Clustering

Evgeny Burnaev

Skoltech, Moscow, Russia

Outline

Supervised Learning

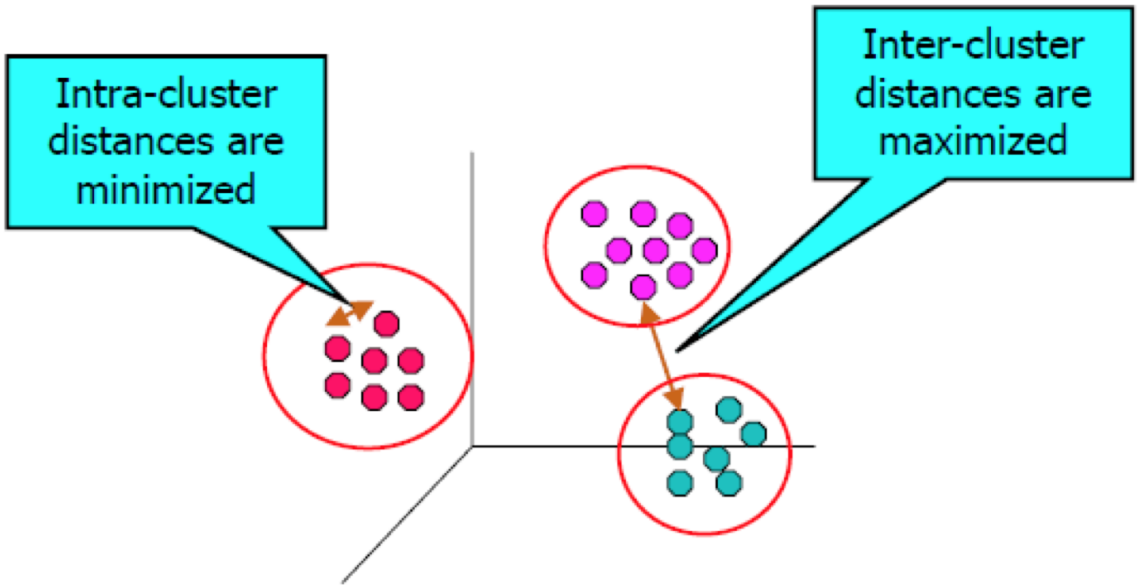
- Given training set: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$
- i.i.d. samples from a distribution D
- Find a general hypothesis $y = h(\mathbf{x})$
- An approximation to a target (true) function $y = f(\mathbf{x})$

Unsupervised Problem

- Given training set: $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$
- i.i.d. samples from a distribution D
- Determine how the data are organized: identify, summarize and explain key features
  - Density estimation
  - Clustering
  - Dimension reduction
  - ...

What is cluster analysis?

- Finding groups of objects such that the objects in a group are
  - Similar (or related) to one another, and
  - Different from (or unrelated to) the objects in other groups
- Two types of approaches: distance based and model-based



Distance based Clustering

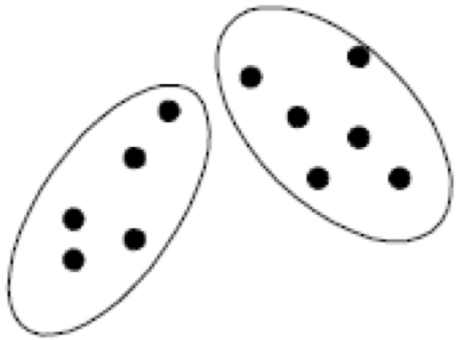
Distance/Similarity Measures

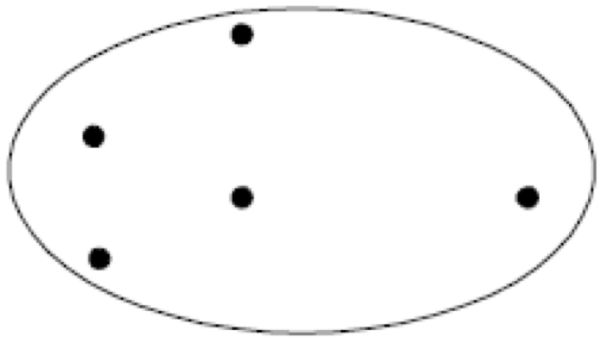
- L_2-norm (Euclidean distance):
  $d(\mathbf{x}, \mathbf{z}) = \sqrt{\sum_{i=1}^{N} (x_i - z_i)^2}$
- L_1-norm (Manhattan distance): $d(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{N} |x_i - z_i|$
- L_∞-norm: $d(\mathbf{x}, \mathbf{z}) = \max_{i \in \overline{1,N}} |x_i - z_i|$
- Cosine similarity: $d(\mathbf{x}, \mathbf{z}) = \frac{\mathbf{x}^{\mathrm{T}} \mathbf{z}}{\|\mathbf{x}\| \|\mathbf{z}\|}$
- ...

Partitional and Hierarchical Clustering

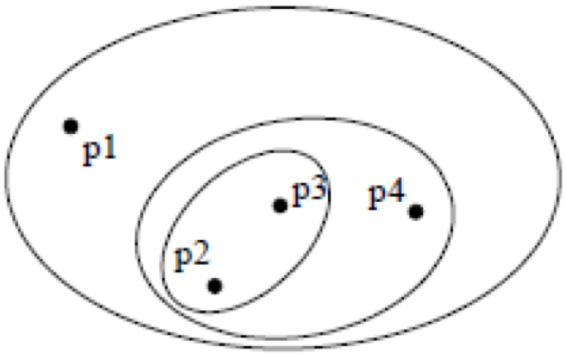
- Partitional clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree (dendrogram)

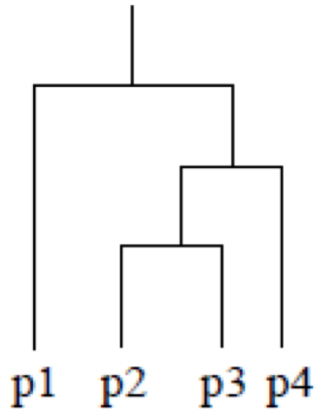
Partitional clustering

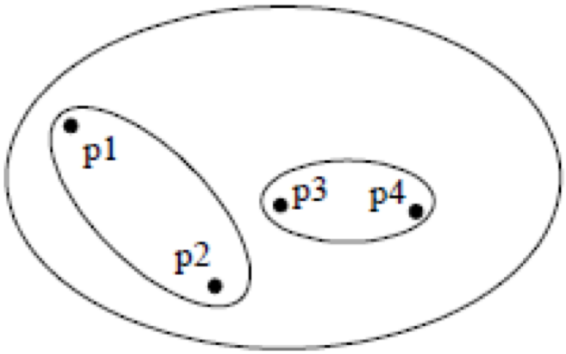


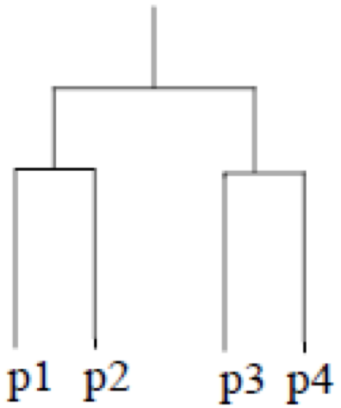


Hierarchical clustering



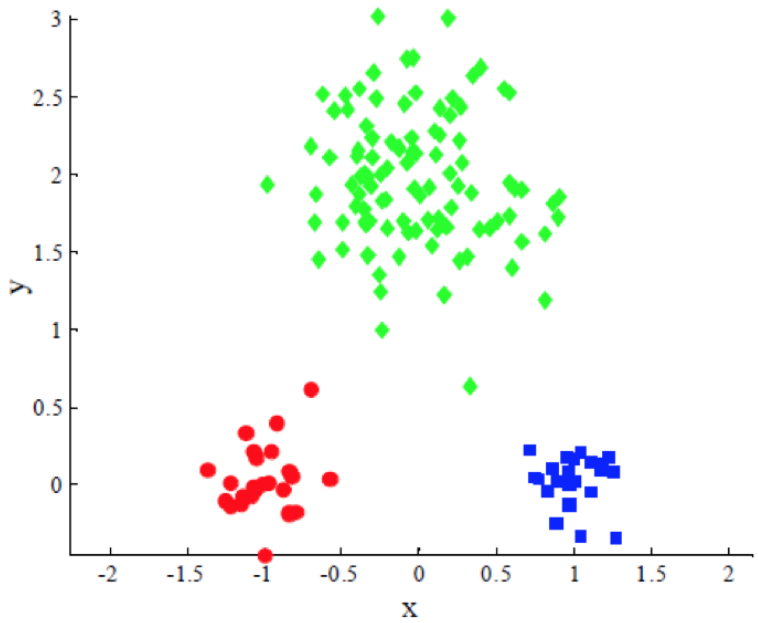


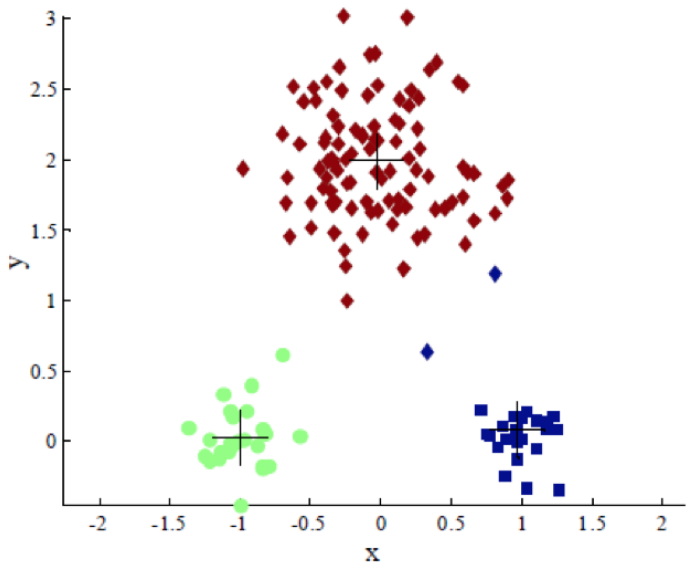




K-means clustering

- For data points in Euclidean space
- 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

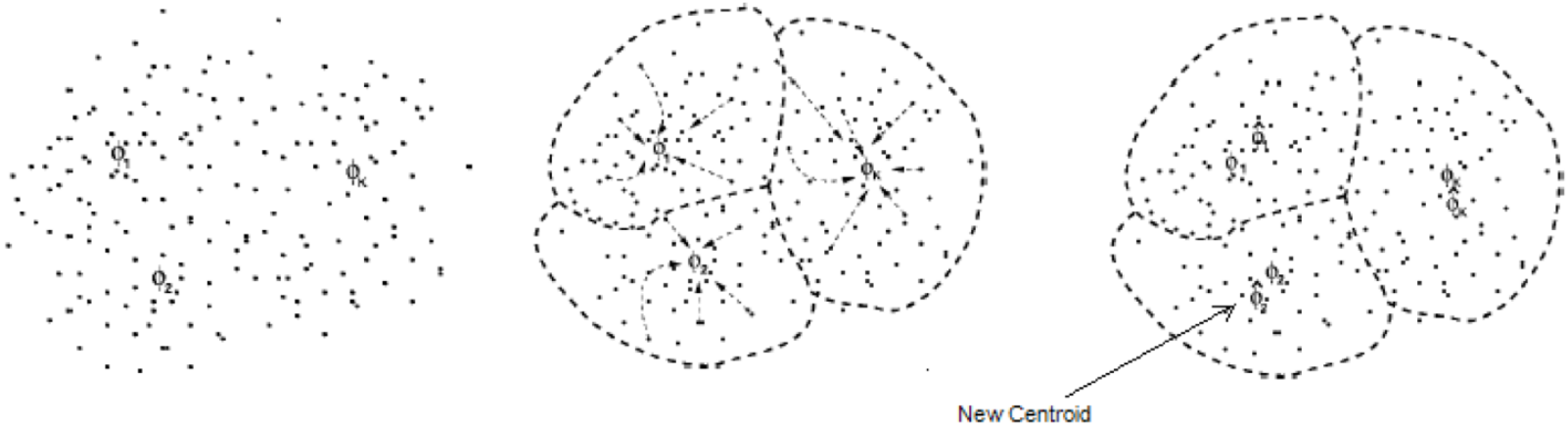




K-means clustering

The basic algorithm

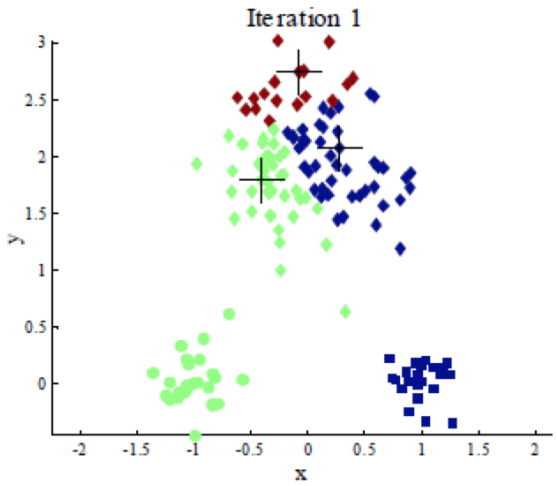
- Select K points as the initial centroids
- repeat:
  - for each of K clusters by assigning all points to the closest centroid
  - recompute the centroid of each cluster
- until the centroids don't change

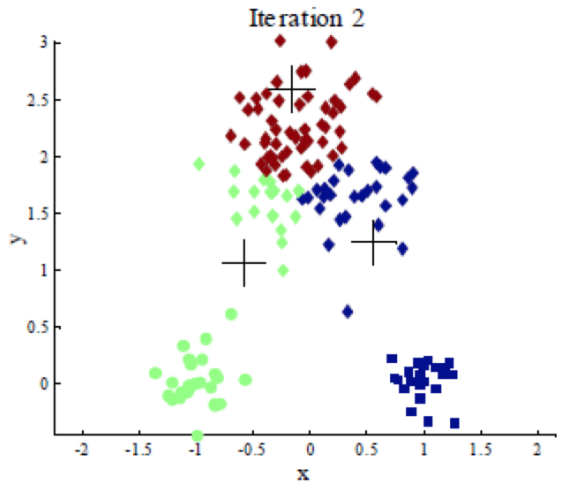


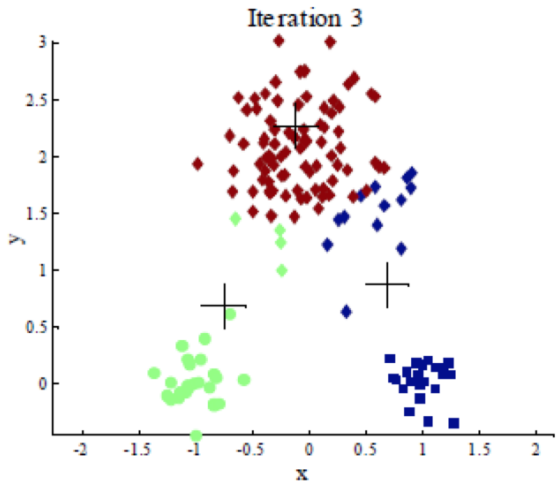
K-means clustering: Details

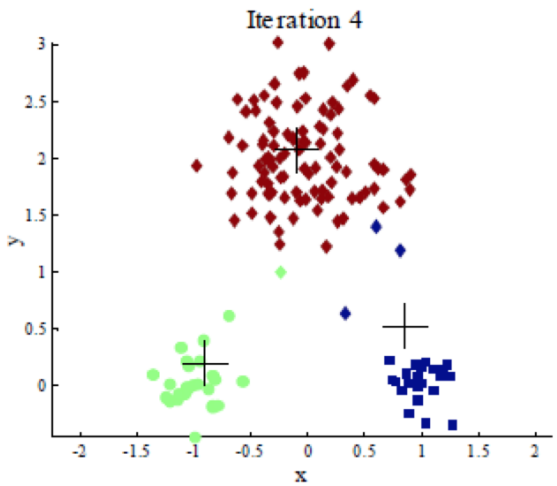
- Initial centroids are often chosen randomly
- The centroid is (typically) the mean of the points in the cluster
- "Closeness" is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means converges for common similarity measures mentioned above, often in the first few iterations
- Clusters produced vary from one run to another
- Complexity is $O(I \times m \times K \times N)$, where
  - I — number of iterations
  - m — number of points
  - K — number of clusters
  - N — number of attributes

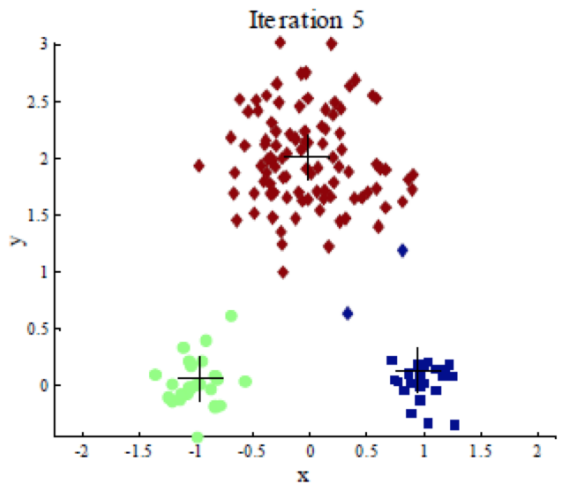
Importance of Choosing Initial Centroids I

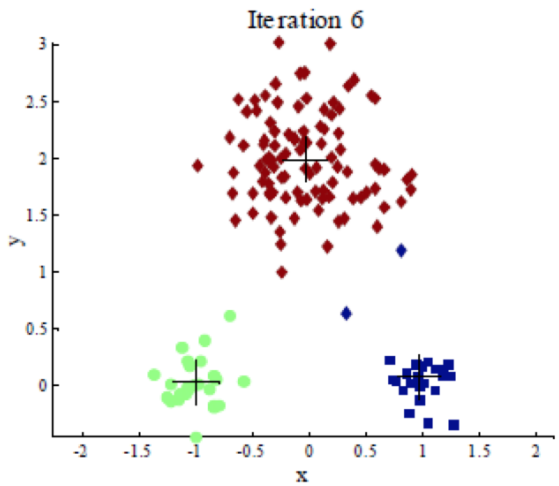




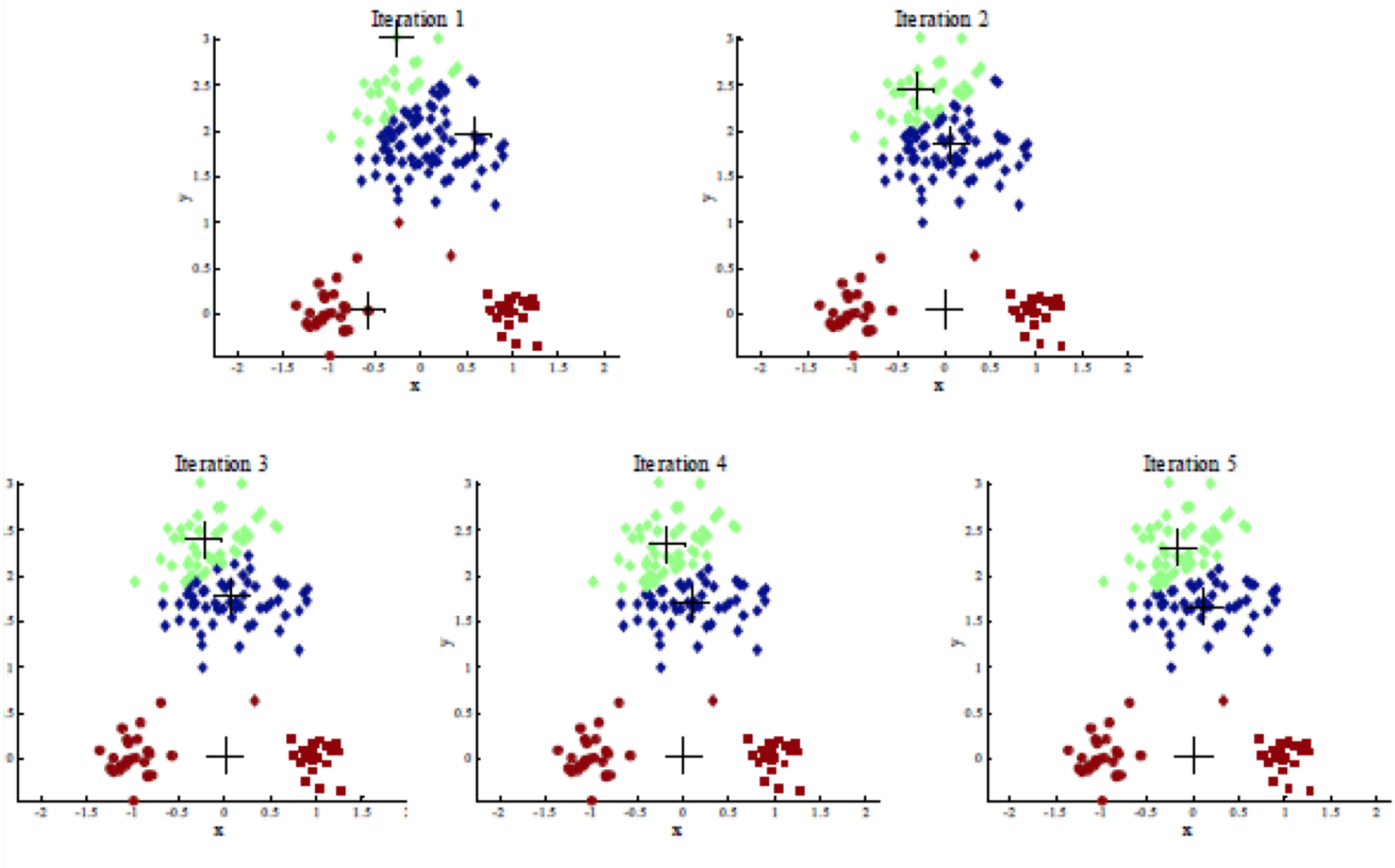








Importance of Choosing Initial Centroids II



Solutions to Initial Centroids Problem

- Multiple runs. Pick the solution with minimum sum of squared error (SSE)
- Bisecting K-means
- Use hierarchical clustering to determine initial centroids
- ...

Sum of Squared Error (SSE)

- For each point, the error is the distance to the nearest cluster center
- To get SSE, we square these errors and sum them

$$SSE = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{m}_i\|^2$$

- C_i is the set of all points in cluster i; $\mathbf{m}_i$ is the i-th centroid
- Given two clusterings, we can choose the one with the smallest error

Bisecting K-means

- Start with ONE cluster that contains all points
- repeat:
  - Select a cluster from the list of clusters
  - Repeat a number of times: run K-means on the cluster to divide it into 2 clusters
  - Among all pairs obtained, pick the one with lowest SSE and add them to the list of clusters
- until there are K clusters

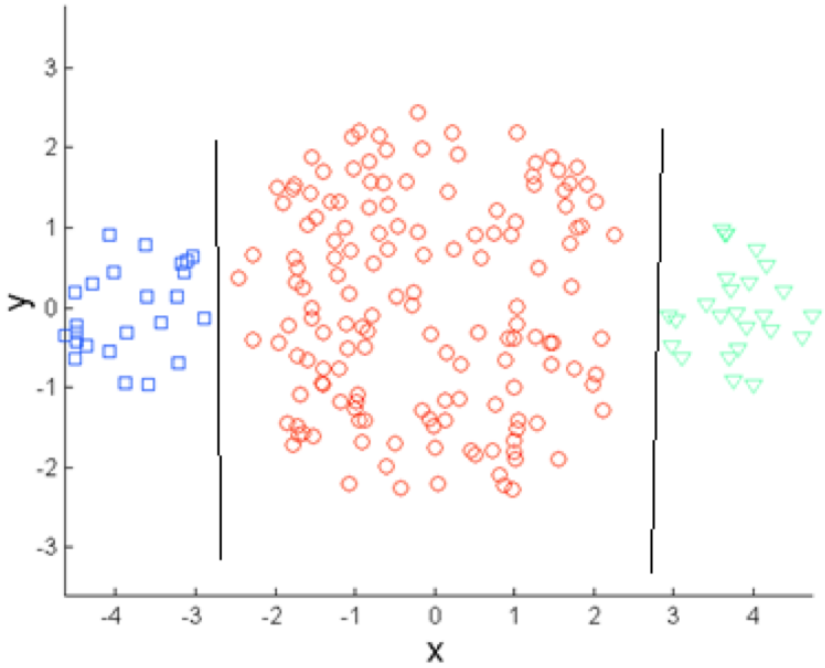
Comments:

- selection of initial points no longer an issue
- usually, select the largest to split

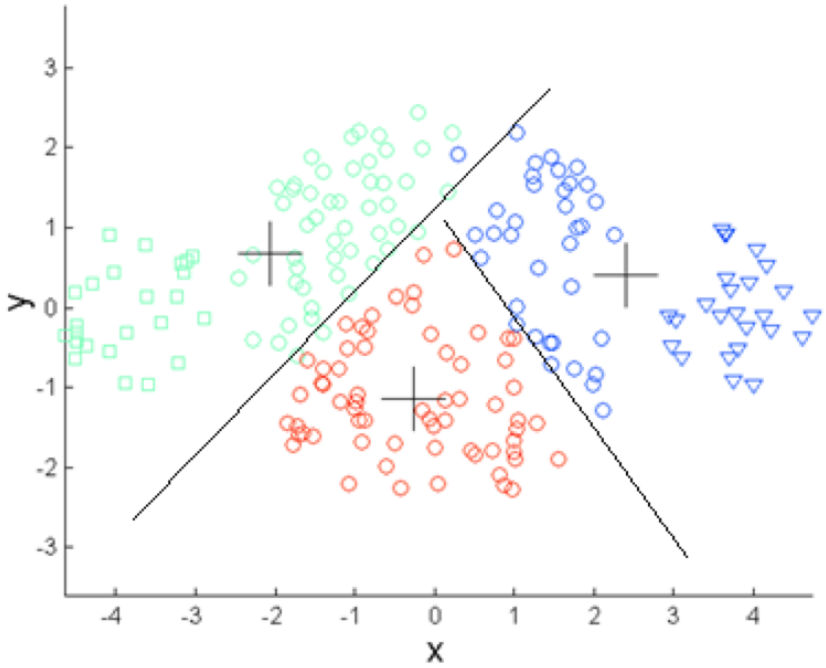
Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-global shapes
- K-means has problems when the data contains outliers

Different Sizes



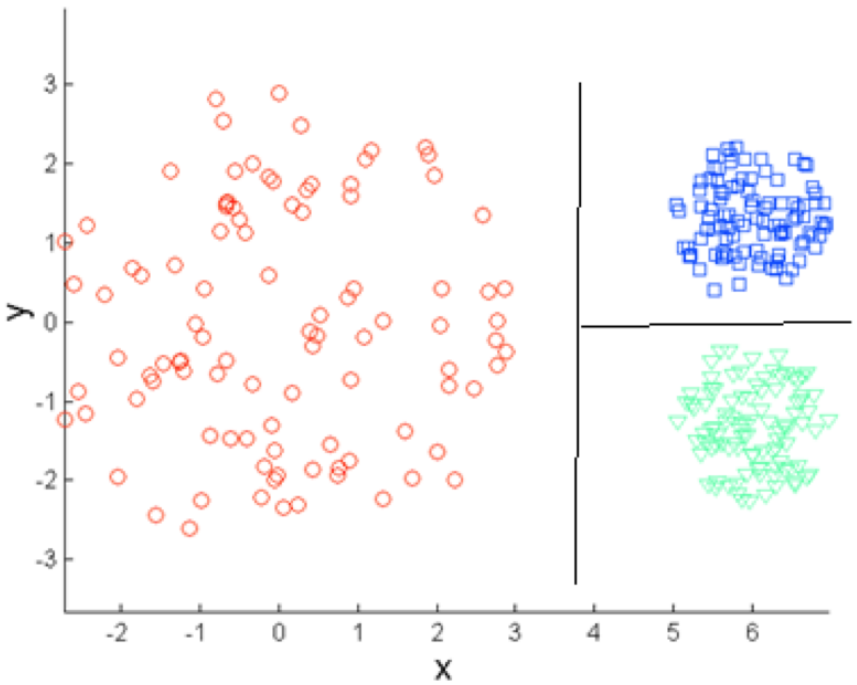
Original Points



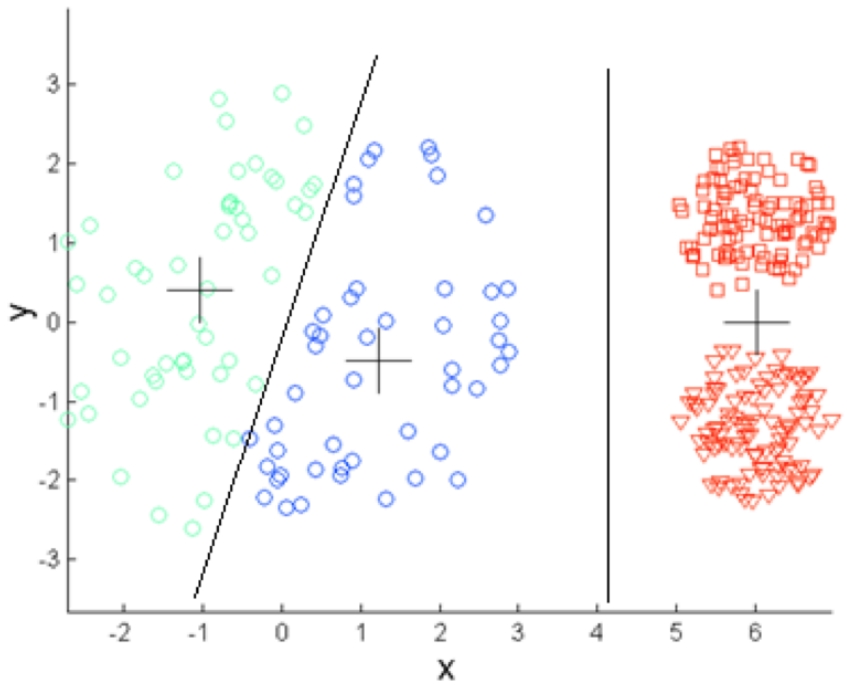
K-means (3 Clusters)

Some points in a large cluster might be closer to centroids of neighboring clusters. Hence, mis-grouped

Different Densities



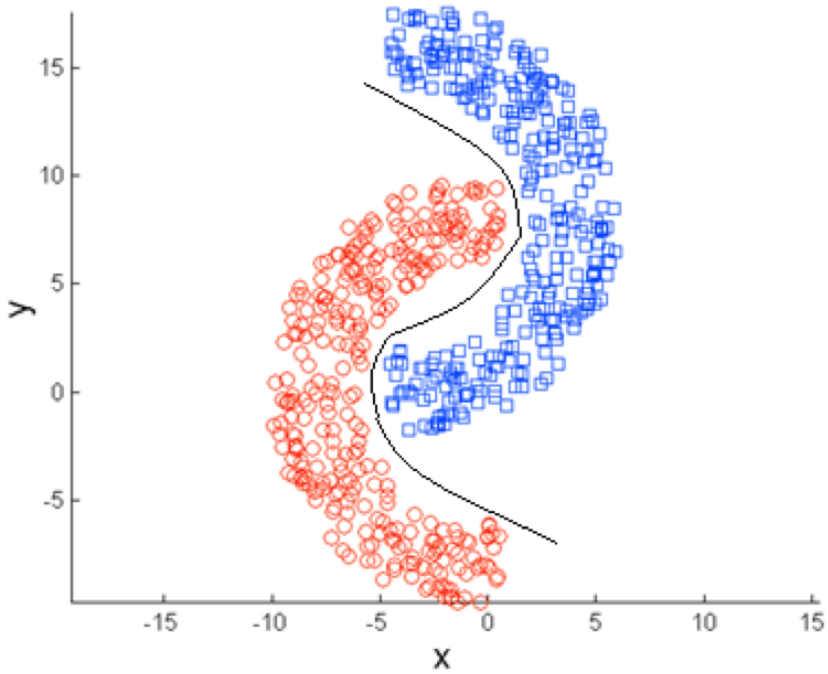
Original Points



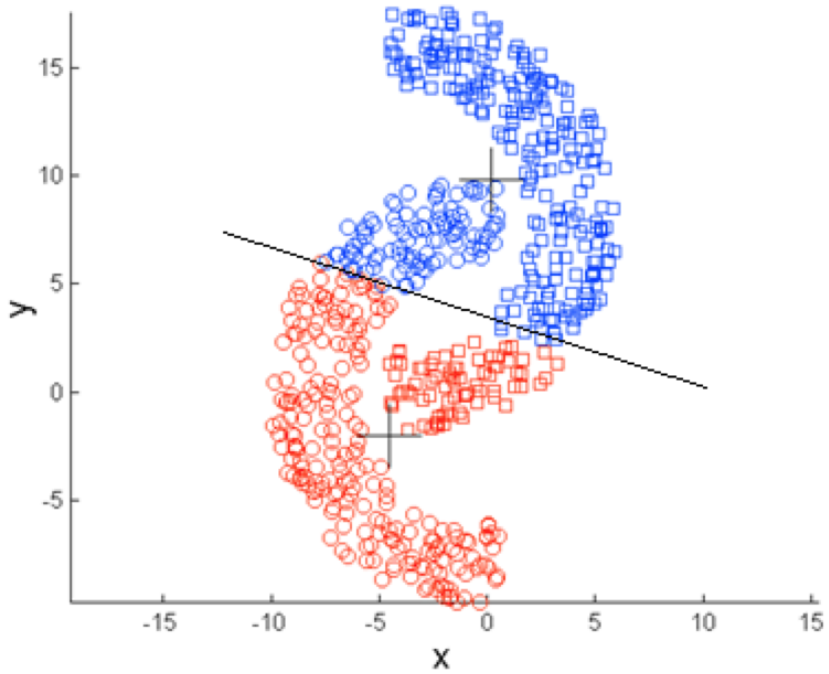
K-means (3 Clusters)

Sparse clusters might get more centroids than fair share

Non-spherical shapes



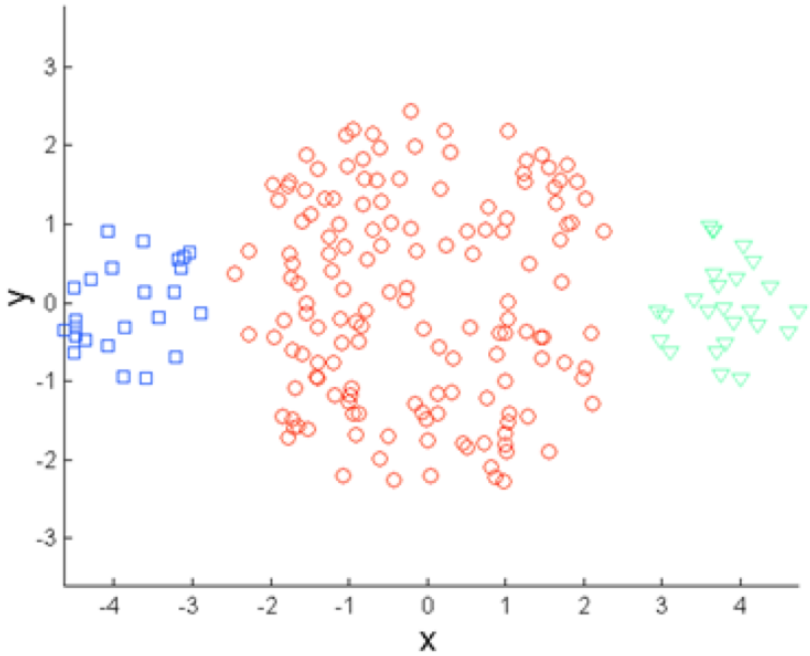
Original Points



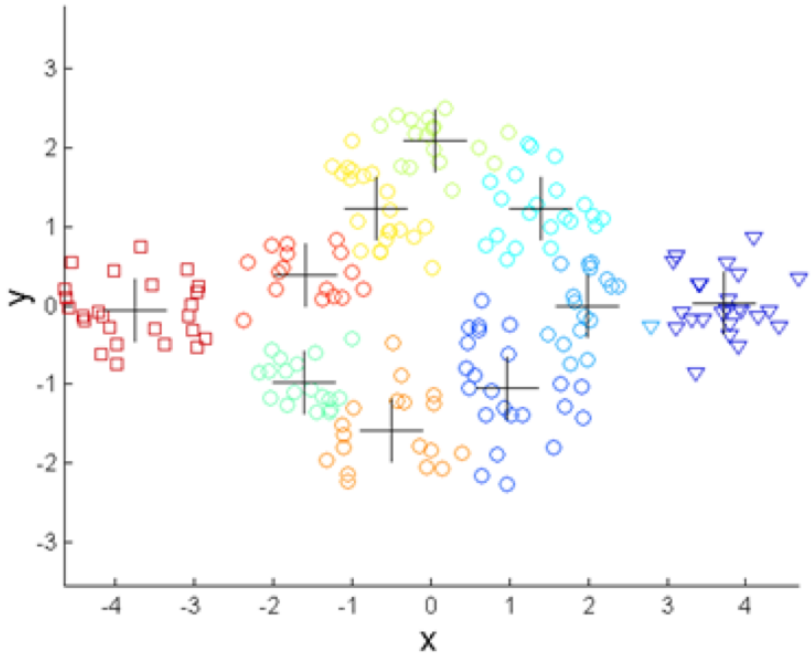
K-means (2 Clusters)

K-means assumes spherical clusters. Can get wrong results when clusters have other shapes

Overcoming limitations I



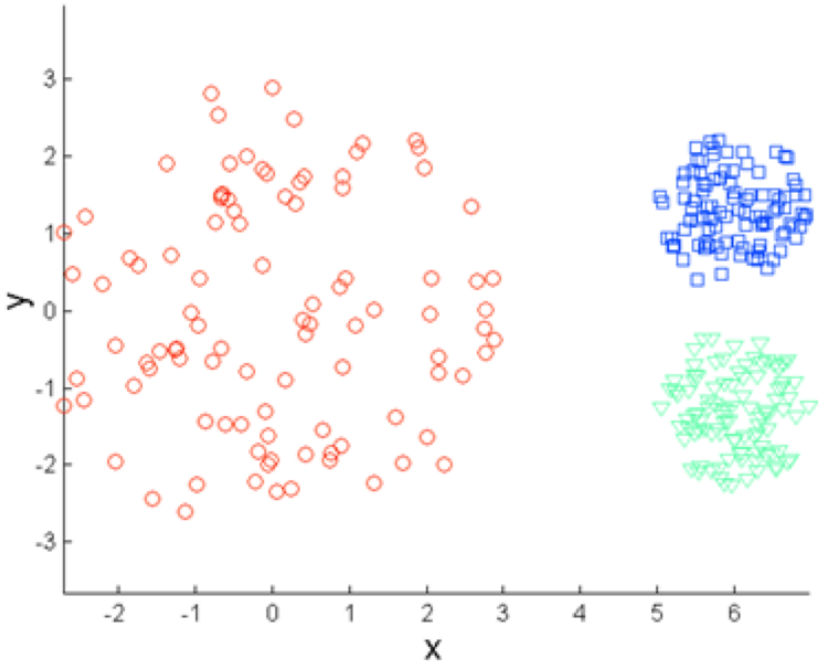
Original Points



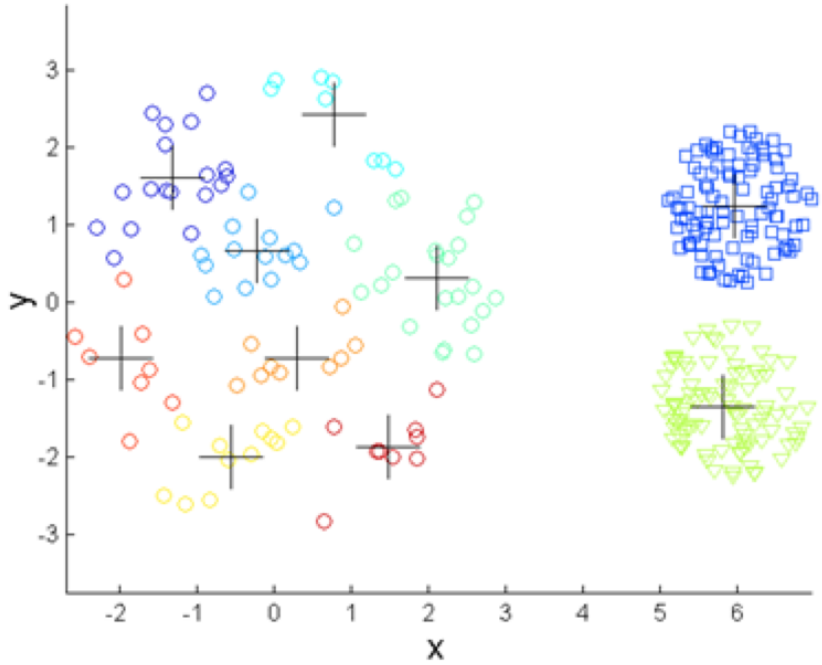
K-means Clusters

Find many clusters, and group small clusters into big ones based on domain knowledge

Overcoming limitations II



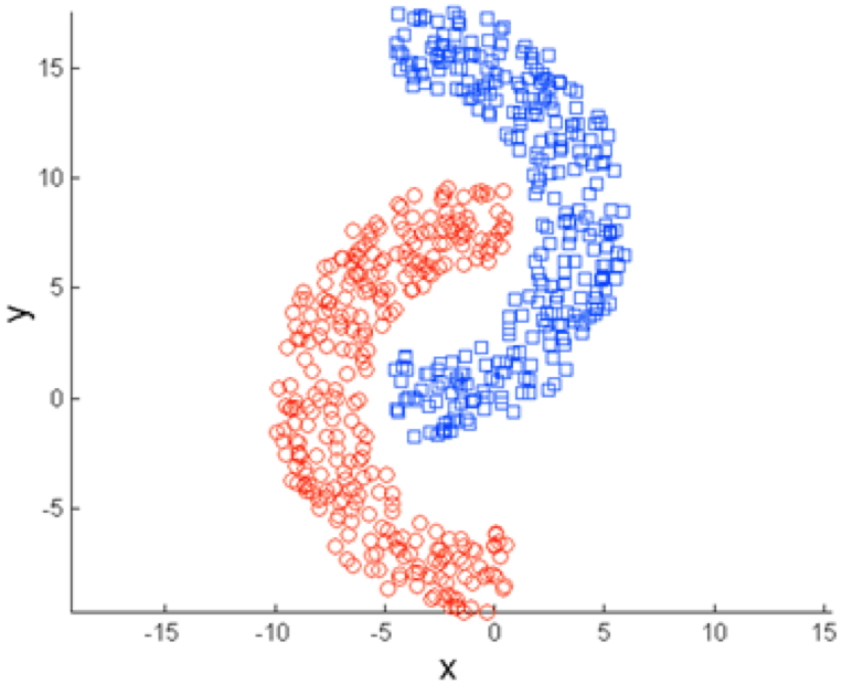
Original Points



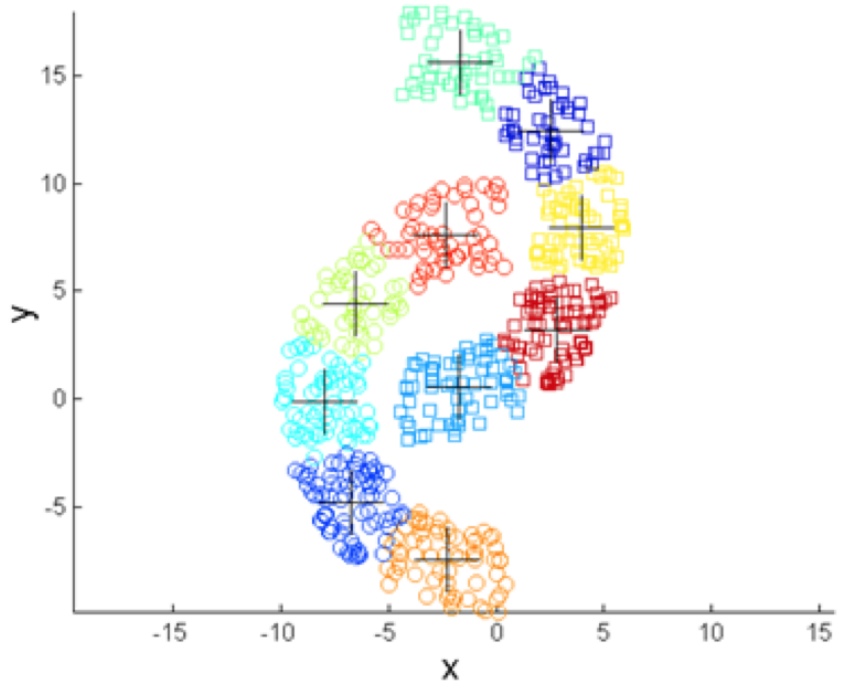
K-means Clusters

Find many clusters, and group small clusters into big ones based on domain knowledge

Overcoming limitations III



Original Points

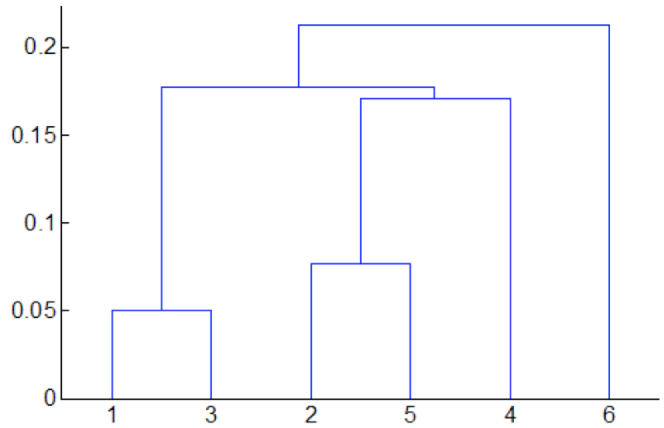


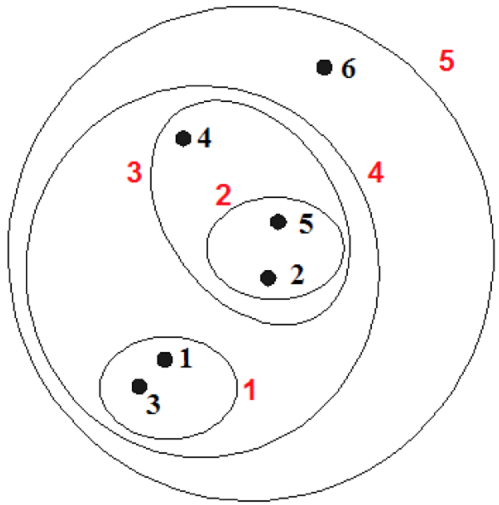
K-means Clusters

Find many clusters, and group small clusters into big ones based on domain knowledge

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





Partitional clustering

- Agglomerative hierarchical clustering
  - 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 hierarchical clustering
  - Start with one, all-inclusive cluster
  - At each step, split a cluster until each cluster contains a point (or there are K clusters)
- Need similarity or distance measure

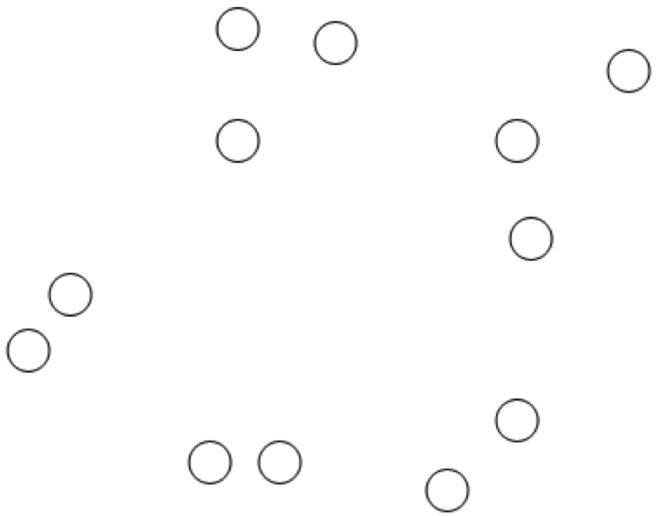
Agglomerative HC

- Compute the proximity matrix
- Let each data point be a cluster
  - repeat:
    A) Merge the two closest clusters
    B) Update the proximity matrix

    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

Initial Situation

- Start with clusters of individual points and a proximity matrix

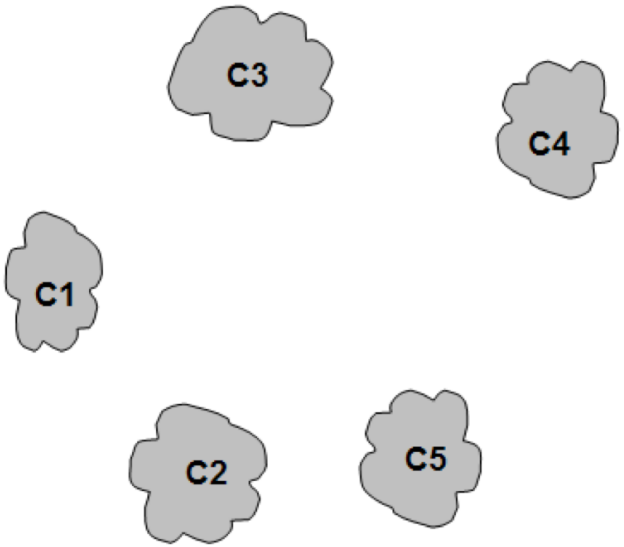


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

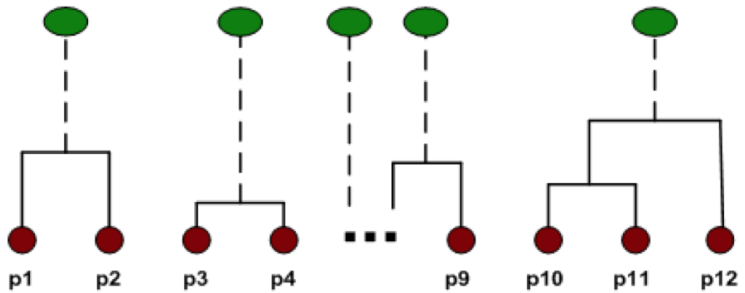
After some steps

- After some merging steps, we have some clusters



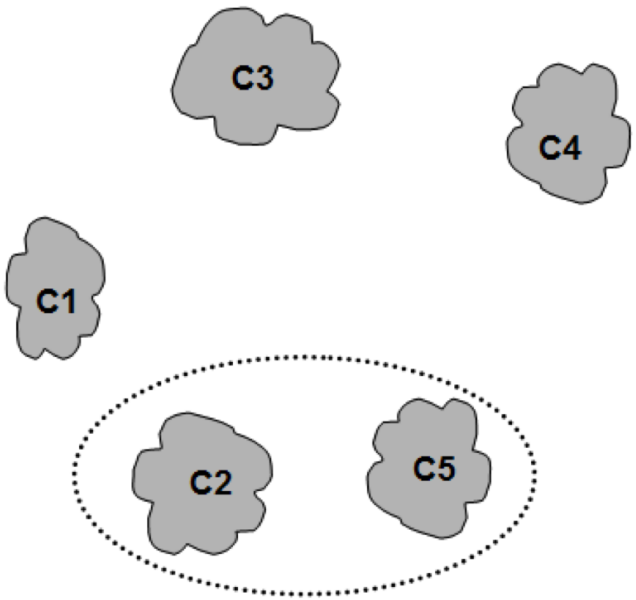
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

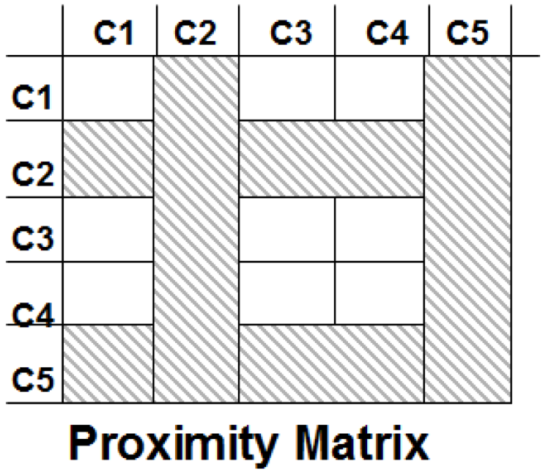
Proximity Matrix



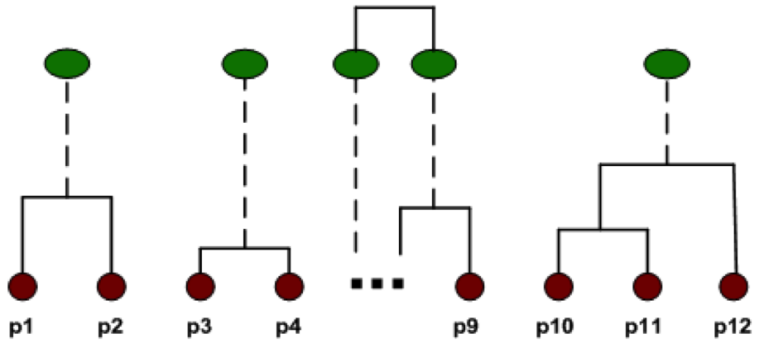
After some steps

- We want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix



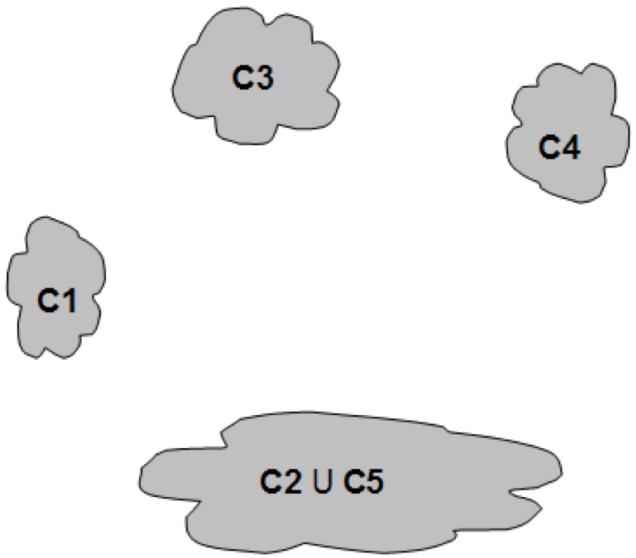


Proximity Matrix



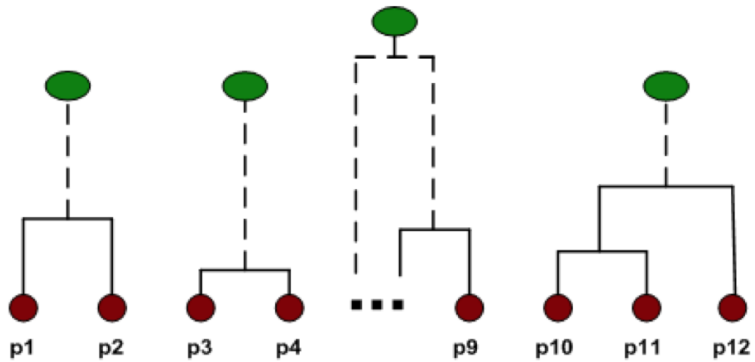
After some steps

- How do we update the proximity matrix?



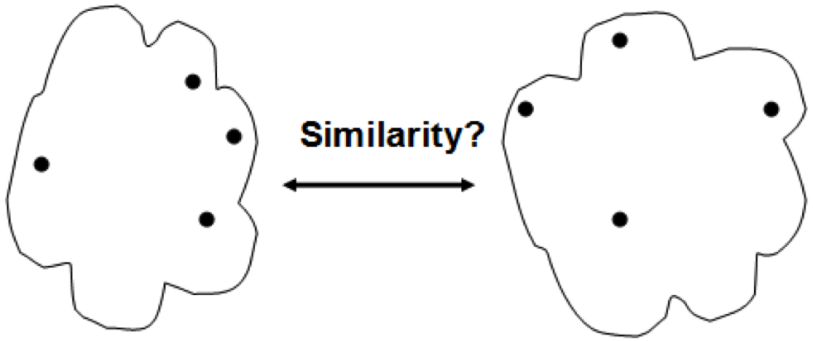
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



Inter-Cluster Similarity

Min, Max, Group average, Distance between centroid, . . .

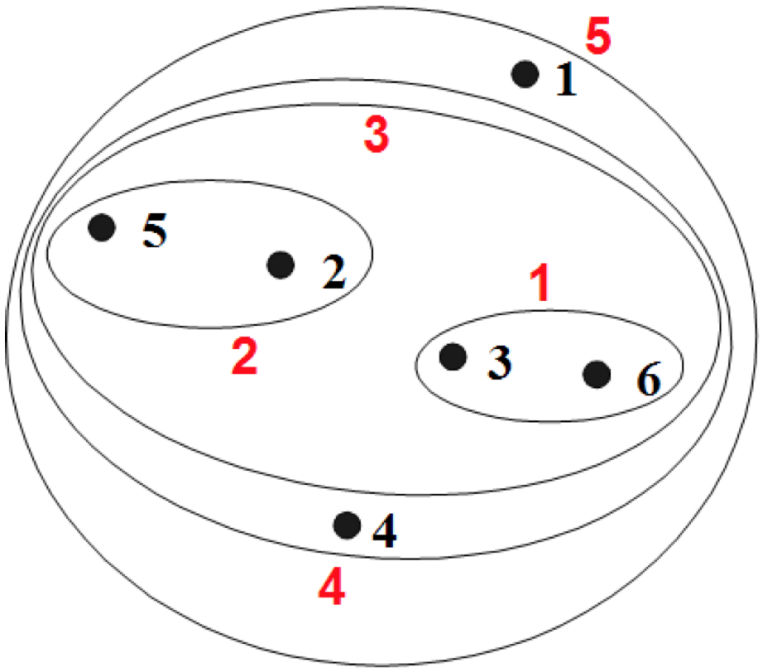


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

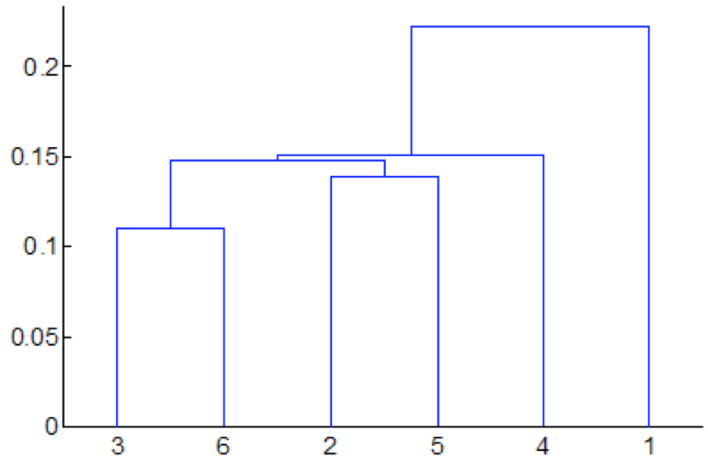
Proximity Matrix

Cluster Similarity: MIN or Single Link

Distance is used



Nested Clusters

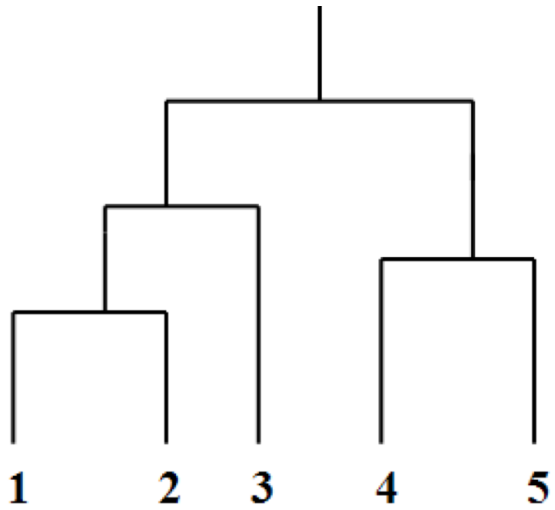


Dendrogram

Cluster Similarity: MIN or Single Link

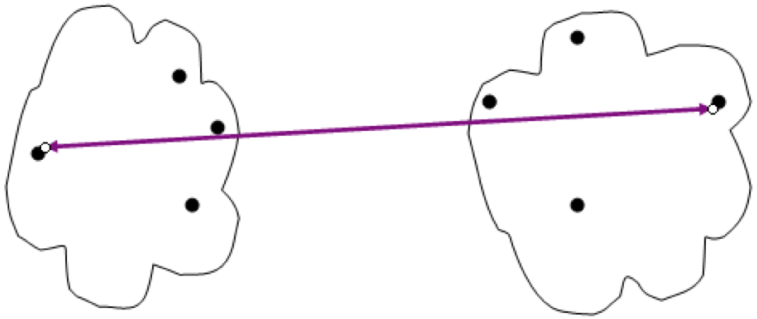
Similarity is used

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



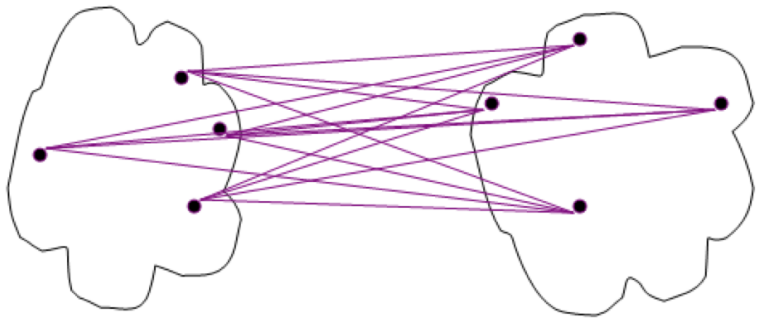
Cluster Similarity: Max

- 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



Cluster Similarity: Group Average

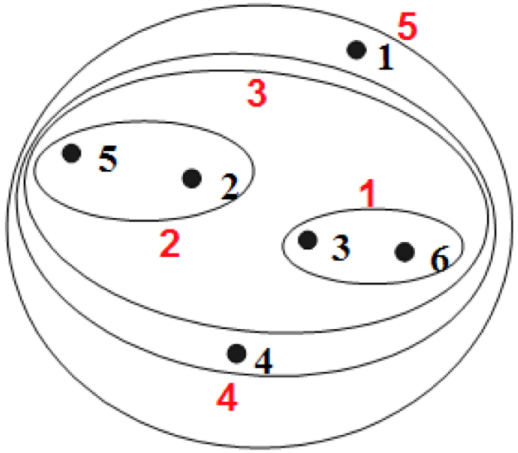
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters
- Need to use average connectivity for scalability since total proximity favors large clusters



Cluster Similarity: Ward's Method

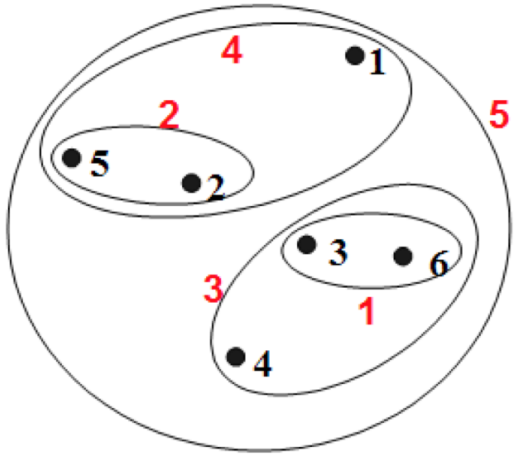
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
- Similar to group average if distance between points is distance squared

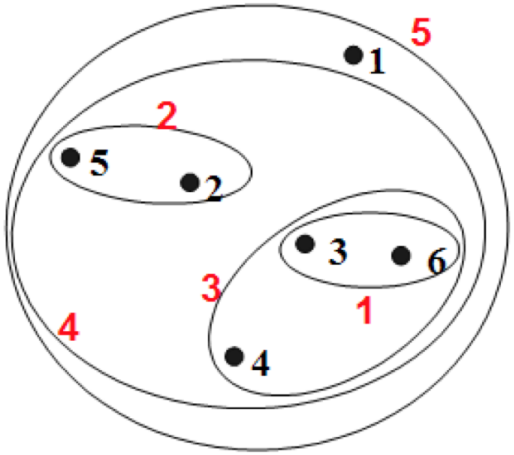
Cluster Similarity: Comparisons



MIN

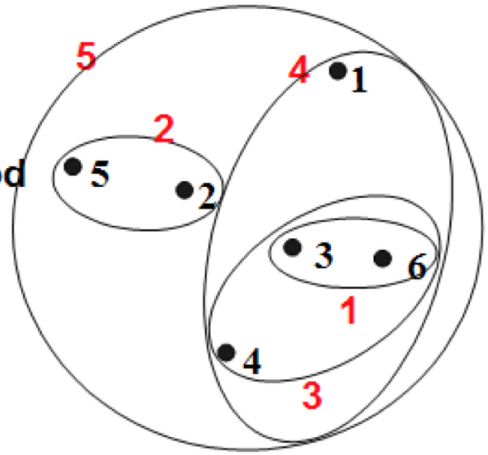
MAX





Group Average

Ward's Method



Partitional clustering

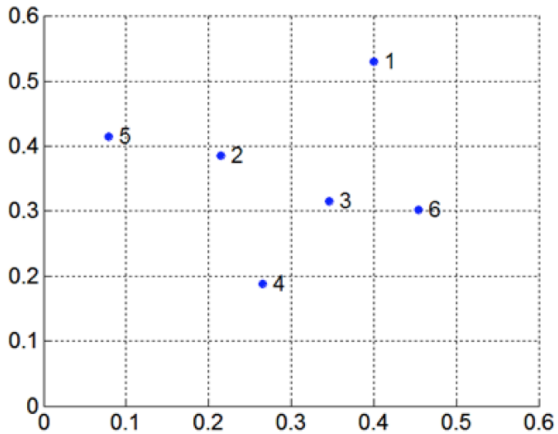
- Min
  - Can handel non-elliptical shapes
  - Sensitive to noise and outliers
- Max
  - Less susceptible to noise and outliers
  - Tends to break large clusters
  - Biased toward global clusters
- Group average/Ward's method
  - Less susceptible to noise and outliers
  - Tends to break large clusters
  - Can be used to initialize K-means

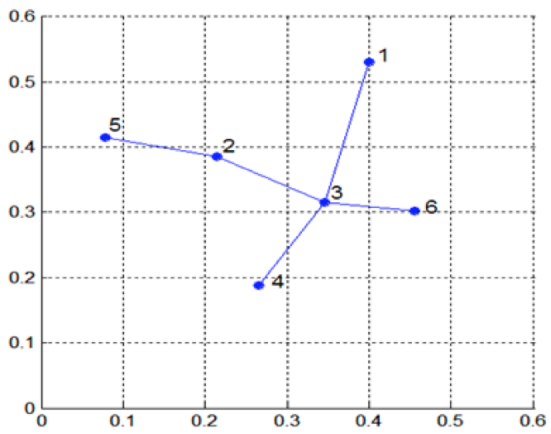
Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficult to handle different sized clusters and convex shapes
  - Breaking large clusters

Divisive Hierarchical Clustering

- Build minimum spanning tree
- Obtain hierarchy of clusters by successively remove the longest edge
- Less popular than agglomerative method





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?
- Still need immediate measurement:
  - To compare clustering algorithms
  - To compare two sets of clusters

Measures of Cluster Validity

- External Index: Used to measure the extent to which cluster labels match externally supplied class labels
  - Entropy
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - Sum of Squared Error (SSE)

External Measures

- Partition obtained: $P = \{P_1, \ldots, P_K\}$
- External (true) partition: $C = \{C_1, \ldots, C_K\}$
- m_{ij} is a number of objects in $P_i \cap C_j$
- Contingency table

		Partition C				$\sum$
		C_1	C_2	$\cdots$	C_K	
Partition P	P_1	n_{11}	n_{12}	$\cdots$	n_{1K}	$n_{1\cdot}$
	P_2	n_{21}	n_{22}	$\cdots$	n_{2K}	$n_{2\cdot}$
	.	.	.	$\cdots$	.	.
	P_K	n_{K1}	n_{K2}	$\cdots$	n_{KK}	$n_{K\cdot}$
	$\sum$	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot K}$	n

External Measures: Entropy

- Distribution of objects in P_i

$$\frac{m_{i1}}{m_{i.}}, \frac{m_{i2}}{m_{i.}}, \ldots, \frac{m_{in}}{m_{i.}}$$

- Entropy (impurity) of the distribution is

$$-\sum_j \frac{m_{ij}}{m_{i.}} \log \frac{m_{ij}}{m_{i.}}$$

- Overall entropy

$$E = -\sum_i \frac{m_{i.}}{m} \sum_j \frac{m_{ij}}{m_{i.}} \log \frac{m_{ij}}{m_{i.}}$$

- The lower the entropy, the better the clustering

External Measures: Mutual Information

- Told: to which P_i an object $\mathbf{x}$ belongs
- The fact tells us something about the true class
- The amount information is measured by

$$MI = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{p_i p_j},$$

  where $p_{ij} = \frac{m_{ij}}{m}$, $p_i = \frac{m_{i\cdot}}{m}$, $p_j = \frac{m_{\cdot j}}{m}$
- Higher mutual information implies a higher clustering quality

External Measures: Jaccard Index

$$J = \frac{a}{a + b + c}$$

- a is the number of pairs of points with the same label in C and assigned to the same cluster in P,
- b is the number of pairs with the same label, but in different clusters, and
- c is the number of pairs in the same cluster, but with different class labels
- The index produces a result in the range $[0, 1]$, where a value of 1.0 indicates that C and P are identical

External Measures: Rand Index

$$J = \frac{a + d}{a + b + c + d}$$

- a, b, c are as above
- d denotes the number of pairs with a different label in C that were assigned to a different cluster in P
- The index produces a result in the range $[0, 1]$, where a value of 1.0 indicates that C and P are identical
- A high value for this measure generally indicates a high level of agreement between a clustering and the true classes

Internal Measure: Dunn's Index

- Clusters: $C_1, \ldots, C_K$
- $d(C_i, C_j)$: distance between clusters C_i and C_j (inter-cluster distance)
- $d'(C_i)$: intra-cluster distance of cluster C_i
- Dunn's index

$$D = \frac{\min_{ij} d(C_i, C_j)}{\max_r d'(C_r)}$$

- Among alternative clusterings, choose one with the max Dun index
- This is to maximize the inter-cluster distances and minimize the intra-cluster distances

Internal Measure: Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE

$$\sum_i \sum_{x \in C_i} \|\mathbf{x} - \mathbf{m}_i\|^2$$

- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
  - Example: Squared Error

$$\sum_i |C_i| \cdot \|\mathbf{m} - \mathbf{m}_i\|^2$$

Internal Measure: Silhouette Coefficient

Consider an i-th individual point

- $a(i) =$ average distance of the i-th point to the points in its cluster
- $b(i) =$ min (average) distance of the i-th to points in another cluster
- The silhouette coefficient for the point is then given by

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- $-1 \leq s(i) \leq 1$

Internal Measure: Silhouette Coefficient

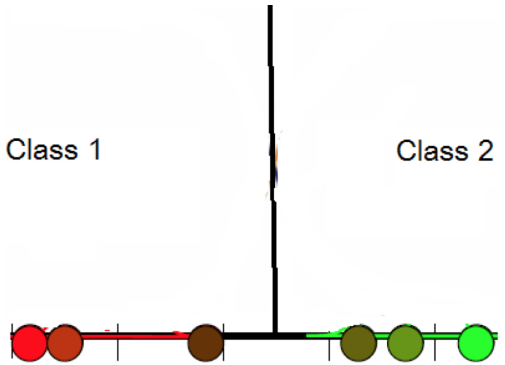
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- If $s(i)$ is close to 1, sample i is well-clustered and it was assigned to a very appropriate cluster
- If $s(i)$ is close to zero, sample i could be assign to another closest cluster as well, and the sample lies equally far away from both clusters
- If $s(i)$ is close to -1, sample i is misclassified
- The average $\frac{\sum_i s(i)}{m}$ of $s(i)$ for all objects in the whole dataset
- The larger it is, the better the clustering

Model based Clustering

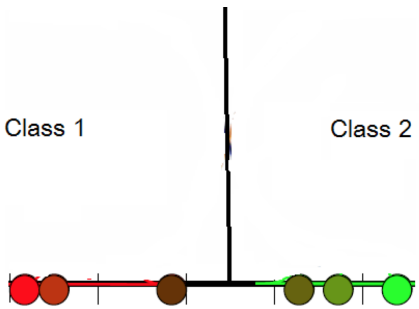
K-means and Hard Assignment

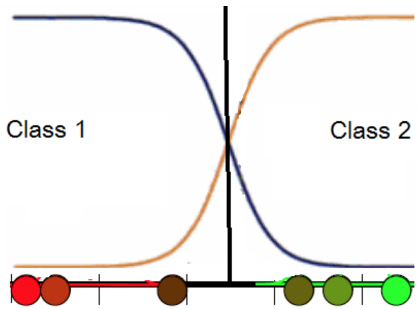
- K-means does hard assignment
- Each data point is assigned to one and only one cluster
- Appropriate for points close to cluster centroids
- Not appropriate to points midway between the two cluster centroids



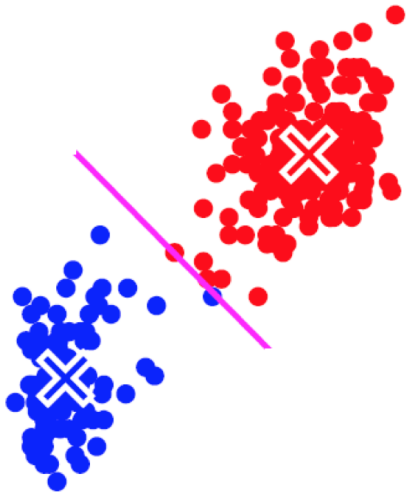
Probabilistic (Soft) Assignment

- Each data point is (partially) assigned to clusters with probabilities
- One point might be (partially) assigned to multiple clusters
- The midway point is assigned to either class with probability 0.5

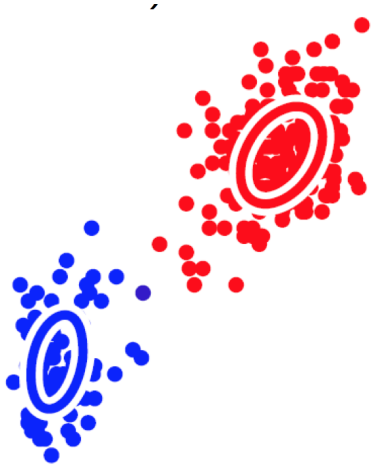




Hard/Soft Assignment: 2D example

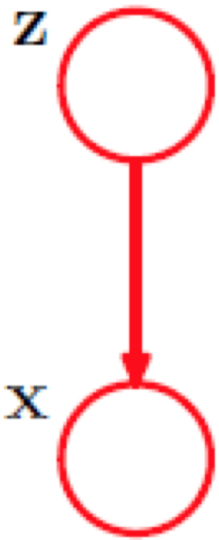


Hard Assignment



Soft Assignment (Eclipses: contour of probability functions). One way achieve soft assignment: mixture models

Mixture Models



- K clusters: $1, 2, \ldots, K$
- Randomly drawn object
  - $\mathbf{x}$: attribute values of the object, observed
  - z: class of the object, latent variable (not observed)
- $P(z)$: distribution of z
  - $\pi_k = P(z = k)$: probability that the object is from class k
- $p(\mathbf{x}|z)$: conditional distribution of attribute values
  - $p(\mathbf{x}|z = k)$: distribution for objects from class k

Mixture Models

- Distribution of data

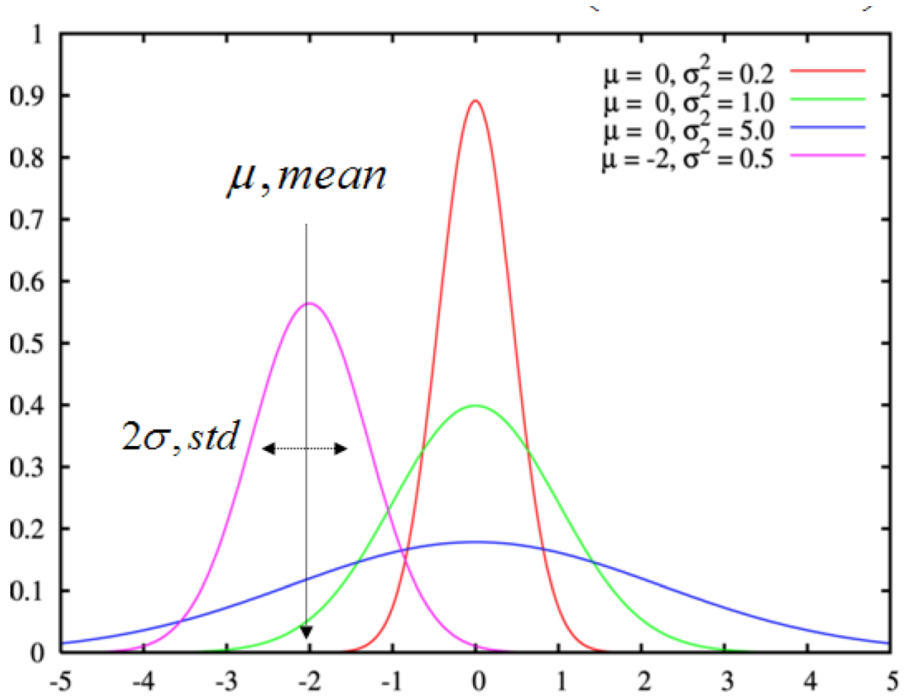
$$p(\mathbf{x}) = \sum_{k=1}^{K} P(z=k)p(\mathbf{x}|z=k) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|z=k)$$

- It is a mixture of the distributions for individual classes
- Each $p(\mathbf{x}|z=k)$ is a component in the mixture
- Gaussian mixtures: each component is a Gaussian distribution

Gaussian Distribution

- Gaussian distribution: $\mathcal{N}(x|\mu, \sigma)$
- Probability density function

$$p(\mathbf{x}|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$



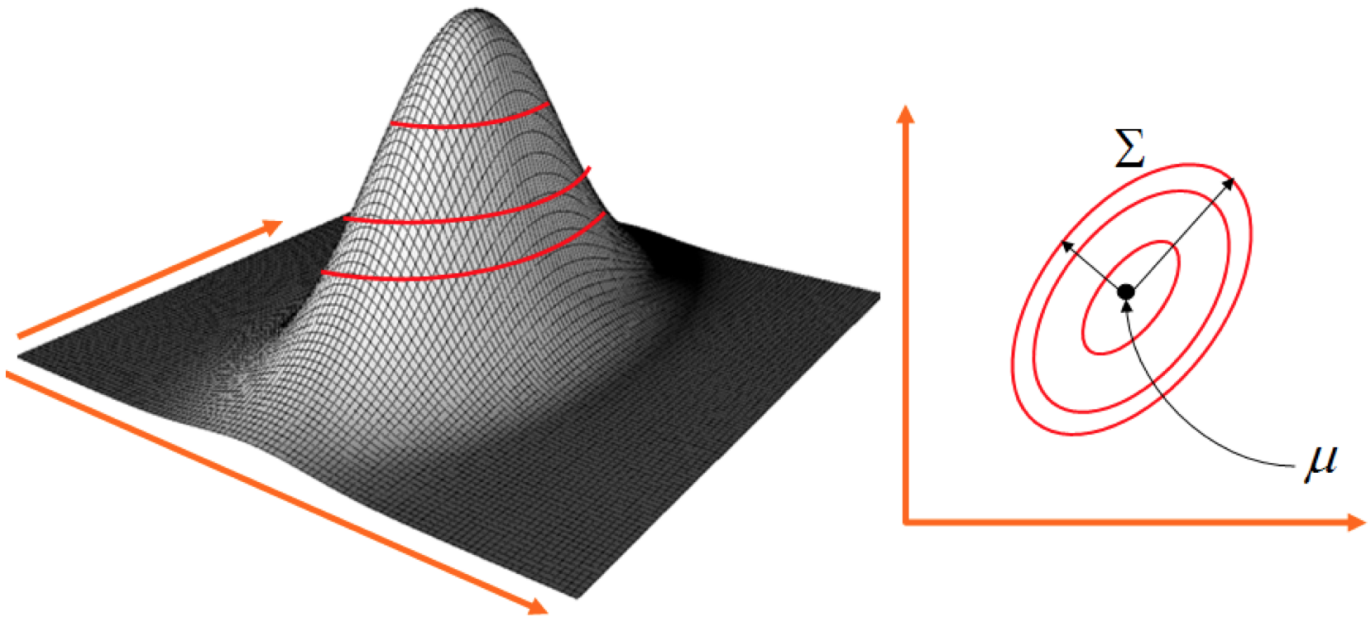
Multivariate Gaussian Distribution

$$p(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^N \det(\boldsymbol{\Sigma})}} \exp\left[-\frac{(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}{2}\right]$$

- N: dimension
- $\mathbf{x}$: vector of N random variables, representing data
- $\boldsymbol{\mu}$: vector of means
- $\boldsymbol{\Sigma}$: covariance matrix

Multivariate Gaussian Distribution

- μ: center of contour lines
- Σ: orientation and size of contour lines



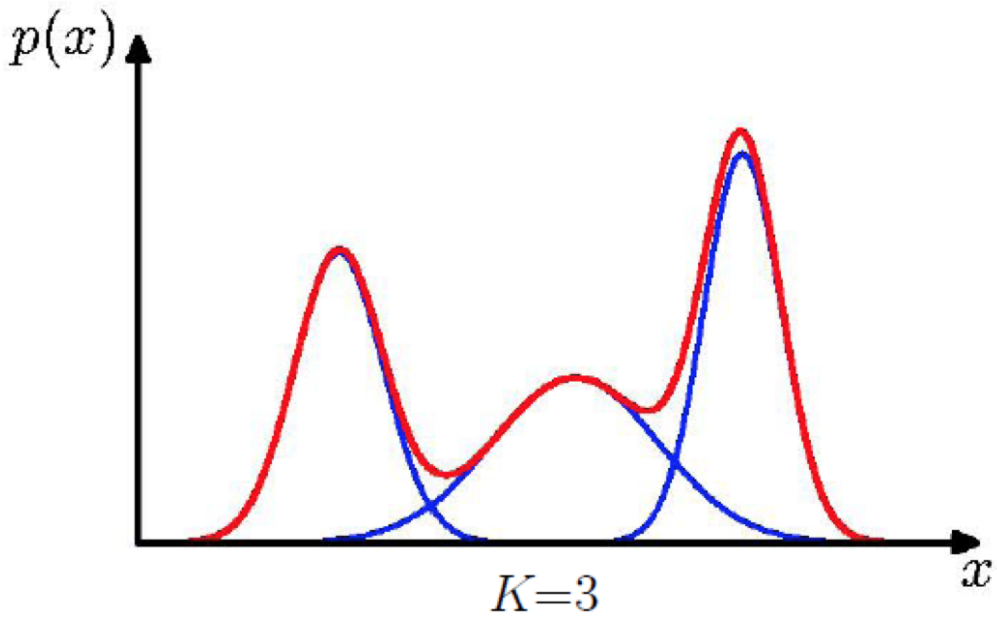
Gaussian Mixture Models

- Mixture distribution

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|z=k)$$

- Each component is a Gaussian distribution

$$p(\mathbf{x}|z=k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



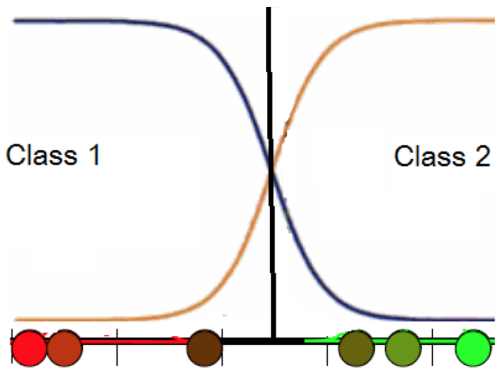
Class Assignment in Mixture Models

- Given mixture model

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(z=k)p(\mathbf{x}|z=k) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|z=k)$$

- Object with features $\mathbf{x}$ belong to class k with probability

$$P(z=k|\mathbf{x}) = \frac{P(z=k)p(\mathbf{x}|z=k)}{p(\mathbf{x})} = \frac{\pi_k p(\mathbf{x}|z=k)}{\sum_{s=1}^{K} \pi_s p(\mathbf{x}|z=s)}$$



Problem Statement

- Given
  - Unlabeled data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$
  - A number of clusters K
- Find a K-component Gaussian mixture model:
  - Mixing coefficients $\{\pi_1, \ldots, \pi_K\}$
  - Components parameters: $\{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^{K}$

  by maximizing data likelihood

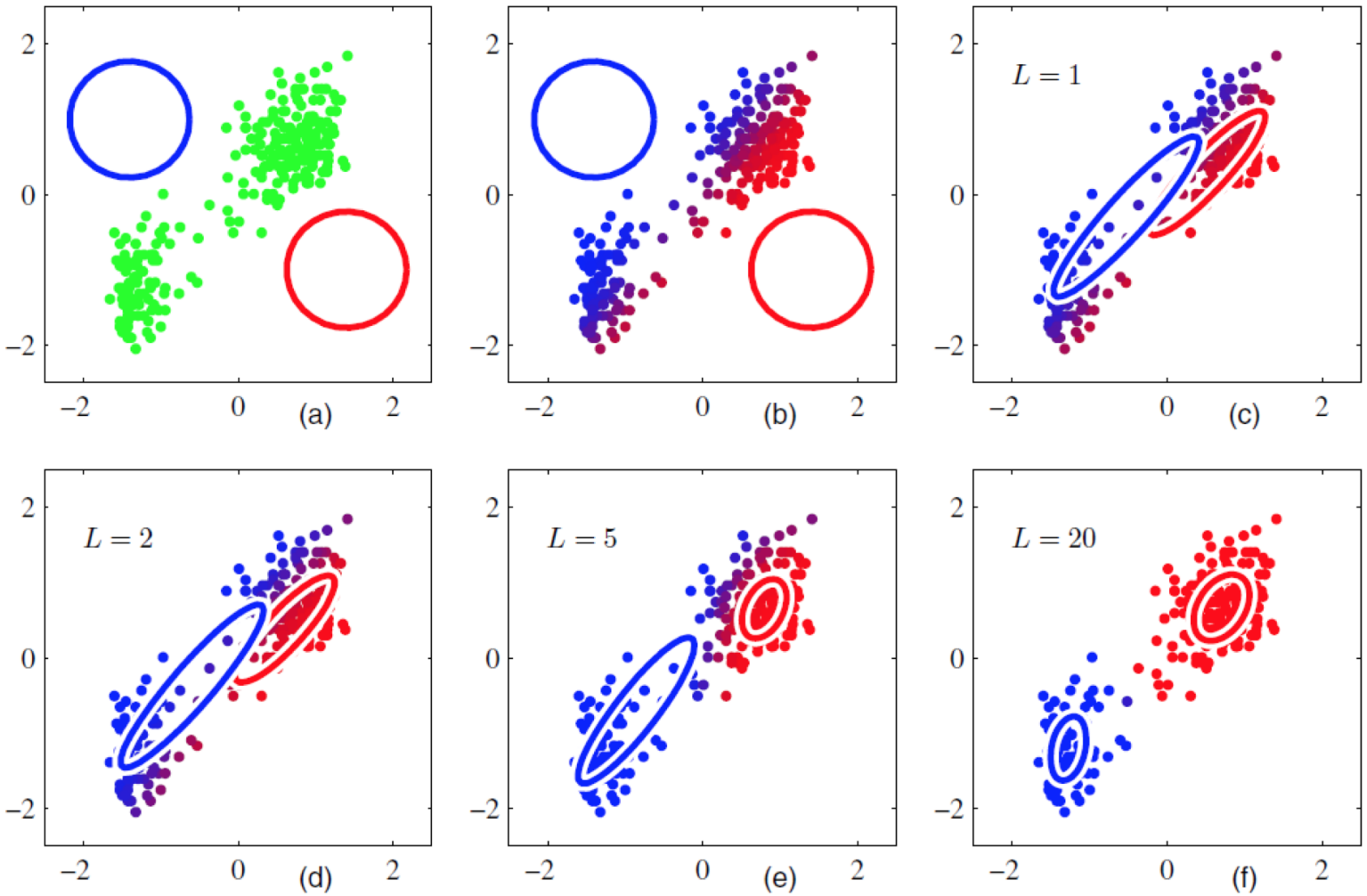
Expectation-Maximization (EM) algorithm

- Choose initial values for $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k,\ k = 1, \ldots, K$
- repeat:
  - Expectation: for each training example $\mathbf{x}_n$
    - (A) Compute $r_{nk} = P(z = k|\mathbf{x}_n),\ k = 1, \ldots, K$
    - (B) Break data into K fractional examples according to the probabilities

$$\mathbf{x}_n[r_{nk}],\ k = 1, \ldots, K$$

    - (C) Assign each fractional example $\mathbf{x}_n[r_{nk}]$ to the corresponding cluster k
  - Maximization: Re-estimate $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k,\ k = 1, \ldots, K$
- Until convergence

EM: 2D illustration



The E-step: comments

- Main computations are in step (a)

$$
\begin{aligned}
r_{nk} &= P(z = k|\mathbf{x}_n) \\
&= \frac{P(z = k)p(\mathbf{x}_n|z = k)}{\sum_{k=1}^{K} P(z = k)p(\mathbf{x}_n|z = k)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{s=1}^{K} \pi_s \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)}
\end{aligned}
$$

- r_{nk} are often called responsibilities

The M-Step: Estimating Parameters

- Fractional Data, assigned to cluster k during the E-step, is: $\mathbf{x}_1[r_{1k}], \mathbf{x}_2[r_{2k}], \ldots, \mathbf{x}_m[r_{mk}]$
- Total number of examples, assigned to cluster k

$$m_k = \sum_{i=1}^{m} r_{ki}$$

- Re-estimate $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$

$$\pi_k^{new} = \frac{m_k}{m},$$

$$\boldsymbol{\mu}_k^{new} = \frac{1}{m_k} \sum_{i=1}^{m} r_{ki} \mathbf{x}_i,$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{m_k} \sum_{i=1}^{m} r_{ki} (\mathbf{x}_i - \boldsymbol{\mu}_k^{new})(\mathbf{x}_i - \boldsymbol{\mu}_k^{new})^{\mathrm{T}}$$

EM Algorithm

- Choose initial values for π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$
- Repeat until convergence
  - Expectation: for each training example $\mathbf{x}_n$ compute

$$r_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{s=1}^{K} \pi_s \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)}, \; k = 1, 2, \ldots, K$$

  - Maximization: Re-estimate π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$

$$\pi_k^{new} = \frac{m_k}{m},$$

$$\boldsymbol{\mu}_k^{new} = \frac{1}{m_k} \sum_{i=1}^{m} r_{ki} \mathbf{x}_i,$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{m_k} \sum_{i=1}^{m} r_{ki} (\mathbf{x}_i - \boldsymbol{\mu}_k^{new})(\mathbf{x}_i - \boldsymbol{\mu}_k^{new})^{\mathrm{T}},$$

where $m_k = \sum_{i=1}^{m} r_{ki}$

EM convergence

- Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$, $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$
- Specification of $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ defines a probability density functions over features $\mathbf{x}$

$$p(\mathbf{x}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Data: $\boldsymbol{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$
- Log-Likelihood as function of model parameters

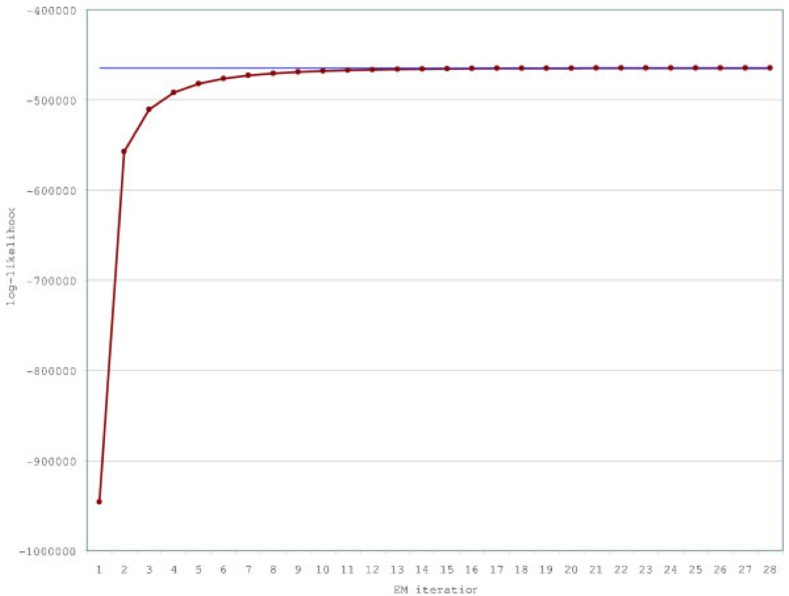
$$l(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{X}) = \log \prod_{i=1}^{m} p(\mathbf{x}_i|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

EM Convergence

- EM aims at computing the maximum likelihood estimation (MLE) of the parameters

$$(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) = \arg\max_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}} \log p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Let $l(t)$ be the log-likelihood after iteration t
- The series $l(1), l(2), \ldots$ increases monotonically with t
- Terminate EM when $l(t+1) - l(t)$ falls below a threshold

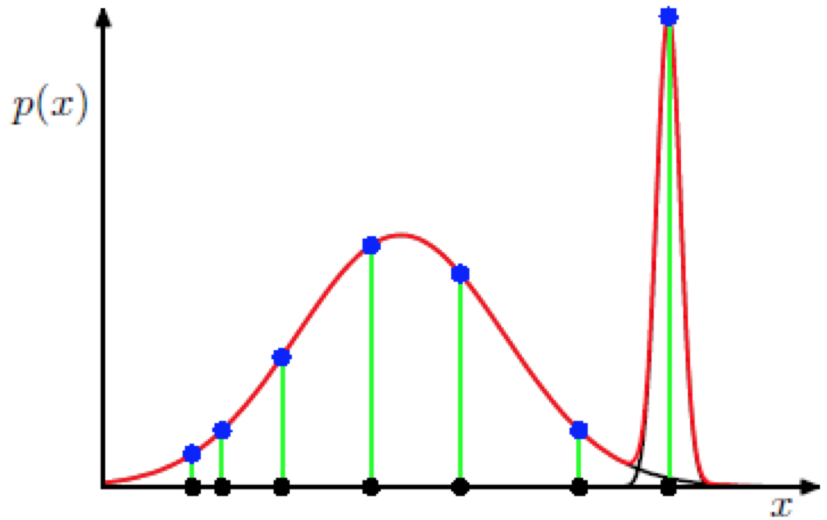


EM Convergence: Singularity

- The maximum log likelihood might be infinite

$$\log p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log \prod_{i=1}^{m} p(\mathbf{x}_i|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Such singularity in likelihood function happens often in case of outliers and repeated points



- Solution: Bound the eigenvalues of covariance matrix
- To avoid local maximum: multiple restart

K-means clustering

Hard Assignment:

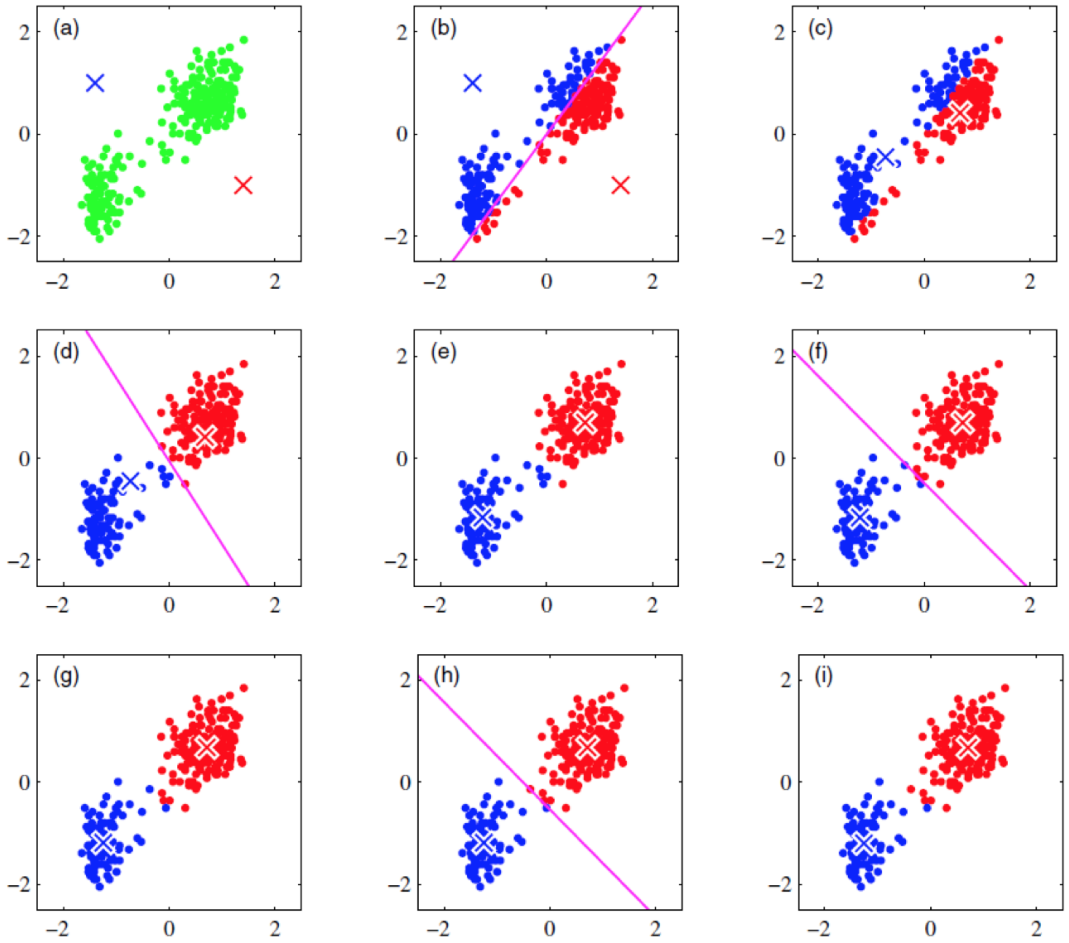
- Select K points as the initial centroids
- repeat
  - For K clusters by assigning all points to the closest centroid
  - Recompute the centroid of each cluster
- until the centroids don't change

Learning GMM

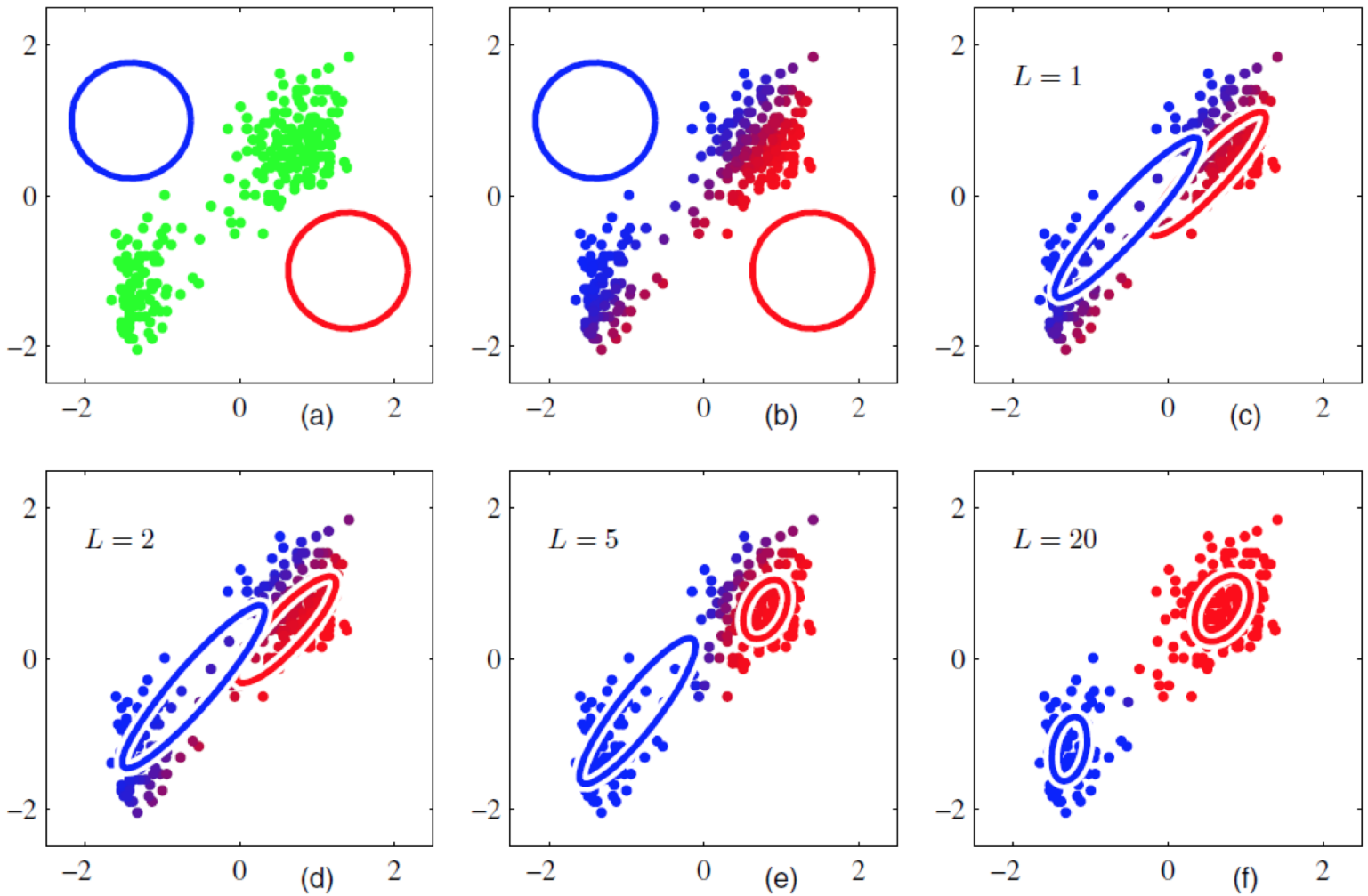
Soft Assignment:

- Choose initial values for π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$
- repeat
  - Expectation:
    (A) Compute $r_{nk} = P(z = k|\mathbf{x}_n)$ for $k = 1, \ldots, K$
    (B) Break it into K fractional examples according to the probabilities
    (C) Assign each fractional examples to the corresponding cluster k
  - Maximization: Re-estimate π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$
- until convergence

K-means example



EM example



Pros and Cons

1. Soft assignment
2. Has global criterion to optimize
3. Obtains Statistical Model: Allow inference among attributes Make prediction about future data Gives density estimation
4. Singularity

1. Hard Assignment
2. Has no global criterion to optimize
3. There is no statistical model
4. No singularity