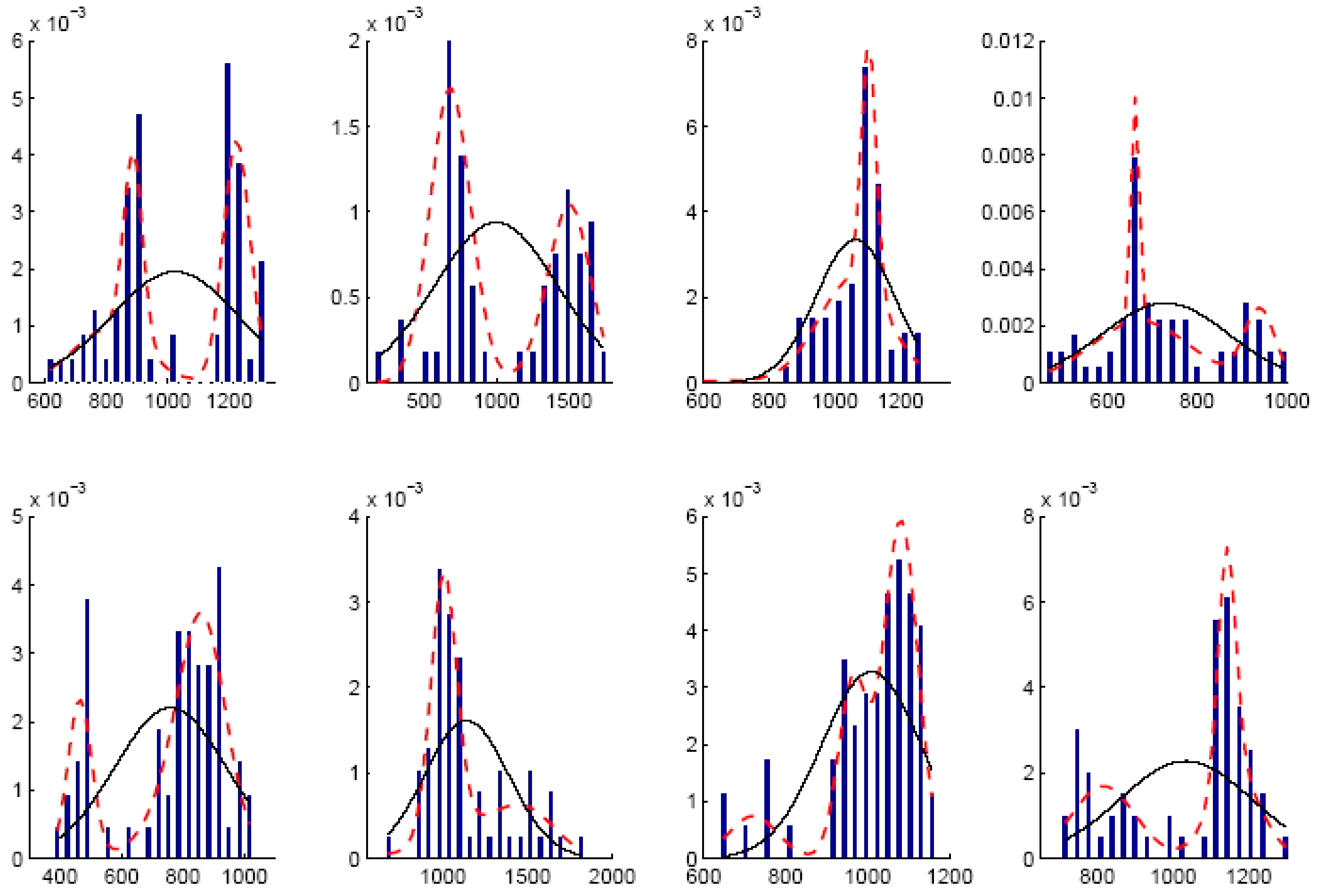
$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{pmatrix} \quad \sigma = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_M \end{pmatrix}, \quad w = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{pmatrix}$$

$$\boxed{f_{x_i}(x_i) = \sum_{j=1}^{M} \frac{w_j}{\sqrt{2\pi}\sigma_j} \exp \frac{-(x_i - m_j)^2}{2\sigma_j^2}.}$$

Gaussian Mixture



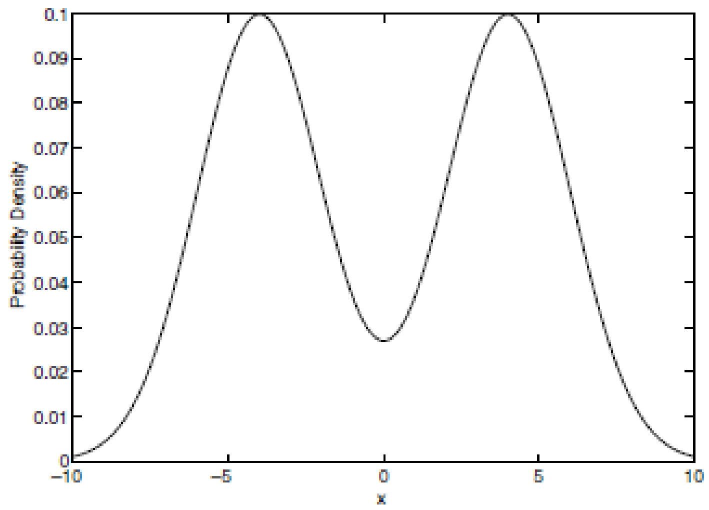
Gaussian Mixture

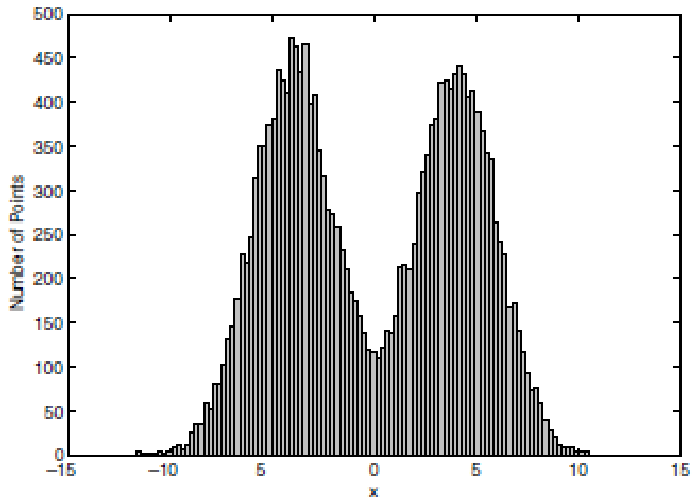


example

$$prob(x_i|\Theta) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$prob(x|\Theta) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x+4)^2}{8}} + \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x-4)^2}{8}}$$





(b) 20,000 points generated from the mixture model.

How estimate the parameters of these distributions from the data and find (clusters)

Mixture Models

objects are generated in an independent manner

$$prob(\mathcal{X}|\Theta) = \prod_{i=1}^{m} prob(\mathbf{x}_i|\Theta) = \prod_{i=1}^{m}\sum_{j=1}^{K} w_j p_j(\mathbf{x}_i|\theta_j)$$

each distribution describes a different cluster

Estimating Model Parameters Using Maximum Likelihood

a set of *m* points that are generated from a one dimensional Gaussian distribution

$$prob(\mathcal{X}|\Theta) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-u)^2}{2\sigma^2}}$$

estimate *u* and *σ*

$$log\ prob(\mathcal{X}|\Theta) = -\sum_{i=1}^{m} \frac{(x_i-u)^2}{2\sigma^2} - 0.5m\log 2\pi - m\log\sigma$$

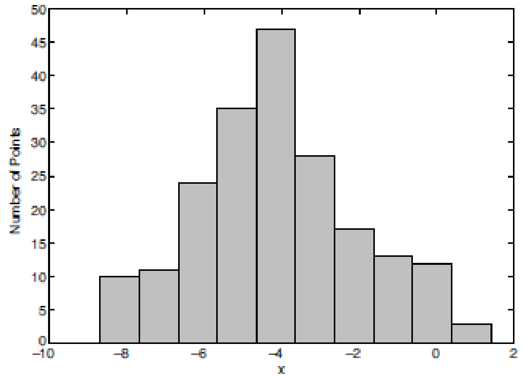
choose the values of the parameters for which the data is most probable (most likely): **maximum likelihood estimation (MLE)**

$$likelihood(\Theta|\mathcal{X}) = L(\Theta|\mathcal{X}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

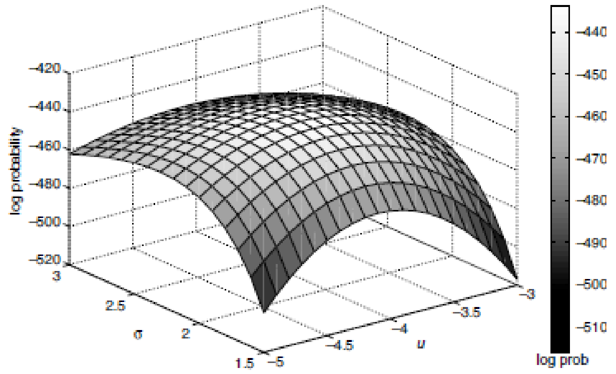
$$log\ likelihood(\Theta|\mathcal{X}) = \ell(\Theta|\mathcal{X}) = -\sum_{i=1}^{m} \frac{(x_i-\mu)^2}{2\sigma^2} - 0.5m\log 2\pi - m\log\sigma$$

Example

Finding the parameters of Gaussian Distribution



(a) Histogram of 200 points from a Gaussian distribution.



(b) Log likelihood plot of the 200 points for different values of the mean and standard deviation.

$$u = -4.1 \text{ and } \sigma = 2.1$$

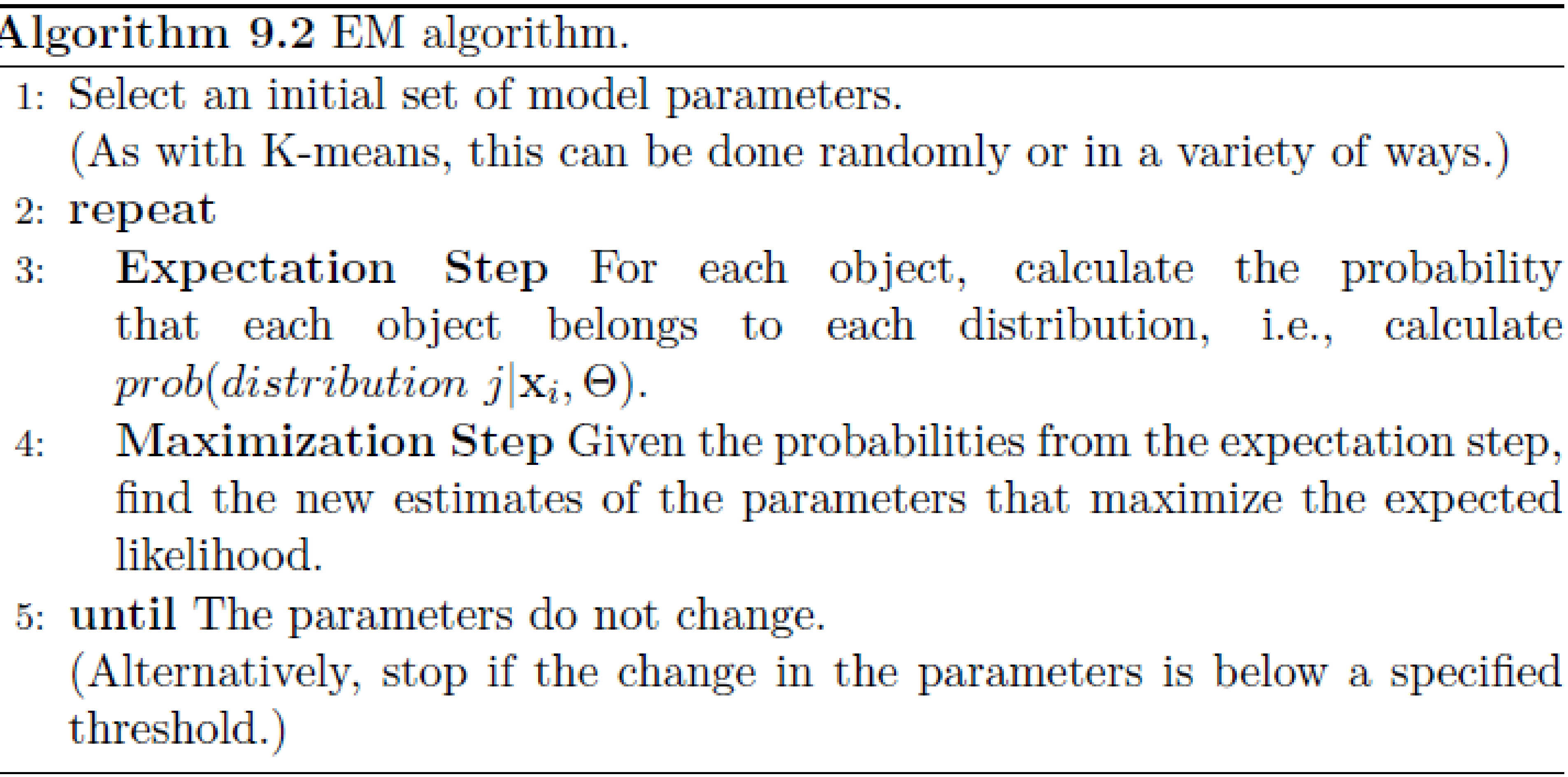
Graphing the likelihood of the data for different values of the parameters is not practical: Optimization
for a Gaussian distribution, it can be shown that the mean and standard deviation of the sample points are the maximum likelihood estimates of the corresponding parameters

Estimating Mixture Model Parameters Using Maximum Likelihood

- simplest case:we know which data objects come from which distributions
- For most common distributions, the maximum likelihood estimates of the parameters are calculated from simple formulas involving the data.
- In a more general (and more realistic) situation, we do not know which points were generated by which distribution
- solution is the EM algorithm

1. given a guess for the parameter values, the EM algorithm calculates probability that each point belongs to each distribution
2. uses these probabilities to compute a new estimate for the parameters

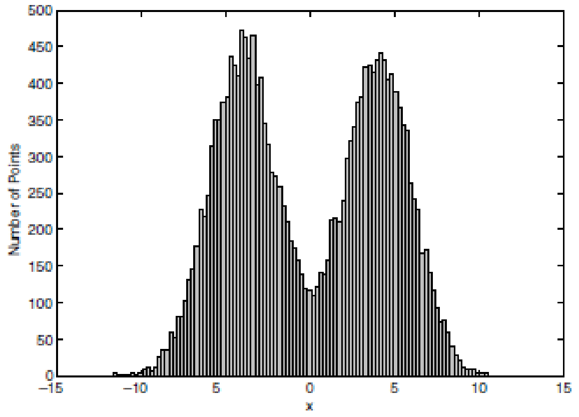
EM algorithm



Algorithm 9.2 EM algorithm.

```
1: Select an initial set of model parameters.
   (As with K-means, this can be done randomly or in a variety of ways.)
2: repeat
3:     Expectation Step For each object, calculate the probability
       that each object belongs to each distribution, i.e., calculate
       $prob(distribution\ j|\mathbf{x}_i, \Theta)$.
4:     Maximization Step Given the probabilities from the expectation step,
       find the new estimates of the parameters that maximize the expected
       likelihood.
5: until The parameters do not change.
   (Alternatively, stop if the change in the parameters is below a specified
   threshold.)
```

example



(b) 20,000 points generated from the mixture model.

- ✓ assume that we know that the standard deviation of both distributions is 2.0
- ✓ points were generated with equal probability from both distributions

guess $\mu_1 = -2$ and $\mu_2 = 3$

$$\theta_1 = (-2, 2) \text{ and } \theta_2 = (3, 2) \quad \Theta = \{\theta_1, \theta_2\}$$

$$prob(distribution\ j|x_i, \theta) = \frac{0.5\ prob(x_i|\theta_j)}{0.5\ prob(x_i|\theta_1) + 0.5\ prob(x_i|\theta_2)},$$

$$prob(0|\theta_1) = 0.12 \qquad prob(0|\theta_2) = 0.06$$

$$prob(distribution\ 1|0, \Theta) = 0.12/(0.12 + 0.06) = 0.66$$

example

After computing the cluster membership probabilities for all 20,000 points,
we compute new estimates for μ_1 and μ_2

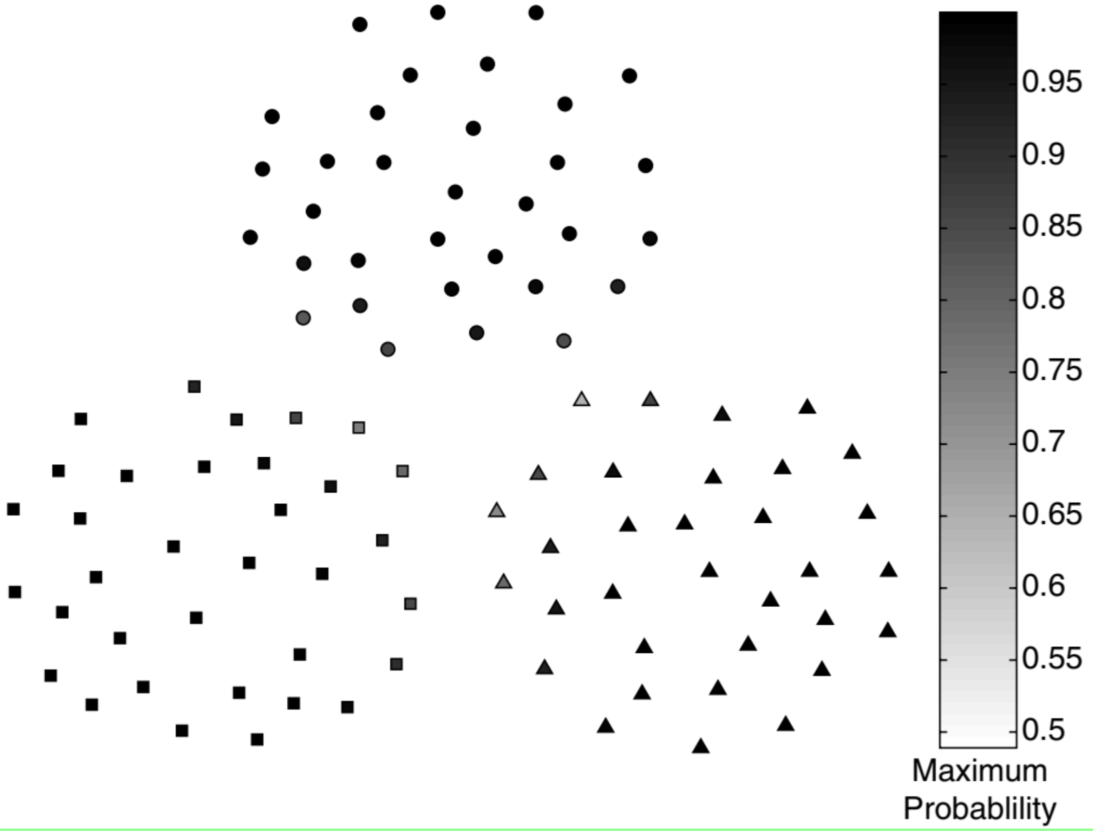
new estimate for the mean of a distribution is just a weighted average of the points

$$\mu_1 = \sum_{i=1}^{20,000} x_i \frac{prob(distribution\ 1|x_i, \Theta)}{\sum_{i=1}^{20,000} prob(distribution\ 1|x_i, \Theta)}$$

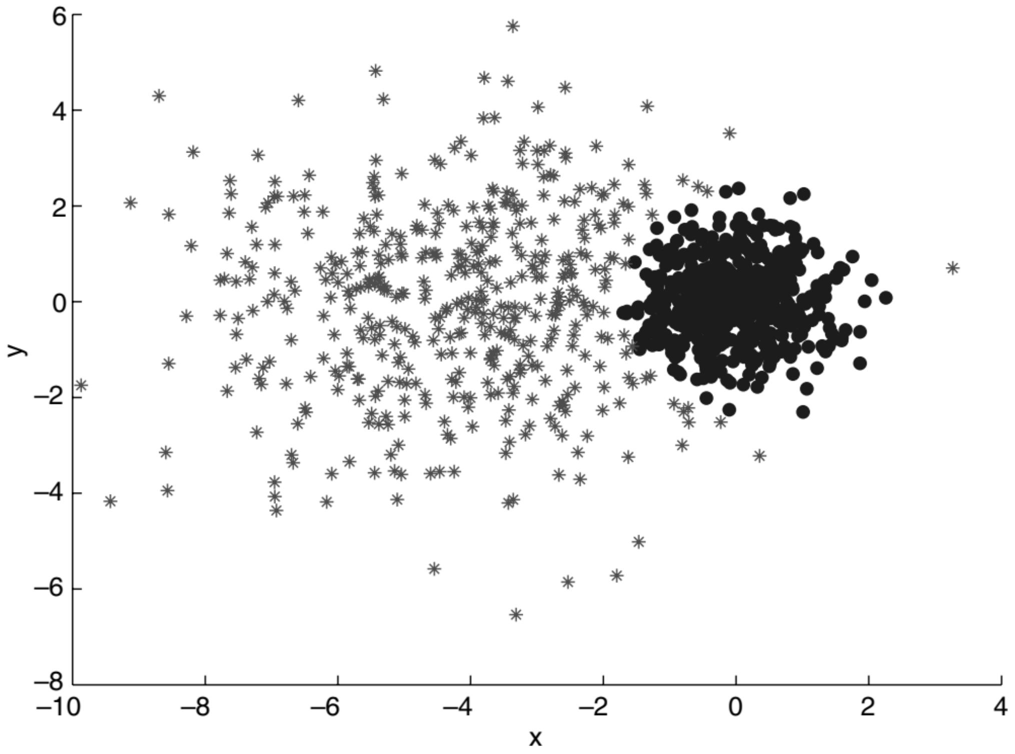
$$\mu_2 = \sum_{i=1}^{20,000} x_i \frac{prob(distribution\ 2|x_i, \Theta)}{\sum_{i=1}^{20,000} prob(distribution\ 2|x_i, \Theta)}$$

Iteration	μ_1	μ_2
0	-2.00	3.00
1	-3.74	4.10
2	-3.94	4.07
3	-3.97	4.04
4	-3.98	4.03
5	-3.98	4.03

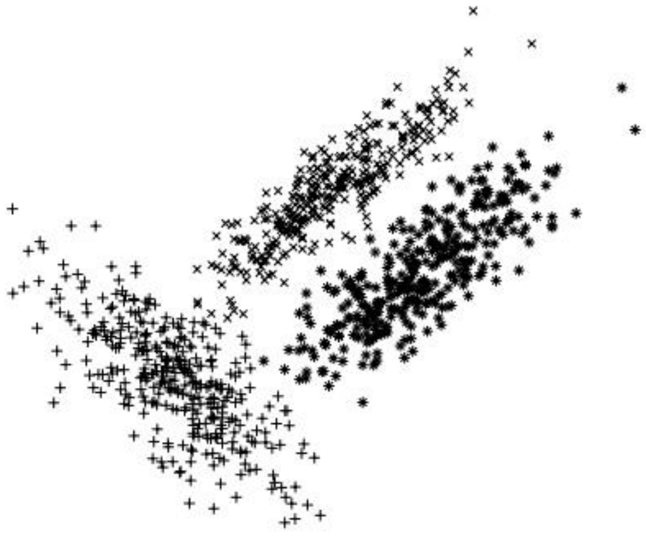
EM Algorithm on Sample Data Sets

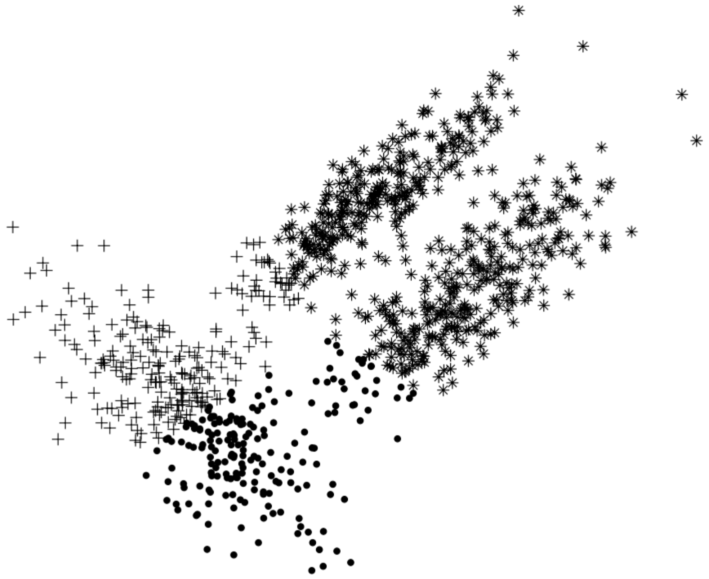


- We modeled this data as a mixture of three two-dimensional Gaussian distributions with different means and identical covariance matrices.
- Each point was assigned to the cluster in which it had the largest membership weigh



EM clustering of a two-dimensional point set with two clusters of differing density


(a) Clusters produced by mixture model clustering.


(b) Clusters produced by K-means clustering.