

# DATA MINING

## LECTURE 8

---

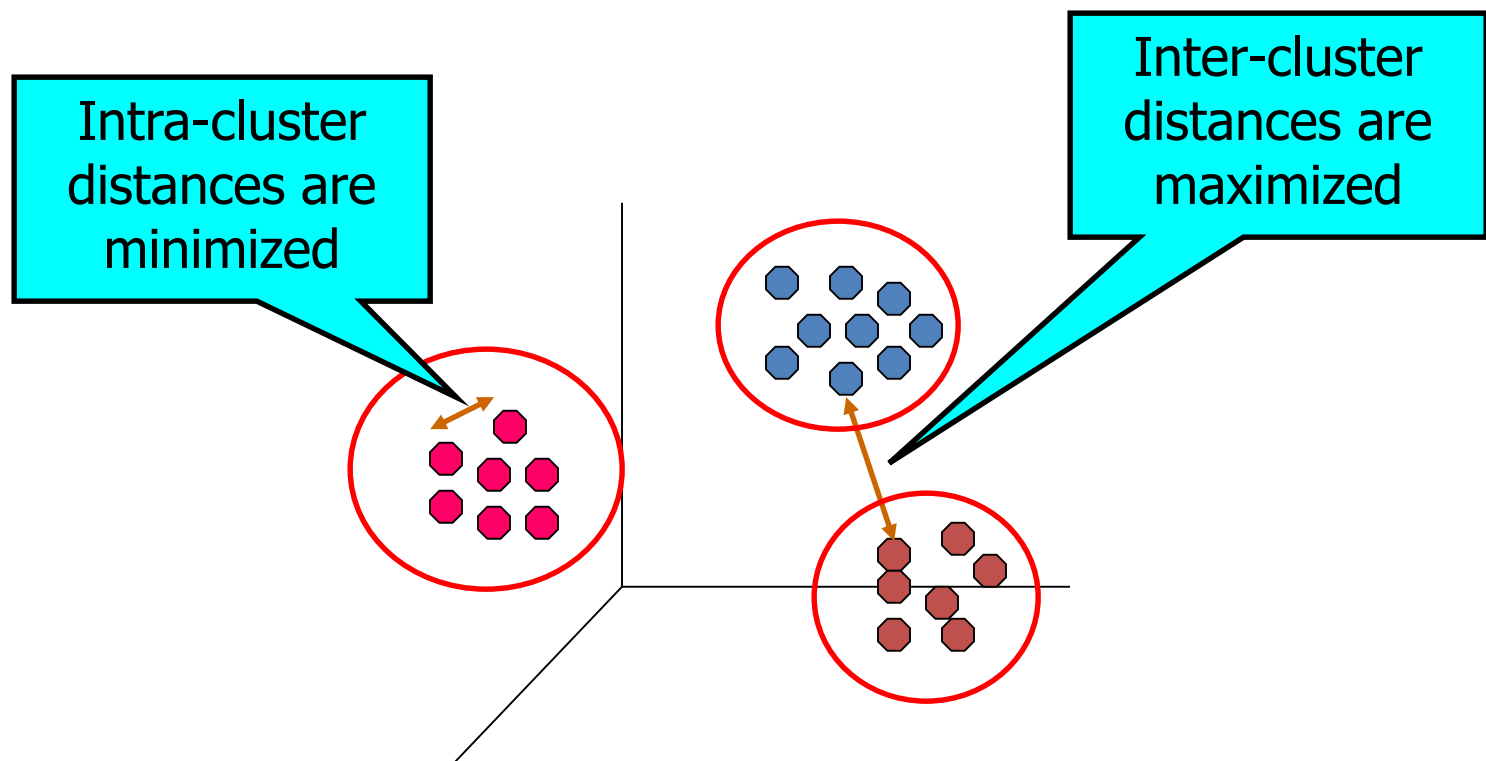
Clustering Evaluation  
The EM Algorithm

# CLUSTERING

---

# What is a Clustering?

- In general a **grouping** of objects such that the objects in a **group** (**cluster**) are similar (or related) to one another and different from (or unrelated to) the objects in other groups



# Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- DBSCAN

# CLUSTERING EVALUATION

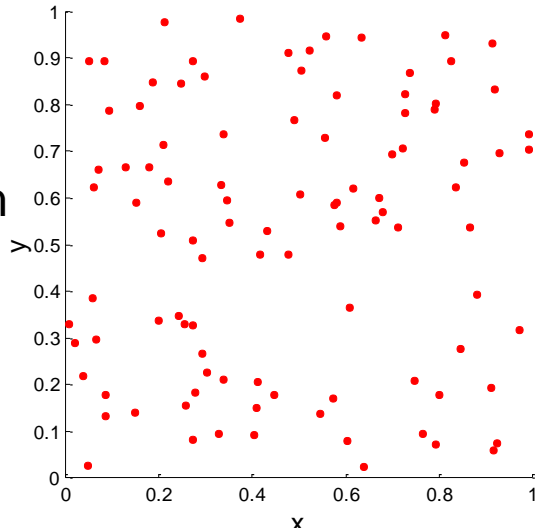
---

# Clustering Evaluation

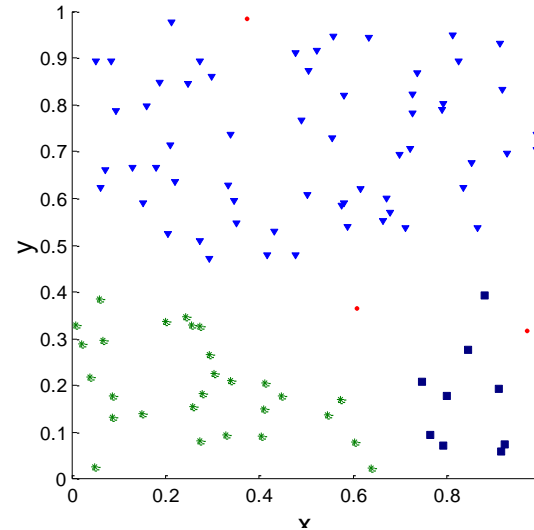
- We need to evaluate the “goodness” of the resulting clusters?
- But “clustering lies in the eye of the beholder”!
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clusterings, or clustering algorithms
  - To compare against a “ground truth”

# Clusters found in Random Data

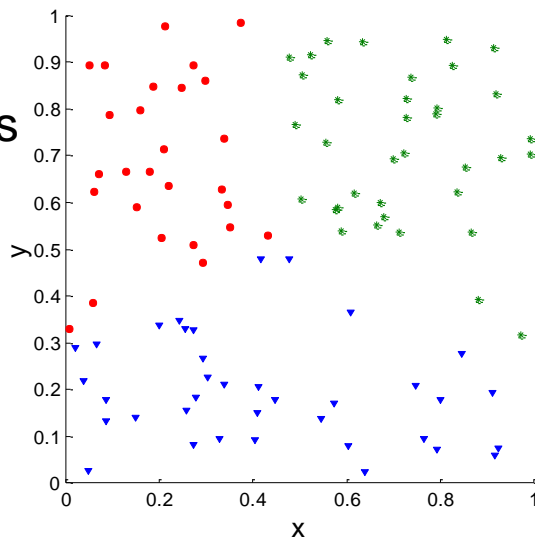
Random  
Points



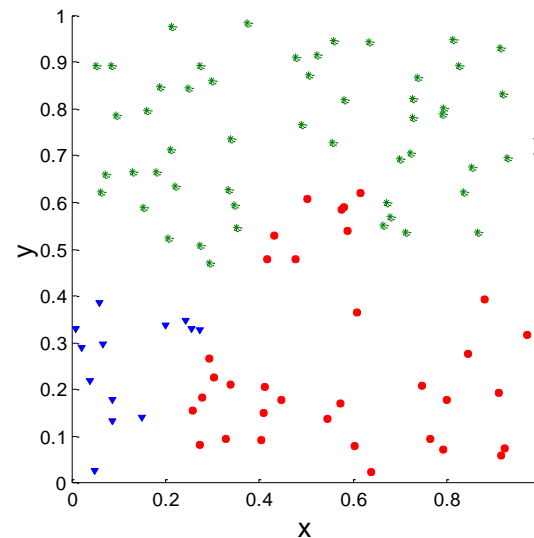
DBSCAN



K-means



Complete  
Link



# Different Aspects of Cluster Validation

1. Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given **class labels**.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
  - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the '**correct**' **number of clusters**.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.



# Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - **External Index:** Used to measure the extent to which cluster labels match **externally supplied class labels**.
    - E.g., entropy, precision, recall
  - **Internal Index:** Used to measure the goodness of a clustering structure **without** reference to external information.
    - E.g., Sum of Squared Error (SSE)
  - **Relative Index:** Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as **criteria** instead of **indices**
  - However, sometimes criterion is the **general strategy** and index is the **numerical measure** that implements the criterion.

# Measuring Cluster Validity Via Correlation

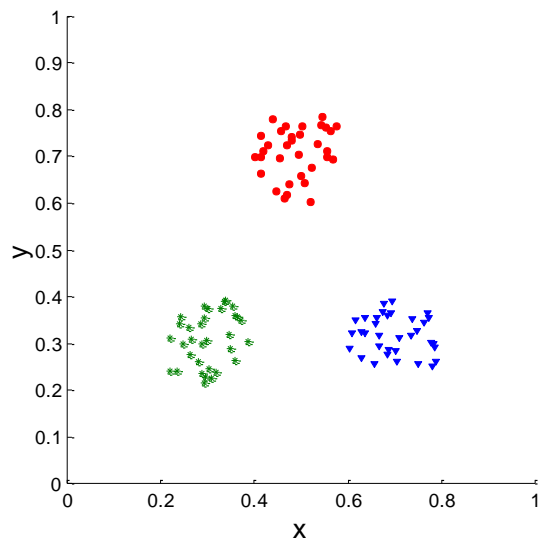
- Two matrices
  - **Similarity** or **Distance** Matrix
    - One row and one column for each data point
    - An entry is the similarity or distance of the associated pair of points
  - **“Incidence” Matrix**
    - One row and one column for each data point
    - An entry is 1 if the associated pair 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.

$$\text{CorrCoeff}(X, Y) = \frac{\sum_i (x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_i (x_i - \mu_X)^2} \sqrt{\sum_i (y_i - \mu_Y)^2}}$$

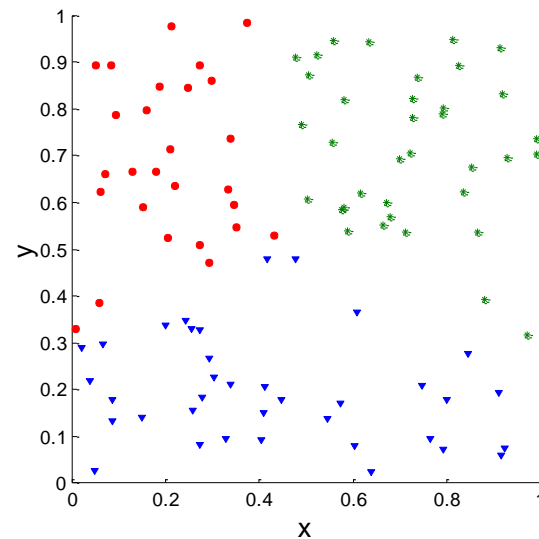
- **High** correlation (**positive** for similarity, **negative** for distance) indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

# Measuring Cluster Validity Via Correlation

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



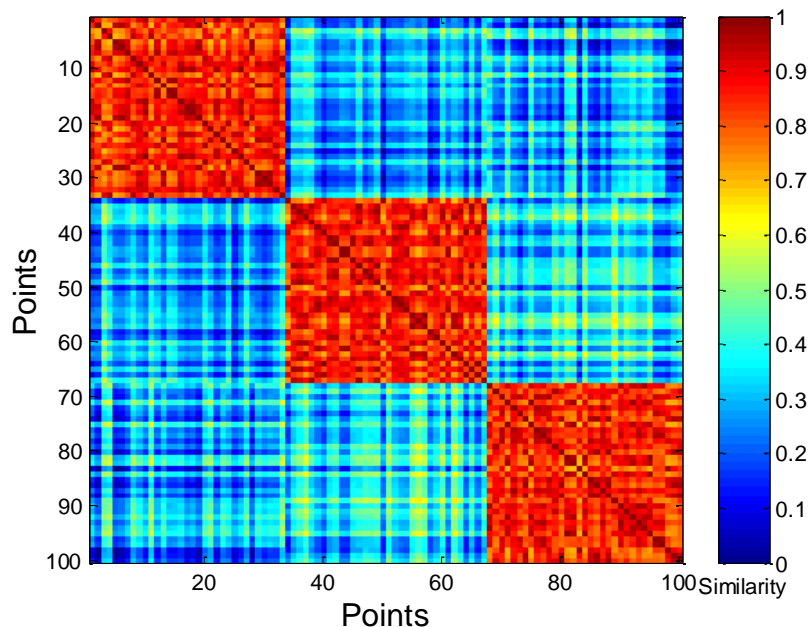
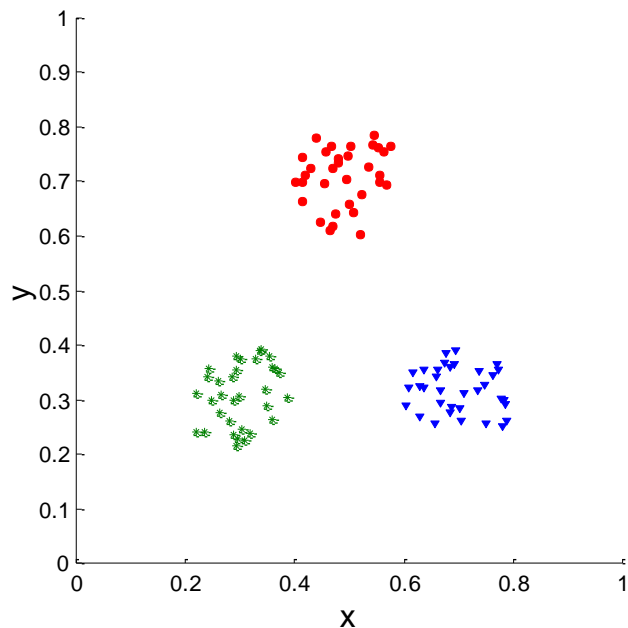
**Corr = -0.9235**



**Corr = -0.5810**

# Using Similarity Matrix for Cluster Validation

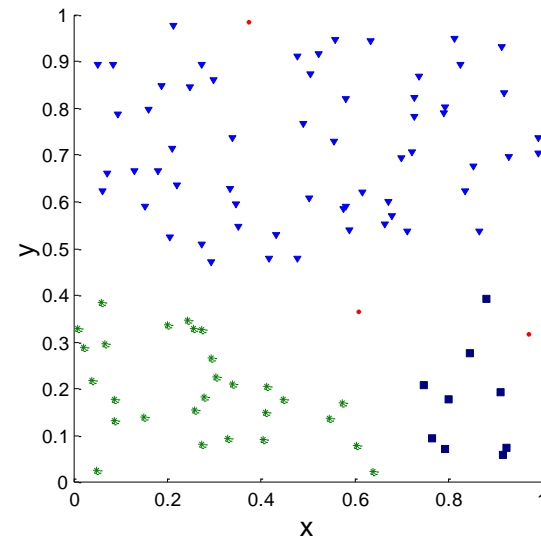
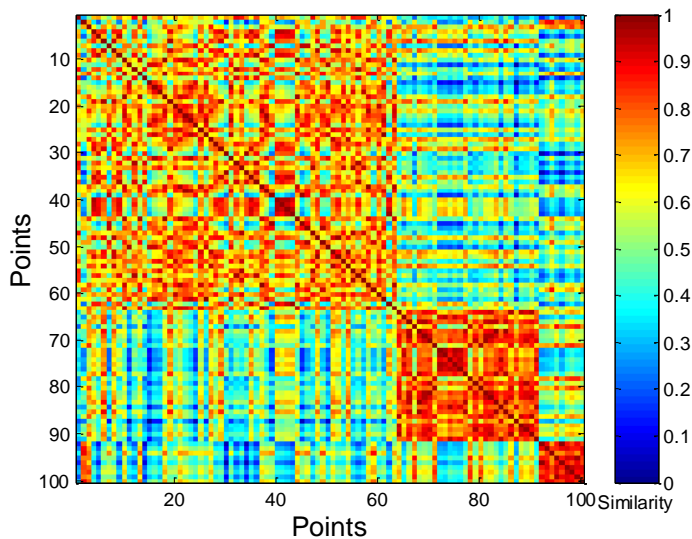
- Order the **similarity** matrix with respect to cluster labels and inspect visually.



$$sim(i,j) = 1 - \frac{d_{ij} - d_{min}}{d_{max} - d_{min}}$$

# Using Similarity Matrix for Cluster Validation

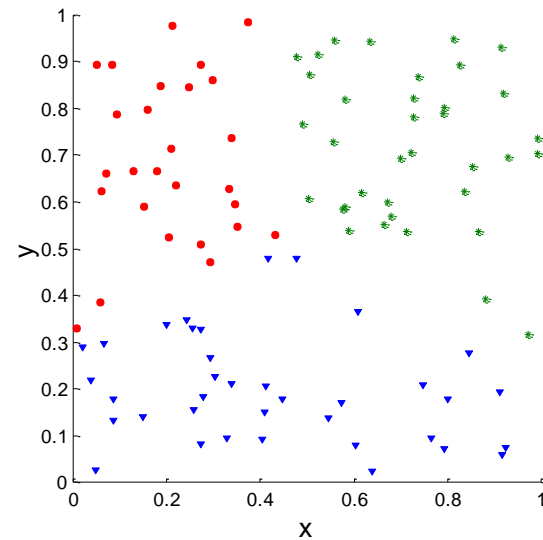
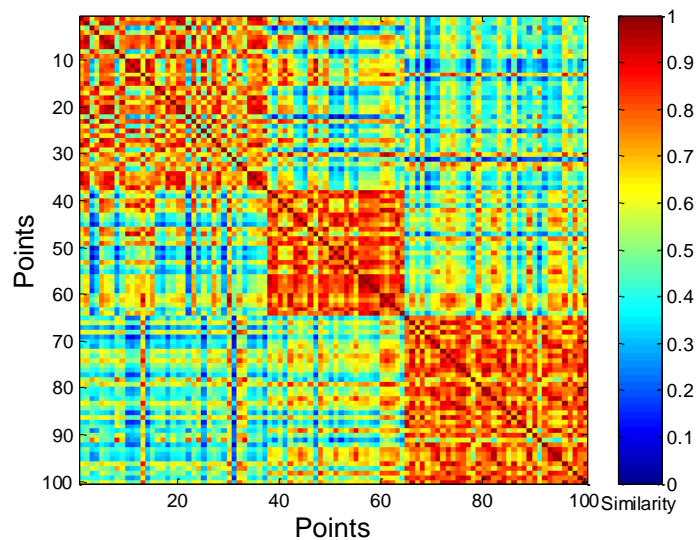
- Clusters in random data are not so crisp



DBSCAN

# Using Similarity Matrix for Cluster Validation

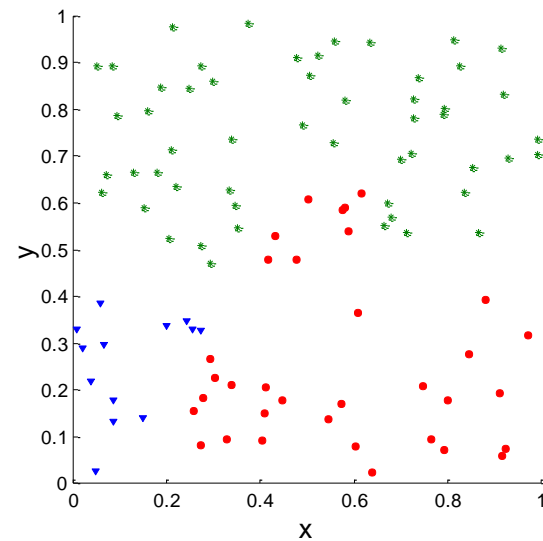
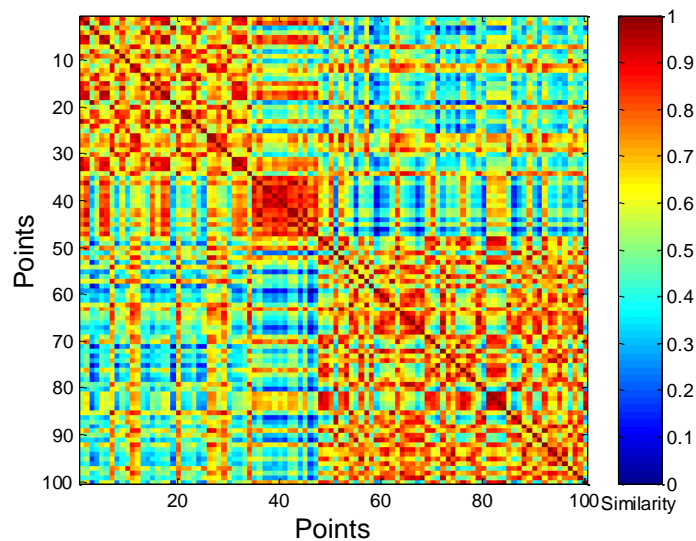
- Clusters in random data are not so crisp



K-means

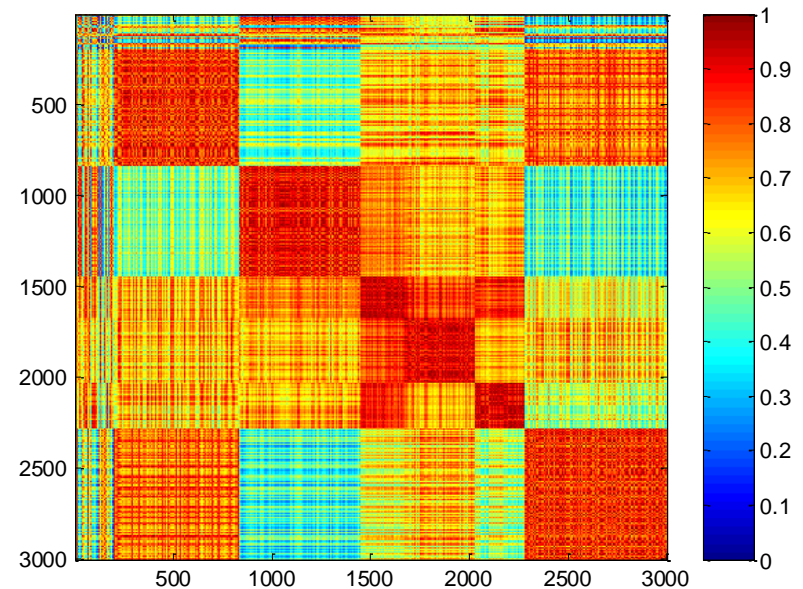
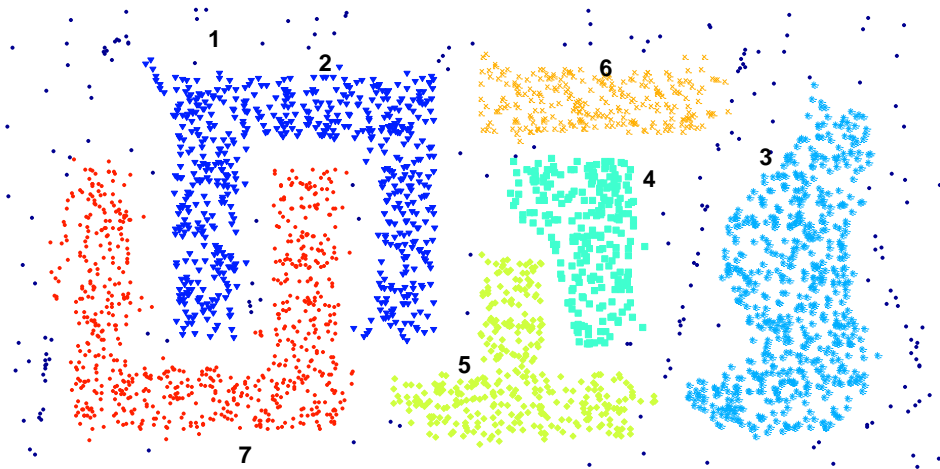
# Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp



Complete Link

# Using Similarity Matrix for Cluster Validation



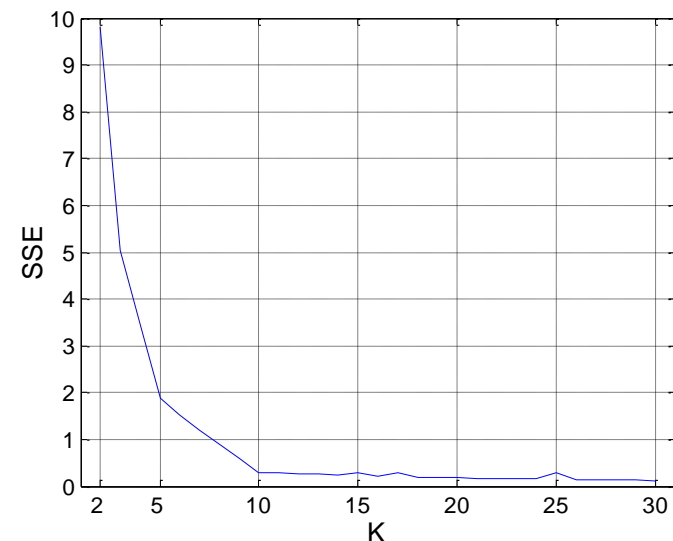
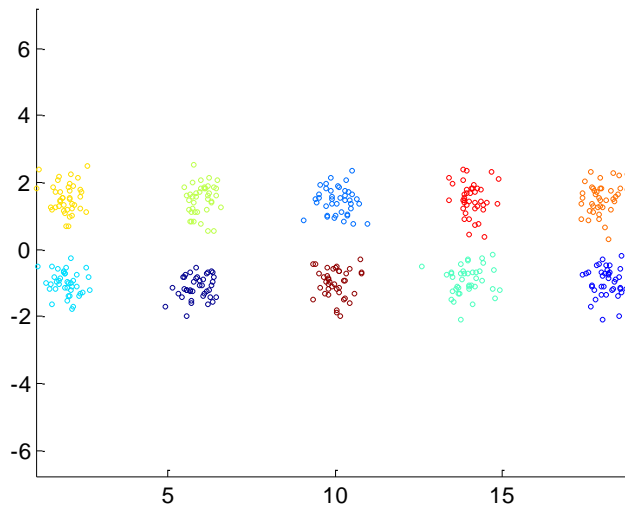
DBSCAN

- Clusters in more complicated figures are not well separated
- This technique can only be used for small datasets since it requires a quadratic computation



# Internal Measures: SSE

- **Internal Index:** Used to measure the goodness of a clustering structure without reference to external information
  - Example: SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters



# Internal Measures: Cohesion and Separation

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

- Cohesion is measured by the **within cluster sum of squares** (SSE)

$$WSS = \sum_i \sum_{x \in C_i} (x - c_i)^2$$

We want this to be small

- Separation is measured by the **between cluster sum of squares**

$$BSS = \sum_i m_i (c - c_i)^2$$

We want this to be large

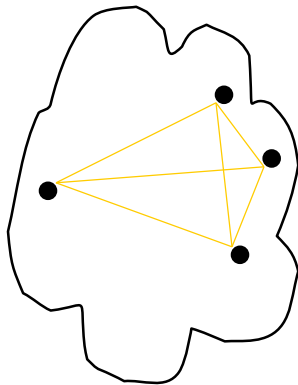
- Where  $m_i$  is the size of cluster  $i$ ,  $c$  the **overall mean**

$$BSS = \sum_{x \in C_i} \sum_{y \in C_j} (x - y)^2$$

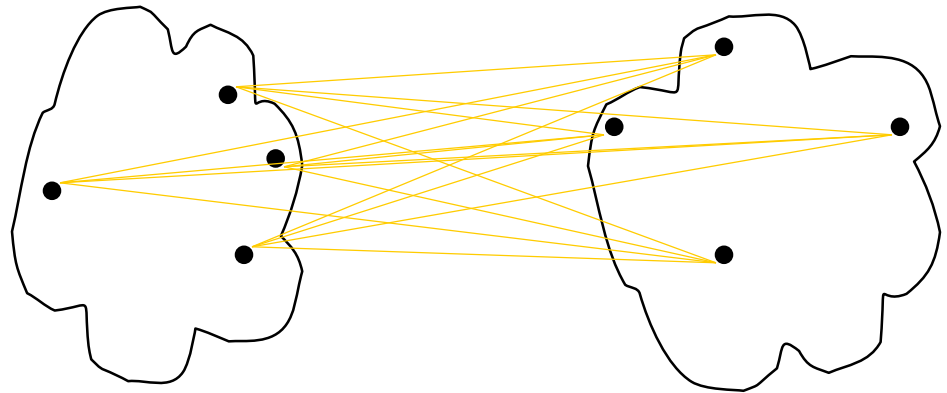
- Interesting observation:  $WSS + BSS = \text{constant}$

# Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion

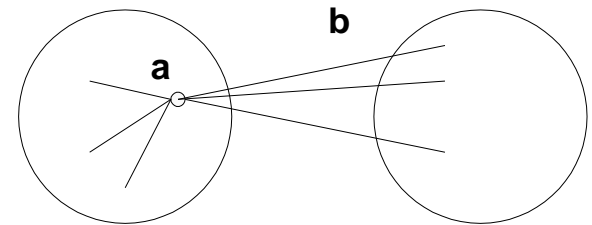


separation

# Internal Measures: Silhouette Coefficient

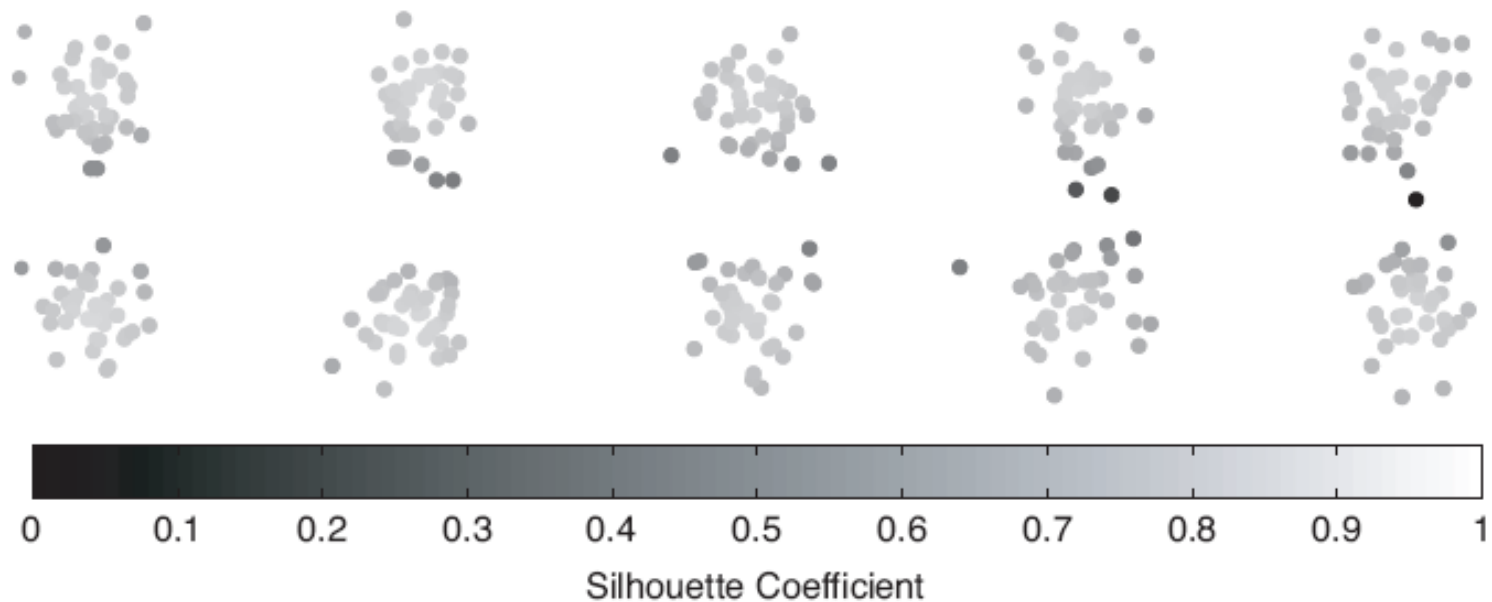
- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point,  $i$ 
  - Calculate  $a$  = average distance of  $i$  to the points in its own cluster
  - Calculate  $b$  = min (over clusters) of the average distance of  $i$  to points in other clusters
  - The silhouette coefficient for a point is then given by

$$s = 1 - a/b$$



- Typically between 0 and 1.
  - The closer to 1 the better.
  - Can be less than 0 but this is a problematic case
- Can calculate the Average Silhouette coefficient for a cluster, or for a clustering

# Silhouette Coefficient Example



**Figure 8.29.** Silhouette coefficients for points in ten clusters.

# Internal measures – caveats

- Internal measures have the problem that the clustering algorithm **did not set out to optimize this measure**, so it is will not necessarily do well with respect to the measure.
- An internal measure can also be used as an objective function for clustering

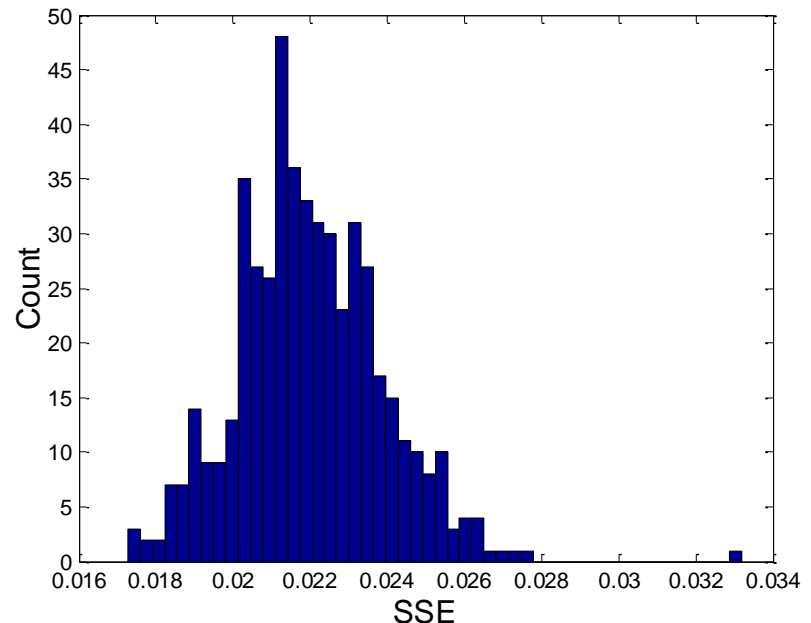
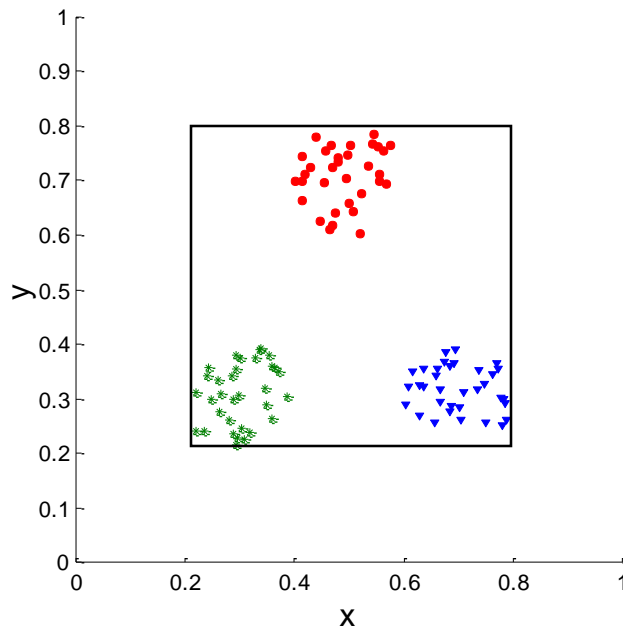
# Framework for Cluster Validity

- Need a **framework** to interpret any measure.
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- **Statistics** provide a framework for cluster validity
  - The more “**non-random**” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from **random** data or clusterings to those of a clustering result.
    - If the value of the index is **unlikely**, then the cluster results are valid
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is **significant**

# Statistical Framework for SSE

- Example

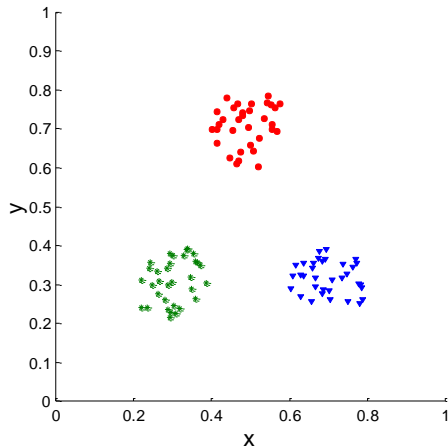
- Compare SSE of 0.005 against three clusters in random data
- Histogram of SSE for three clusters in 500 random data sets of 100 random points distributed in the range 0.2 – 0.8 for x and y
  - Value 0.005 is very unlikely



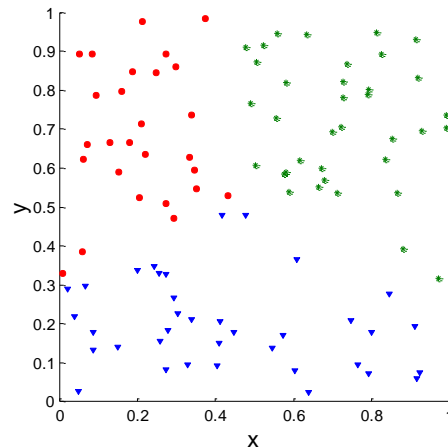


# Statistical Framework for Correlation

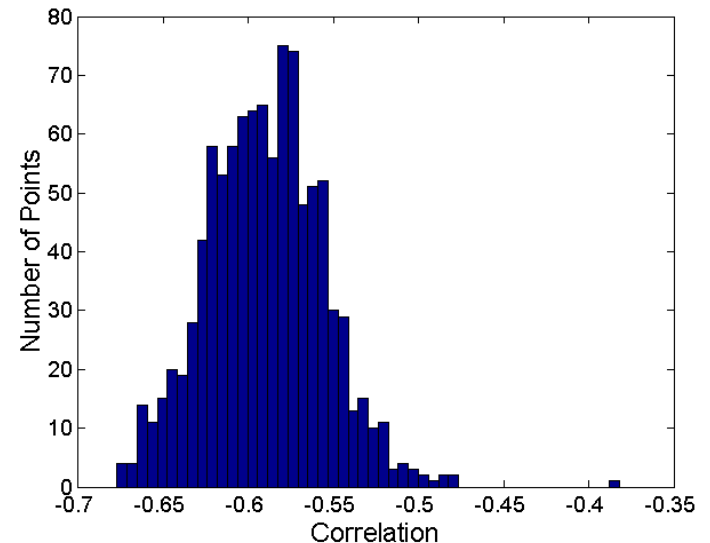
- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



Corr = -0.9235



Corr = -0.5810

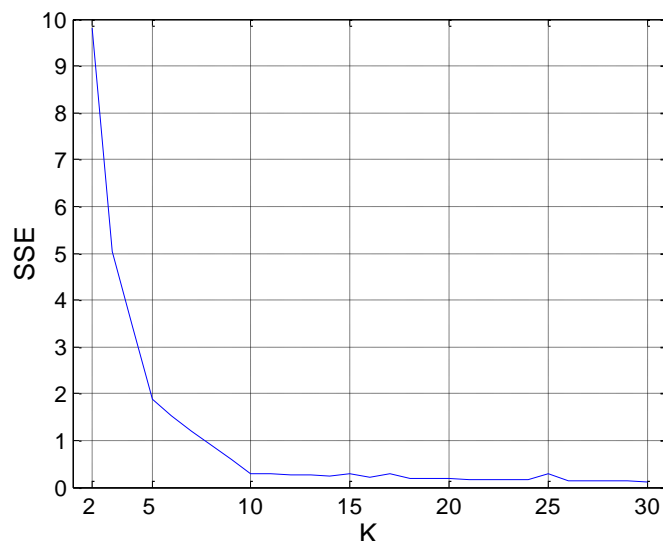


# Empirical p-value

- If we have a **measurement  $v$**  (e.g., the SSE value)
- ..and we have  **$N$**  measurements on **random datasets**
- ...the **empirical p-value** is the **fraction** of measurements in the random data that have value **less or equal** than value  **$v$**  (or greater or equal if we want to maximize)
  - i.e., the value in the random dataset is **at least as good** as that in the real data
- We usually require that  **$p\text{-value} \leq 0.05$**
- **Hard question**: what is the right notion of a random dataset?

# Estimating the “right” number of clusters

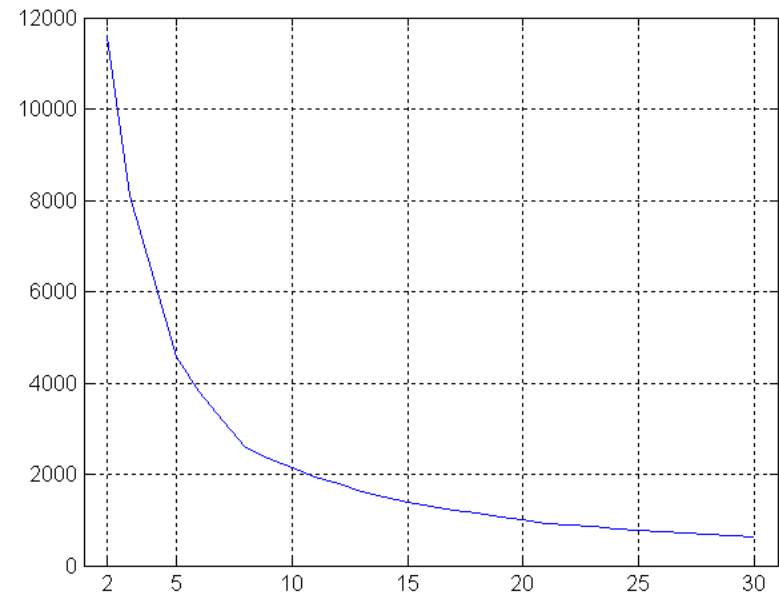
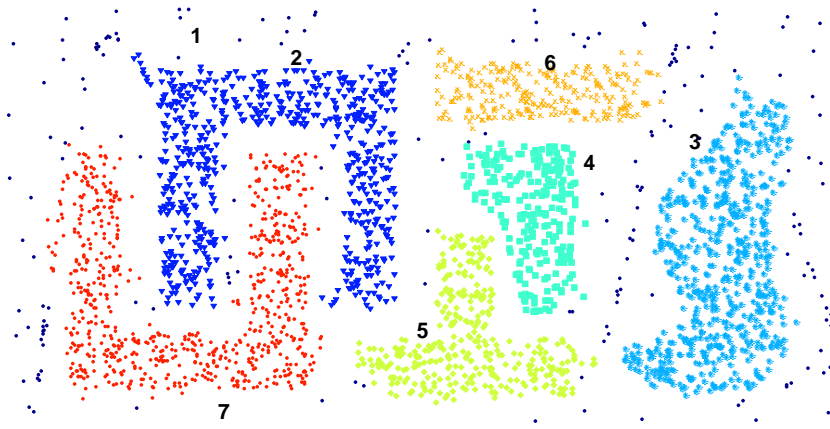
- Typical approach: find a “knee” in an internal measure curve.



- Question: why not the k that **minimizes** the SSE?
  - Forward reference: minimize a measure, but with a “**simple**” clustering
- **Desirable property**: the clustering algorithm does not require the number of clusters to be specified (e.g., DBSCAN)

# Estimating the “right” number of clusters

- SSE curve for a more complicated data set



SSE of clusters found using K-means

# External Measures for Clustering Validity

- Assume that the data is **labeled** with some class labels
  - E.g., **documents** are classified into **topics**, **people** classified according to their **income**, **politicians** classified according to the **political party**.
  - This is called the “**ground truth**”
- In this case we want the clusters to be **homogeneous** with respect to classes
  - **Each cluster** should contain elements of **mostly one class**
  - **Each class** should ideally be assigned to a **single cluster**
- This does not always make sense
  - **Clustering** is not the same as **classification**
  - ...but this is what people use most of the time

# Confusion matrix

- $n$  = number of points
- $m_i$  = points in cluster  $i$
- $c_j$  = points in class  $j$
- $n_{ij}$  = points in cluster  $i$  coming from class  $j$
- $p_{ij} = n_{ij}/m_i$  = probability of element from cluster  $i$  to be assigned in class  $j$

	Class 1	Class 2	Class 3	
Cluster 1	$n_{11}$	$n_{12}$	$n_{13}$	$m_1$
Cluster 2	$n_{21}$	$n_{22}$	$n_{23}$	$m_2$
Cluster 3	$n_{31}$	$n_{32}$	$n_{33}$	$m_3$
	$c_1$	$c_2$	$c_3$	$n$

	Class 1	Class 2	Class 3	
Cluster 1	$p_{11}$	$p_{12}$	$p_{13}$	$m_1$
Cluster 2	$p_{21}$	$p_{22}$	$p_{23}$	$m_2$
Cluster 3	$p_{31}$	$p_{32}$	$p_{33}$	$m_3$
	$c_1$	$c_2$	$c_3$	$n$

# Measures

	Class 1	Class 2	Class 3	
Cluster 1	$p_{11}$	$p_{12}$	$p_{13}$	$m_1$
Cluster 2	$p_{21}$	$p_{22}$	$p_{23}$	$m_2$
Cluster 3	$p_{31}$	$p_{32}$	$p_{33}$	$m_3$
	$c_1$	$c_2$	$c_3$	$n$

- **Entropy:**

- Of a **cluster**  $i$ :  $e_i = -\sum_{j=1}^L p_{ij} \log p_{ij}$ 
  - Highest when uniform, zero when single class
- Of a clustering:  $e = \sum_{i=1}^K \frac{m_i}{n} e_i$

- **Purity:**

- Of a **cluster**  $i$ :  $p_i = \max_j p_{ij}$
- Of a clustering:  $p(C) = \sum_{i=1}^K \frac{m_i}{n} p_i$

# Measures

	Class 1	Class 2	Class 3	
Cluster 1	$p_{11}$	$p_{12}$	$p_{13}$	$m_1$
Cluster 2	$p_{21}$	$p_{22}$	$p_{23}$	$m_2$
Cluster 3	$p_{31}$	$p_{32}$	$p_{33}$	$m_3$
	$c_1$	$c_2$	$c_3$	$n$

- **Precision:**

- Of cluster  $i$  with respect to class  $j$ :  $Prec(i, j) = p_{ij}$

- **Recall:**

- Of cluster  $i$  with respect to class  $j$ :  $Rec(i, j) = \frac{n_{ij}}{c_j}$

- **F-measure:**

- **Harmonic Mean** of Precision and Recall:

$$F(i, j) = \frac{2 * Prec(i, j) * Rec(i, j)}{Prec(i, j) + Rec(i, j)}$$



# Measures

## Precision/Recall for clusters and clusterings

	Class 1	Class 2	Class 3	
Cluster 1	$n_{11}$	$n_{12}$	$n_{13}$	$m_1$
Cluster 2	$n_{21}$	$n_{22}$	$n_{23}$	$m_2$
Cluster 3	$n_{31}$	$n_{32}$	$n_{33}$	$m_3$
	$c_1$	$c_2$	$c_3$	$n$

- Assign to cluster  $i$  the class  $k_i$  such that  $k_i = \arg \max_j n_{ij}$
- **Precision:**
  - Of cluster  $i$ :  $Prec(i) = \frac{n_{ik_i}}{m_i}$
  - Of the clustering:  $Prec(C) = \sum_i \frac{m_i}{n} Prec(i)$
- **Recall:**
  - Of cluster  $i$ :  $Rec(i) = \frac{n_{ik_i}}{c_{k_i}}$
  - Of the clustering:  $Rec(C) = \sum_i \frac{m_i}{n} Rec(i)$
- **F-measure:**
  - **Harmonic Mean** of Precision and Recall

# Good and bad clustering

	Class 1	Class 2	Class 3	
Cluster 1	2	3	85	90
Cluster 2	90	12	8	110
Cluster 3	8	85	7	100
	100	100	100	300

**Purity:** (0.94, 0.81, 0.85)

– overall 0.86

**Precision:** (0.94, 0.81, 0.85)

– overall 0.86

**Recall:** (0.85, 0.9, 0.85)

– overall 0.87

	Class 1	Class 2	Class 3	
Cluster 1	20	35	35	90
Cluster 2	30	42	38	110
Cluster 3	38	35	27	100
	100	100	100	300

**Purity:** (0.38, 0.38, 0.38)

– overall 0.38

**Precision:** (0.38, 0.38, 0.38)

– overall 0.38

**Recall:** (0.35, 0.42, 0.38)

– overall 0.39

# Another clustering

	Class 1	Class 2	Class 3	
Cluster 1	0	0	35	35
Cluster 2	50	77	38	165
Cluster 3	38	35	27	100
	100	100	100	300

**Cluster 1:**  
Purity: 1  
Precision: 1  
Recall: 0.35

# External Measures of Cluster Validity: Entropy and Purity

**Table 5.9.** K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

**entropy** For each cluster, the class distribution of the data is calculated first, i.e., for cluster  $j$  we compute  $p_{ij}$ , the ‘probability’ that a member of cluster  $j$  belongs to class  $i$  as follows:  $p_{ij} = m_{ij}/m_j$ , where  $m_j$  is the number of values in cluster  $j$  and  $m_{ij}$  is the number of values of class  $i$  in cluster  $j$ . Then using this class distribution, the entropy of each cluster  $j$  is calculated using the standard formula  $e_j = \sum_{i=1}^L p_{ij} \log_2 p_{ij}$ , where the  $L$  is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e.,  $e = \sum_{j=1}^K \frac{m_j}{m} e_j$ , where  $m_j$  is the size of cluster  $j$ ,  $K$  is the number of clusters, and  $m$  is the total number of data points.

**purity** Using the terminology derived for entropy, the purity of cluster  $j$ , is given by  $\text{purity}_j = \max_i p_{ij}$  and the overall purity of a clustering by  $\text{purity} = \sum_{j=1}^K \frac{m_j}{m} \text{purity}_j$ .

# Final Comment on Cluster Validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

*Algorithms for Clustering Data*, Jain and Dubes

# MIXTURE MODELS AND THE EM ALGORITHM

---

# Model-based clustering

- In order to understand our data, we will assume that there is a **generative process** (a **model**) that creates/describes the data, and we will try to find the model that **best fits** the data.
  - Models of different complexity can be defined, but we will assume that our model is a **distribution** from which data points are sampled
  - Example: the data is the height of all people in Greece
- In most cases, a single distribution is not good enough to describe all data points: different parts of the data follow a different distribution
  - Example: the data is the height of all people in Greece and China
  - We need a **mixture model**
  - Different distributions correspond to different clusters in the data.

# Gaussian Distribution

- Example: the data is the height of all people in Greece
  - Experience has shown that this data follows a Gaussian (Normal) distribution
  - Reminder: Normal distribution:

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- $\mu$  = mean,  $\sigma$  = standard deviation



# Gaussian Model

- What is a model?
  - A Gaussian distribution is fully defined by the mean  $\mu$  and the standard deviation  $\sigma$
  - We define our model as the pair of parameters  $\theta = (\mu, \sigma)$
- This is a general principle: a model is defined as a **vector of parameters**  $\theta$

# Fitting the model

- We want to find the normal distribution that best fits our data
  - Find the best values for  $\mu$  and  $\sigma$
  - But what does best fit mean?

# Maximum Likelihood Estimation (MLE)

- Find the **most likely parameters given the data**. Given the data observations  $X$ , find  $\theta$  that maximizes  $P(\theta|X)$ 
  - Problem: We do not know how to compute  $P(\theta|X)$

- Using Bayes Rule:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

- If we have no **prior information** about  $\theta$ , or  $X$ , we can assume uniform. Maximizing  $P(\theta|X)$  is the same as maximizing  $P(X|\theta)$

# Maximum Likelihood Estimation (MLE)

- We have a vector  $X = (x_1, \dots, x_n)$  of values and we want to fit a Gaussian  $N(\mu, \sigma)$  model to the data
  - Our parameter set is  $\theta = (\mu, \sigma)$

- Probability of observing point  $x_i$  given the parameters  $\theta$

$$P(x_i|\theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- Probability of observing all points (assume independence)

$$P(X|\theta) = \prod_{i=1}^n P(x_i|\theta) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- We want to find the parameters  $\theta = (\mu, \sigma)$  that maximize the probability  $P(X|\theta)$

# Maximum Likelihood Estimation (MLE)

- The probability  $P(X|\theta)$  as a function of  $\theta$  is called the **Likelihood** function

$$L(\theta) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

- It is usually easier to work with the **Log-Likelihood** function

$$LL(\theta) = -\sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma^2} - \frac{1}{2}n \log 2\pi - n \log \sigma$$

- Maximum Likelihood Estimation**

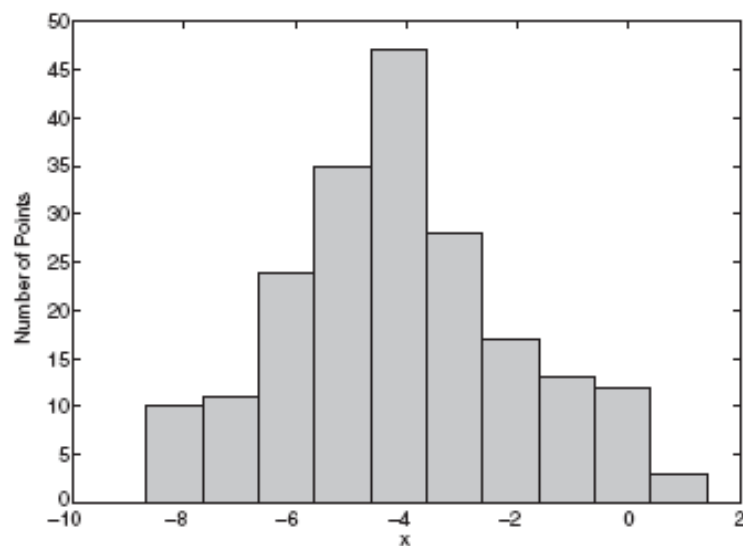
- Find parameters  $\mu, \sigma$  that maximize  $LL(\theta)$

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i = \mu_X$$

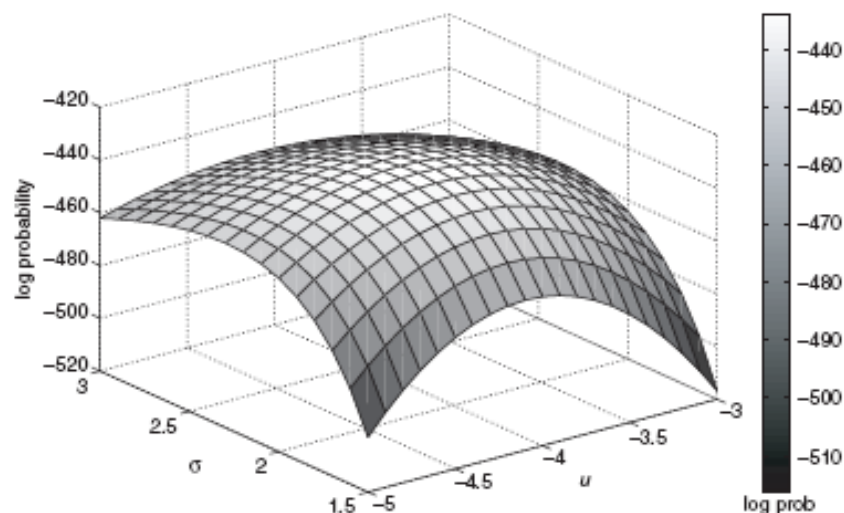
Sample Mean

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 = \sigma_X^2$$

Sample Variance



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

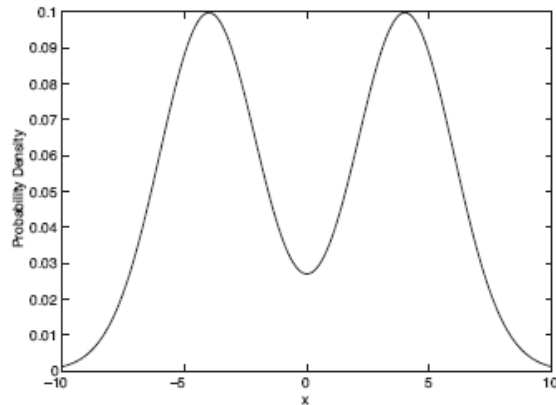


(b) Log likelihood plot of the 200 points for different values of the mean and standard deviation.

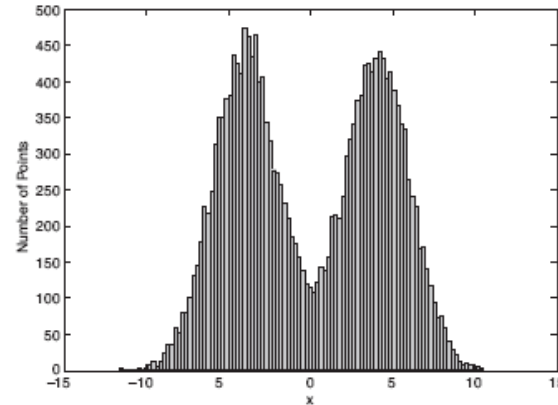
**Figure 9.3.** 200 points from a Gaussian distribution and their log probability for different parameter values.

# Mixture of Gaussians

- Suppose that you have the heights of people from Greece and China and the distribution looks like the figure below (dramatization)



(a) Probability density function for the mixture model.

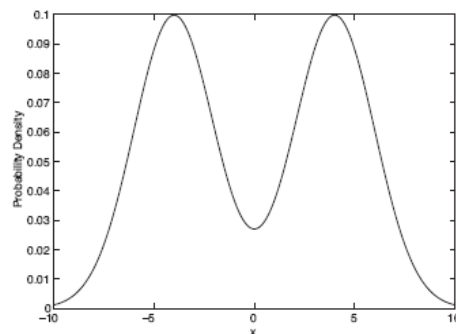


(b) 20,000 points generated from the mixture model.

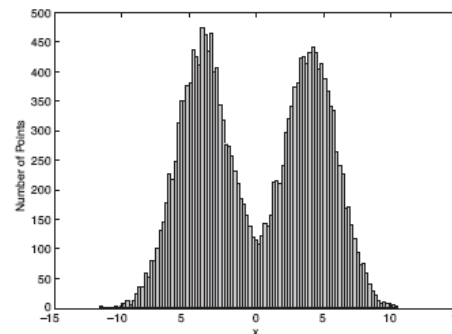
**Figure 9.2.** Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

# Mixture of Gaussians

- In this case the data is the result of the **mixture** of **two Gaussians**
  - One for Greek people, and one for Chinese people
  - Identifying **for each value** which Gaussian is **most likely to have generated it** will give us a **clustering**.



(a) Probability density function for the mixture model.



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

**Figure 9.2.** Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.



# Mixture model

- A value  $x_i$  is generated according to the following process:
  - First **select the nationality**
    - With probability  $\pi_G$  select Greece, with probability  $\pi_C$  select China ( $\pi_G + \pi_C = 1$ )
- Given the nationality, **generate the point** from the corresponding Gaussian
  - $P(x_i|\theta_G) \sim N(\mu_G, \sigma_G)$  if Greece
  - $P(x_i|\theta_C) \sim N(\mu_C, \sigma_C)$  if China

We can also think of this as a **Hidden Variable**  $Z$  that takes two values: Greece and China

$\theta_G$ : parameters of the Greek distribution

$\theta_C$ : parameters of the China distribution

# Mixture Model

- Our model has the following parameters

$$\Theta = (\pi_G, \pi_C, \mu_G, \sigma_G, \mu_C, \sigma_C)$$

Mixture probabilities

$\theta_G$ : parameters of the Greek distribution

$\theta_C$ : parameters of the China distribution

# Mixture Model

- Our model has the following parameters

$$\Theta = (\pi_G, \pi_C, \mu_G, \sigma_G, \mu_C, \sigma_C)$$

Mixture probabilities

Distribution Parameters

- For value  $x_i$ , we have:

$$P(x_i|\Theta) = \pi_G P(x_i|\theta_G) + \pi_C P(x_i|\theta_C)$$

- For all values  $X = (x_1, \dots, x_n)$

$$P(X|\Theta) = \prod_{i=1}^n P(x_i|\Theta)$$

- We want to estimate the parameters that **maximize** the Likelihood of the data

# Mixture Models

- Once we have the parameters  $\Theta = (\pi_G, \pi_C, \mu_G, \mu_C, \sigma_G, \sigma_C)$  we can **estimate** the **membership probabilities**  $P(G|x_i)$  and  $P(C|x_i)$  for each point  $x_i$ :
  - This is the probability that point  $x_i$  belongs to the Greek or the Chinese population (**cluster**)

Given from the Gaussian distribution  $N(\mu_G, \sigma_G)$  for Greek

$$\begin{aligned} P(G|x_i) &= \frac{P(x_i|G)P(G)}{P(x_i|G)P(G) + P(x_i|C)P(C)} \\ &= \frac{P(x_i|\theta_G)\pi_G}{P(x_i|\theta_G)\pi_G + P(x_i|\theta_C)\pi_C} \end{aligned}$$

# EM (Expectation Maximization) Algorithm

- Initialize the values of the parameters in  $\Theta$  to some random values
- Repeat until convergence
  - **E-Step**: Given the parameters  $\Theta$  **estimate** the membership probabilities  $P(G|x_i)$  and  $P(C|x_i)$
  - **M-Step**: Compute the parameter values that (in expectation) **maximize** the data likelihood

$$\pi_C = \frac{1}{n} \sum_{i=1}^n P(C|x_i)$$

$$\pi_G = \frac{1}{n} \sum_{i=1}^n P(G|x_i)$$

Fraction of  
population in G,C

$$\mu_C = \frac{1}{n * \pi_C} \sum_{i=1}^n P(C|x_i) x_i$$

$$\mu_G = \frac{1}{n * \pi_G} \sum_{i=1}^n P(G|x_i) x_i$$

MLE Estimates  
if  $\pi$ 's were fixed

$$\sigma_C^2 = \frac{1}{n * \pi_C} \sum_{i=1}^n P(C|x_i) (x_i - \mu_C)^2$$

$$\sigma_G^2 = \frac{1}{n * \pi_G} \sum_{i=1}^n P(G|x_i) (x_i - \mu_G)^2$$

# Relationship to K-means

- **E-Step**: Assignment of points to clusters
  - K-means: **hard** assignment, EM: **soft** assignment
- **M-Step**: Computation of centroids
  - K-means assumes common fixed variance (**spherical clusters**)
  - EM: can change the variance for different clusters or different dimensions (**ellipsoid clusters**)
- If the variance is fixed then both minimize the same error function

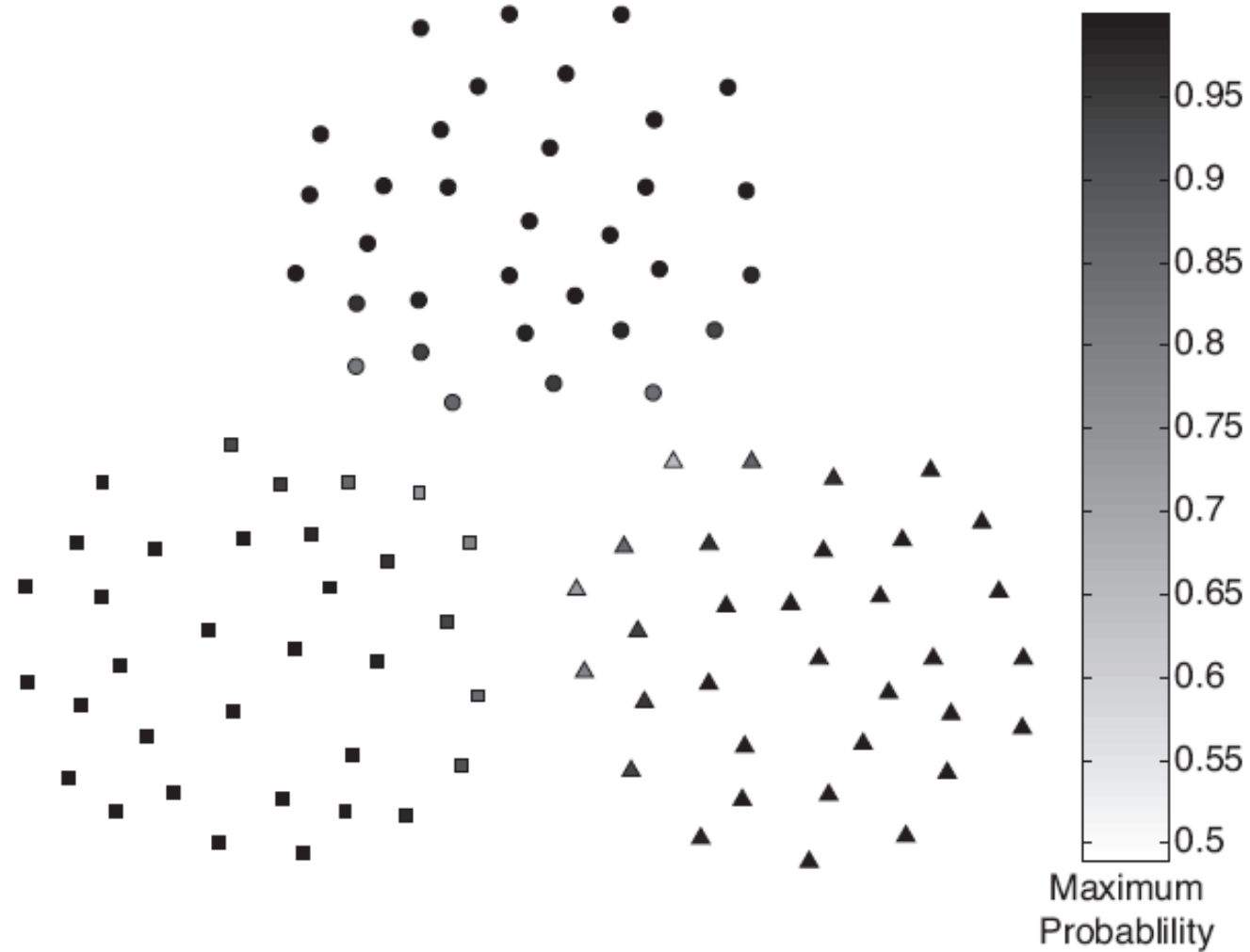
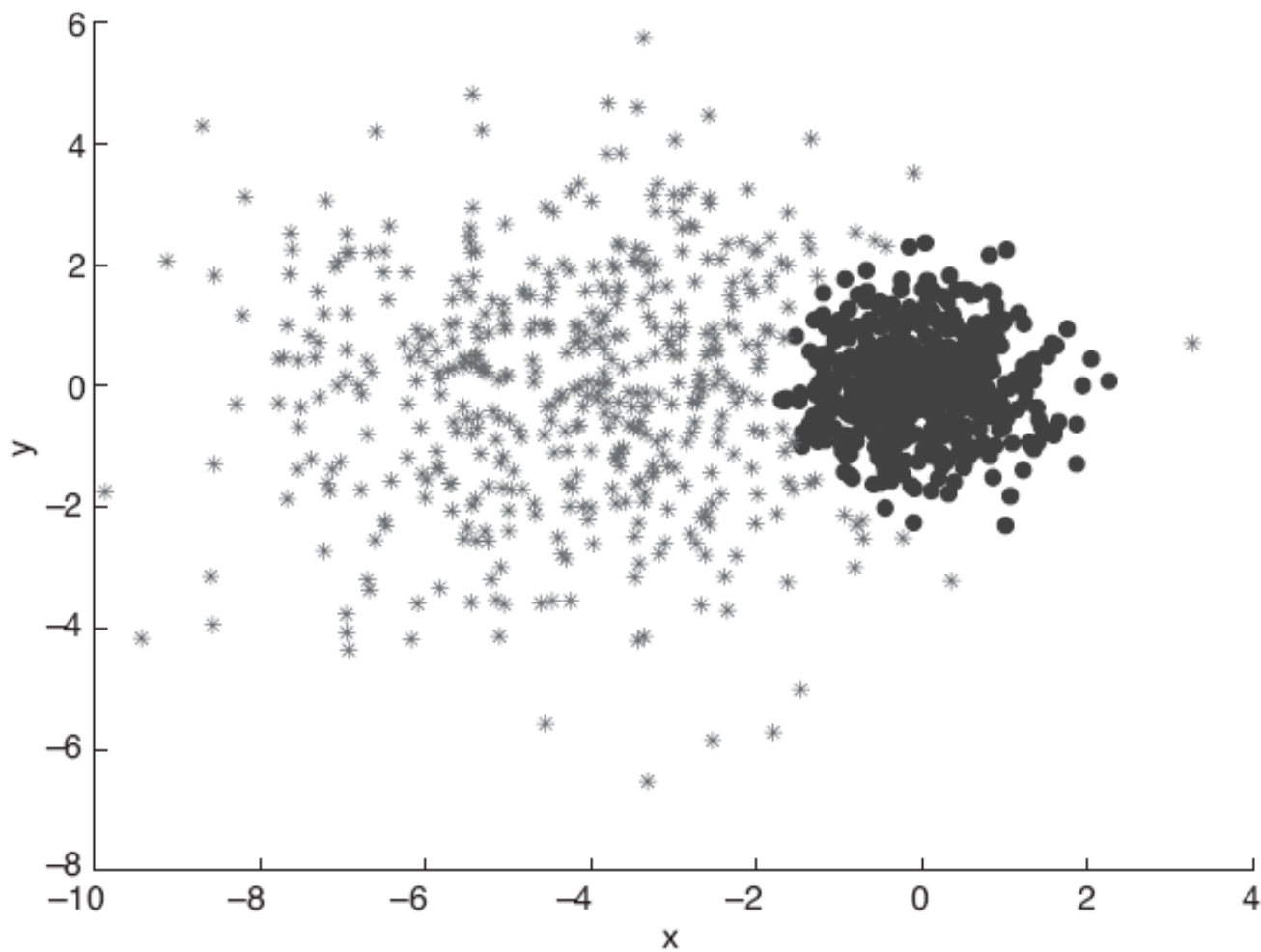
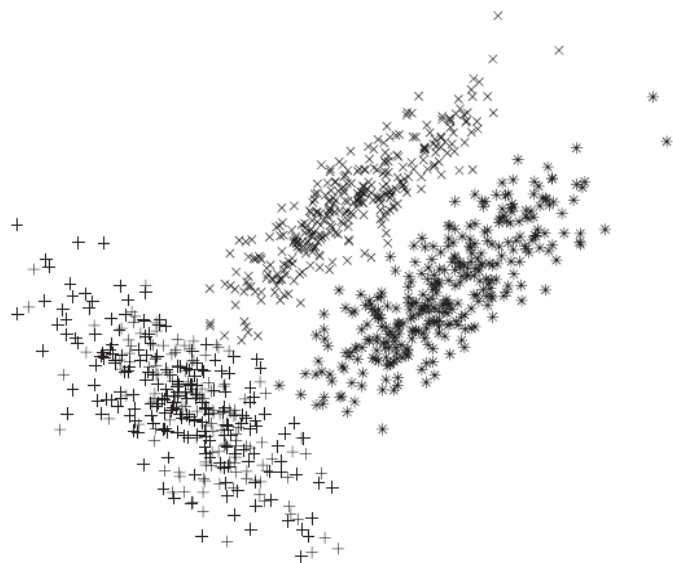


Figure 9.4. EM clustering of a two-dimensional point set with three clusters.

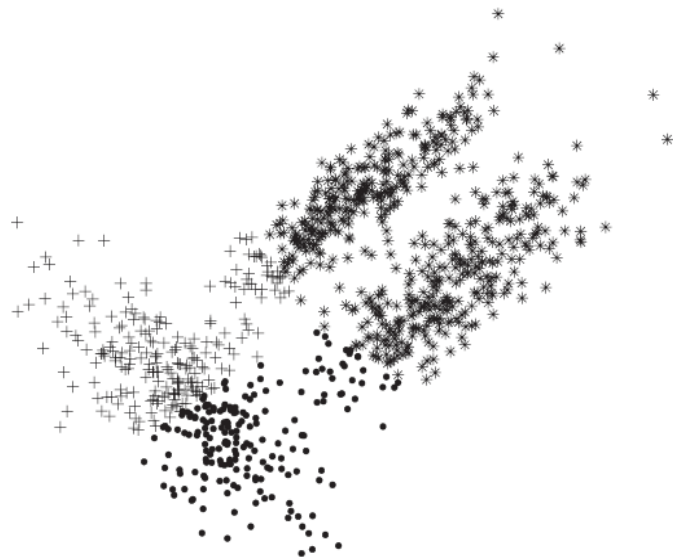


**Figure 9.5.** EM clustering of a two-dimensional point set with two clusters of differing density.





(a) Clusters produced by mixture model clustering.



(b) Clusters produced by K-means clustering.

**Figure 9.6.** Mixture model and K-means clustering of a set of two-dimensional points.