Introduction to Multivariate Analysis

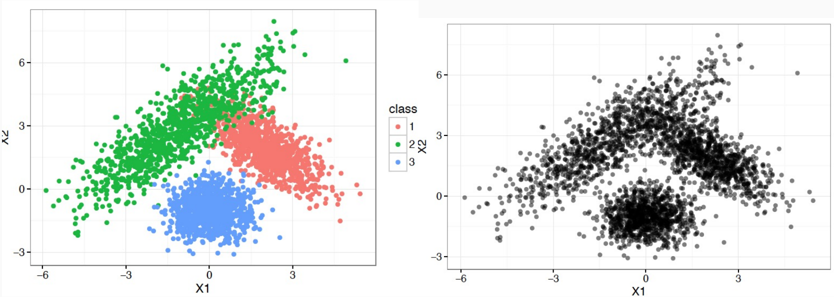
Liu, Xin

Fall, 2021

School of Statistics and Management
Shanghai University of Finance and Economics

Email: liu.xin@mail.shufe.edu.cn

Chapter 4 Clustering


class
1
2
3
X1
X2
-6
-3
0
3
6

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
  - Unsupervised learning: no predefined classes (i.e., learning by observations)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Basic Steps to Develop a Clustering Task

- Feature selection
  - Select info concerning the task of interest
  - Minimal information redundancy
- Proximity measure: Similarity of two feature vectors
- Clustering criterion: Expressed via a cost function or some rules
- Clustering algorithms: Choice of algorithms
- Validation of the results: Validation test (also, clustering tendency test)
- Interpretation of the results
- Integration with applications

§4.1 Partitioning Algorithms

- Partitioning a dataset D of n objects into a set of k clusters, such that the sum of squared distances is minimized

$$J = \sum_{j=1}^{k} \sum_{p \in C_j} \left(d\left(p, c_j\right)\right)^2$$

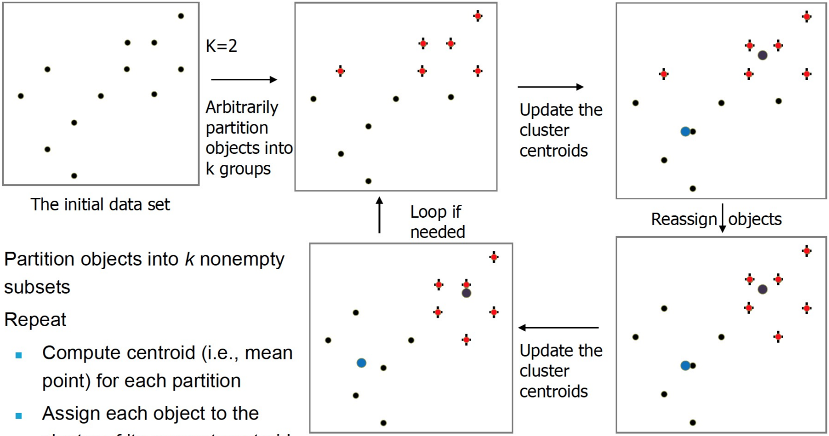
where c_j is the centroid or medoid of cluster C_j. Given k, find a partition of k clusters that optimizes the chosen partitioning criterion

- Global optimal: exhaustively enumerate all partitions

- Heuristic methods: k -means and k -medoids algorithms
  - k-means (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - k-medoids or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in four steps:
  - Step 0 : Partition objects into k nonempty subsets
  - Step 1: Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., mean point, of the cluster)
  - Step 2 : Assign each object to the cluster with the nearest seed point
  - Step 3 : Go back to Step 1 , stop when the assignment does not change



Partition objects into *k* nonempty subsets

Repeat

- Compute centroid (i.e., mean point) for each partition
- Assign each object to the cluster of its nearest centroid

Until no change

Theory Behind K-Means

- Objective function

$$J = \sum_{j=1}^{k} \sum_{C(i)=j} \|x_i - c_j\|^2,$$

  the total within-cluster variance

- Re-arrange the objective function

$$J = \textstyle\sum_{j=1}^{k} \sum_i w_{ij} \|x_i - c_j\|^2 \quad w_{ij} \in \{0, 1\}$$
$$w_{ij} = 1, \text{ if } x_i \text{ belongs to cluster } j;$$
$$w_{ij} = 0, \text{ otherwise}$$

- Looking for: The best assignment w_{ij} and the best center c_j

Solution of K-Means

- Iterations $J = \sum_{j=1}^{k} \sum_i w_{ij} \left\|x_i - c_j\right\|^2$
  - Step 1 : Fix centers c_j, find assignment w_{ij} that minimizes J

$$w_{ij} = 1, \text{ if } \left\|x_i - c_j\right\|^2$$

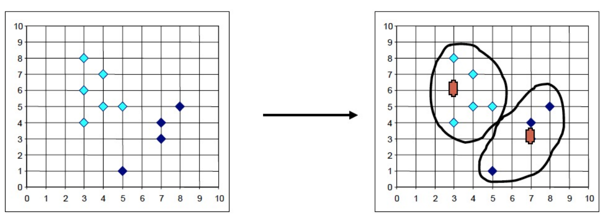
    is the smallest

  - Step 2 : Fix assignment w_{ij}, find centers that minimize J
    - $\frac{\partial J}{\partial c_j} = -2 \sum_i w_{ij} \left(x_i - c_j\right) = 0$
    - $c_j = \frac{\sum_i w_{ij} x_i}{\sum_i w_{ij}}$
    - Note $\sum_i w_{ij}$ is the total number of objects in cluster j

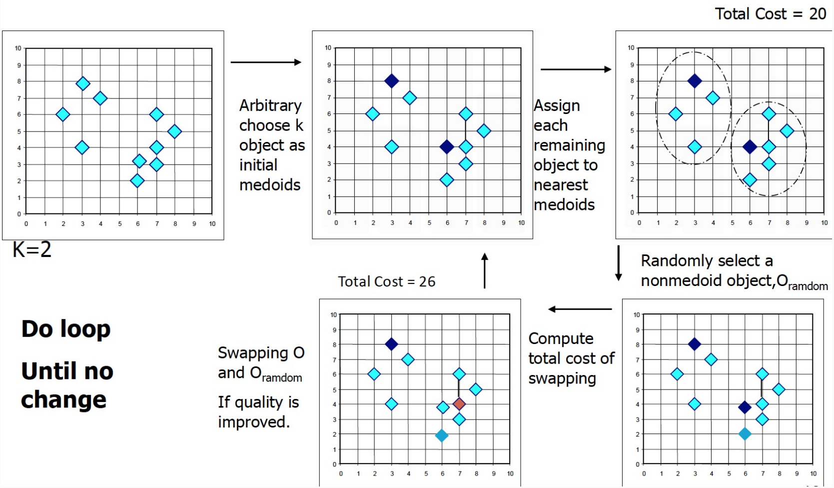
Comments on the K-MeansMethod

- Strength:
  - Efficient: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations.
  - Normally, $k, t << n$
- Weakness
  - Often terminates at a local optimal
  - Applicable only to objects in a continuous n-dimensional space data
  - Need to specify k, the number of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes

- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.



PAM: A Typical K-Medoids Algorithm



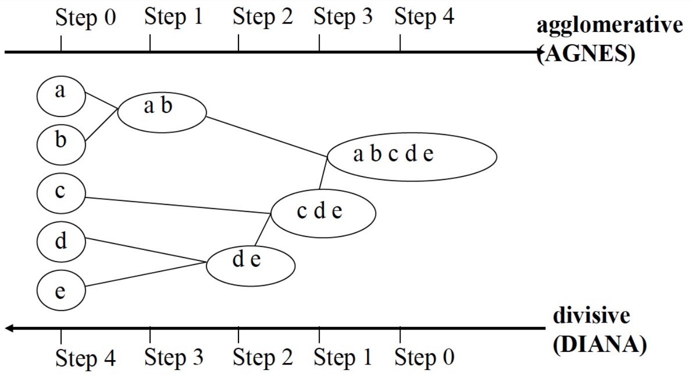
K-Medoids Clustering:

- Find representative objects (medoids) in clusters PAM (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
  - Starts from an initial set of medoids and
  - iteratively replaces one of the medoids by one of the non-medoids
  - if it improves the total distance of the resulting clustering
- PAM works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - CLARA (Kaufmann & Rousseeuw, 1990): PAM on samples
  - CLARANS (Ng & Han, 1994): Randomized re-sampling

§4.2 Hierarchical Methods

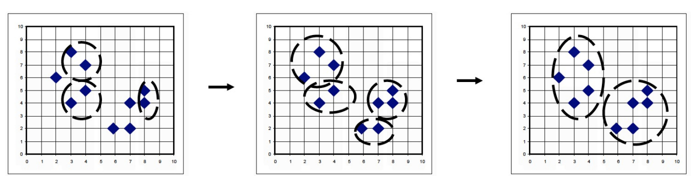
Hierarchical Clustering

- Use distance matrix as clustering criteria.
- This method does not require the number of clusters ***k*** as an input, but needs a termination condition.



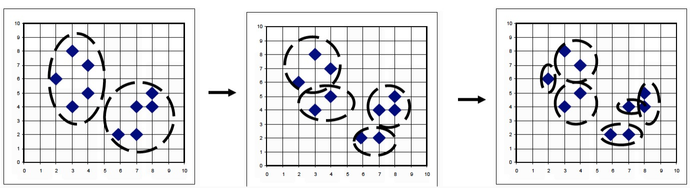
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Use the single-link method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Distance between Clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $\text{dist}\left(\mathrm{K_i}, \mathrm{K_j}\right) = \min \text{dist}\left(\mathrm{t_{ip}}, \mathrm{t_{jq}}\right)$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $\text{dist}\left(\mathrm{K_i}, \mathrm{K_j}\right) = \max \text{dist}\left(\mathrm{t_{ip}}, \mathrm{t_{jq}}\right)$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $\text{dist}\left(\mathrm{K_i}, \mathrm{K_j}\right) = \text{avg } \text{dist}\left(\mathrm{t_{ip}}, \mathrm{t_{jq}}\right)$
- Centroid: distance between the centroids of two clusters, i.e., $\text{dist}\left(\mathrm{K_i}, \mathrm{K_j}\right) = \text{dist}\left(C_i, C_j\right)$
- Medoid: distance between the medoids of two clusters, i.e., dist($\mathrm{K_i}, \mathrm{K_j}) = \text{dist}\left(\mathrm{M_i}, \mathrm{M_j}\right)$

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the "middle" of a cluster

$$C_i = \frac{\sum_{p=1}^{N_i} (t_{ip})}{N_i}$$

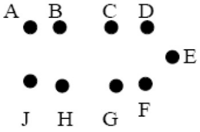
- Radius: square root of average distance from any point of the cluster to its centroid

$$R_i = \sqrt{\frac{\sum_{p=1}^{N_i} (t_{ip} - c_i)^2}{N_i}}$$

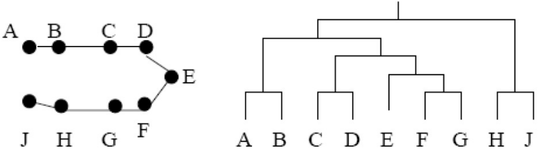
- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_i = \sqrt{\frac{\sum_{p=1}^{N_i} \sum_{q=1}^{N_i} (t_{ip} - t_{iq})^2}{N_i (N_i - 1)}}$$

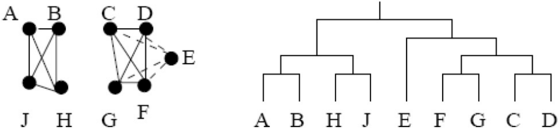
Example: Single Link vs. Complete Link



(a) Data set



(b) Clustering using single linkage



(c) Clustering using complete linkage

Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Do not scalewell: time complexity of at least $O(n^2)$ where n is the number of total objects
- Integration of hierarchical & distance-based clustering
  - BIRCH (1996) : uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CHAMELEON (1999) : hierarchical clustering using dynamic modeling

§4.3 Density-Based Clustering Methods

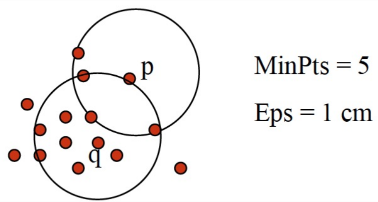
DBSCAN: Basic Concepts

- Two parameters:
  - Eps: Maximum radius of the neighborhood
  - MinPts: Minimum number of points in an Epsneighborhood of that point

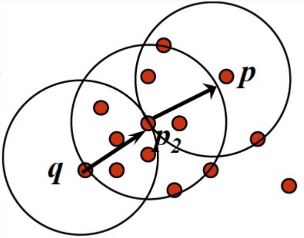
$$N_{\text{Eps}}(q) : \{p \text{ belongs to } D \mid \text{dist}(p, q) \leq Eps\}$$

- Directly density-reachable: A point p is directly densityreachable from a point q w.r.t. Eps, MinPts if
  - p belongs to $N_{Eps}(q)$
  - Core point condition:

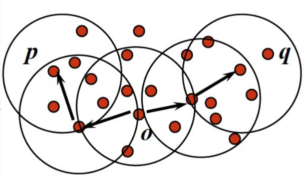
$$\left|N_{Eps}(q)\right| \geq \text{MinPts}$$



- **Density-reachable:** A point p is density-reachable from a point q w.r.t. Eps, MinPts if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i

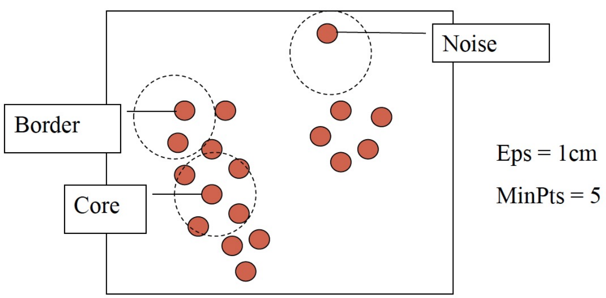


- **Density-connected** A point p is density-connected to a point q w.r.t. Eps, MinPts if there is a point o such that both, p and q are density reachable from o w.r.t. Eps and MinPts



DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- Relies on a density-based notion of cluster: A cluster is defined as a maximal set of density-connected points.
- Noise: object not contained in any cluster is noise
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

(1) mark all objects as `unvisited`;
(2) do
(3)   randomly select an unvisited object p;
(4)   mark p as `visited`;
(5)   if the ϵ-neighborhood of p has at least $MinPts$ objects
(6)   create a new cluster C, and add p to C;
(7)   let N be the set of objects in the ϵ-neighborhood of p;
(8)   for each point p' in N
(9)   if p' is `unvisited`
(10)   mark p' as `visited`;
(11)   if the ϵ-neighborhood of p' has at least $MinPts$ points, add those points to N;
(12)   if p' is not yet a member of any cluster, add p' to C;
(13)   end for
(14)   output C;
(15)   else mark p as `noise`;
(16) until no object is `unvisited`;

DBSCAN: Sensitive to Parameters

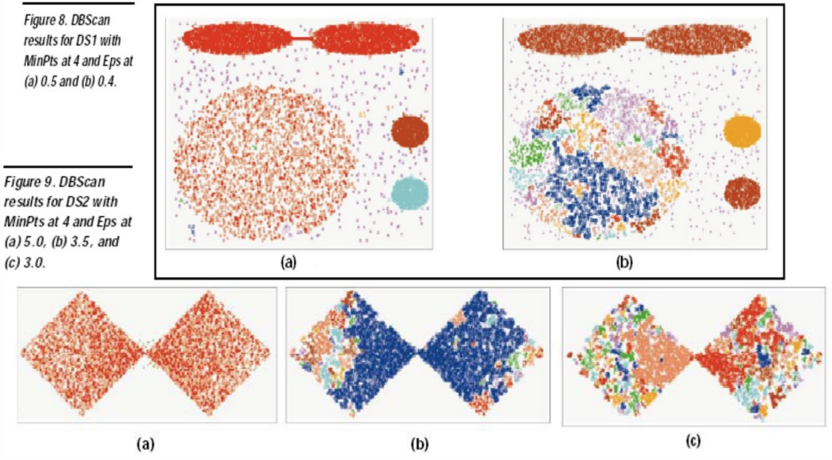


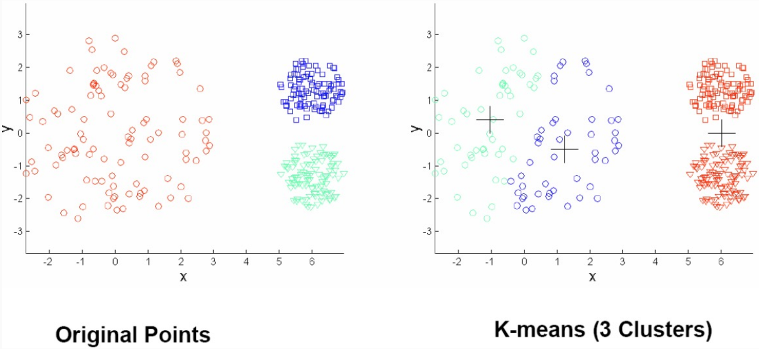
Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

§4.4 Kernel K-means and Gaussian Mixture Model

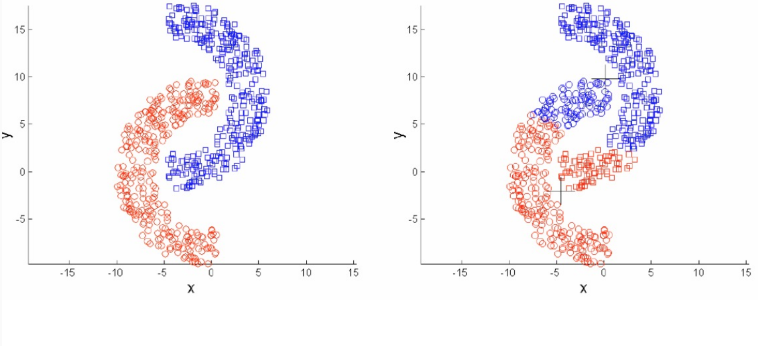
Limitations of K-Means

- K-means has problems when clusters are of different
  - Sizes and density
  - Non-Spherical Shapes



Limitations of K-Means

- K-means has problems when clusters are of different
  - Sizes and density
  - Non-Spherical Shapes

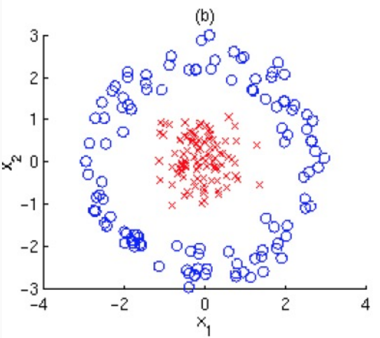


Original Points

K-means (2 Clusters)

Kernel K-Means

- How to cluster the following data?
- A non-linear map: $\phi : R^p \rightarrow F$
  Map a data point into a higher/infinite dimensional space
  $x \rightarrow \phi(x)$
- Dot product matrix K_{ij}
  $K_{ij} = <\phi(x_i), \phi(x_j)>$



- Recall kernel SVM:
  - Polynomial kernel of degree h : $K(\boldsymbol{X}_i, \boldsymbol{X}_j) = (\boldsymbol{X}_i \cdot \boldsymbol{X}_j + 1)^h$
  - Gaussian radial basis function kernel : $K(\boldsymbol{X}_i, \boldsymbol{X}_j) = e^{-\|X_i - X_j\|^2/2\sigma^2}$
  - Sigmoid kernel :

$$K(\boldsymbol{X}_i, \boldsymbol{X}_j) = \tanh(\kappa \boldsymbol{X}_i \cdot \boldsymbol{X}_j - \delta)$$

Solution of Kernel K-Means

- Objective function under new feature space:

$$J = \sum_{j=1}^{k} \sum_{i} w_{ij} \left\| \phi\left(x_i\right) - c_j \right\|^2$$

- Algorithm By fixing assignment w_{ij}

$$c_j = \sum_{i} w_{ij} \phi\left(x_i\right) / \sum_{i} w_{ij}$$

- In the assignment step, assign the data points to the closest center

$$\begin{aligned} d\left(x_k, c_j\right) =& || \phi\left(x_k\right) - \frac{\sum_i w_{ij} \phi\left(x_i\right)}{\sum_i w_{ij}} ||^2 \\ =& \phi\left(x_k\right) \cdot \phi\left(x_k\right) - \\ & 2 \frac{\sum_i w_{ij} \phi\left(x_k\right) \cdot \phi\left(x_i\right)}{\sum_i w_{ij}} + \frac{\sum_i \sum_l w_{ij} w_{lj} \phi\left(x_i\right) \cdot \phi\left(x_l\right)}{\left(\sum_i w_{ij}\right)^2} \end{aligned}$$

Advantages and Disadvantages of Kernel K-Means

- Advantages
  - Algorithm is able to identify the non-linear structures.
- Disadvantages
  - Number of cluster centers need to be predefined.
  - Algorithm is complex in nature and time complexity is large.
- References
  - Kernel k-means and Spectral Clustering by Max Welling.
  - Kernel k-means, Spectral Clustering and Normalized Cut by Inderjit S. Dhillon, Yuqiang Guan and Brian Kulis.
  - An Introduction to kernel methods by Colin Campbell.

Mixture Model-Based Clustering

- A set C of k probabilistic clusters $C_1, \ldots, C_k$
  - probability density functions: $f_1, \ldots, f_k$
  - Cluster prior probabilities: $w_1, \ldots, w_k$, $\Sigma_j w_j = 1$
- Joint Probability of an object i and its cluster C_j is: $P\left(x_i, z_i = C_j\right) = w_j f_j\left(x_i\right)$
- Probability of i is: $P\left(x_i\right) = \sum_j w_j f_j\left(x_i\right)$

Maximum Likelihood Estimation

- Objects are assumed to be generated independently, for a data set $D = \{x_1, \ldots, x_n\}$ we have,

$$P(D) = \prod_i P(x_i) = \prod_i \sum_j w_j f_j(x_i)$$
$$\Rightarrow \log P(D) = \sum_i \log P(x_i) = \sum_i \log \sum_j w_j f_j(x_i)$$

- Task: Find a set C of k probabilistic clusters s.t. $P(D)$ is maximized

The EM (Expectation Maximization) Algorithm

- E-step assigns objects to clusters according to the current fuzzy clustering or parameters of probabilistic clusters

$$w_{ij}^{t+1} = p\left(z_i = j \mid \theta_j^t, x_i\right) \propto p\left(x_i \mid z_i = j, \theta_j^t\right) p\left(z_i = j\right)$$

- M-step finds the new clustering or parameters that maximize the expected likelihood, with respect to conditional distribution $p\left(z_i = j \mid \theta_j^t, x_i\right)$

$$\theta^{t+1} = \mathrm{argmax}_\theta \sum_i \sum_j w_{ij}^{t+1} \log L\left(x_i, z_i = j \mid \theta\right)$$

Gaussian mixtures

- Generative model
  - For each object: Pick its distribution component: $Z \sim \text{Multi}\left(w_1, \ldots, w_k\right)$
  - Sample a value from the selected distribution: $X \sim N\left(\mu_Z, \sigma_Z^2\right)$

- Overall log likelihood function is

$$L(D;\theta) = \sum_i \log \sum_j w_j p\left(x_i \mid \mu_j, \sigma_j^2\right)$$

Considering the first derivative of μ_j

$$\begin{aligned}
\frac{\partial L}{\partial \mu_j} &= \sum_i \frac{w_j}{\sum_j w_j p\left(x_i \mid \mu_j, \sigma_j^2\right)} \frac{\partial p\left(x_i \mid \mu_j, \sigma_j^2\right)}{\partial \mu_j} \\
&= \sum_i \frac{w_j p\left(x_i \mid \mu_j, \sigma_j^2\right)}{\sum_j w_j p\left(x_i \mid \mu_j, \sigma_j^2\right)} \frac{1}{p\left(x_i \mid \mu_j, \sigma_j^2\right)} \frac{\partial p\left(x_i \mid \mu_j, \sigma_j^2\right)}{\partial \mu_j} \\
&\triangleq \sum_i w_{ij} \frac{\partial \log p\left(x_i \mid \mu_j, \sigma_j^2\right)}{\partial \mu_j},
\end{aligned}$$

where $w_{ij} = P\left(Z = j \mid X = x_i, \theta\right)$.

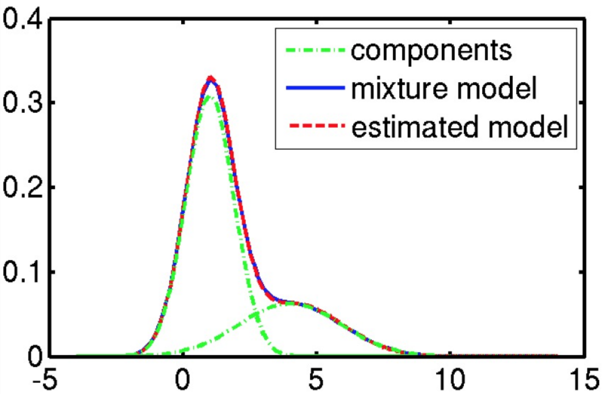
Apply EM algorithm: 1-d

- An iterative algorithm (at iteration $\mathrm{t}+1$)
- E(expectation)-step Evaluate the weight w_{ij} when μ_j, σ_j, w_j are given

$$w_{ij}^{t+1} = \frac{w_j^t p\left(x_i \mid \mu_j^t, \left(\sigma_j^2\right)^t\right)}{\sum_j w_j^t p\left(x_i \mid \mu_j^t, \left(\sigma_j^2\right)^t\right)}$$

- M(maximization)-step Evaluate μ_j, σ_j, w_j when w_{ij} 's are given that maximize the weighted likelihood It is equivalent to Gaussian distribution parameter estimation when each point has a weight belonging to each distribution

$$\mu_j^{t+1} = \frac{\sum_i w_{ij}^{t+1} x_i}{\sum_i w_{ij}^{t+1}}; \left(\sigma_j^2\right)^{t+1} = \frac{\sum_i w_{ij}^{t+1} \left\|x_i - \mu_j^t\right\|^2}{\sum_i w_{ij}^{t+1}}; w_j^{t+1} \propto \sum_i w_{ij}^{t+1}$$


components
mixture model
estimated model
0.4
0.3
0.2
0.1
0
-5
0
5
10
15

Apply EM algorithm: 2-d

- E(expectation)-step Evaluate the weight w_{ij} when $\boldsymbol{\mu}_j, \Sigma_j, w_j$ are given

$$w_{ij}^{t+1} = \frac{w_j^t p\left(x_i \mid \mu_j^t, \Sigma_j^t\right)}{\sum_j w_j^t p\left(x_i \mid \mu_j^t, \Sigma_j^t\right)}$$

- M(maximization)-step Evaluate $\boldsymbol{\mu}_j, \Sigma_j, w_j$ when w_{ij} 's are given that maximize the weighted likelihood

$$\boldsymbol{\mu}_j^{t+1} = \frac{\sum_i w_{ij}^{t+1} x_i}{\sum_i w_{ij}^{t+1}};$$

$$\left(\sigma_{j,1}^2\right)^{t+1} = \frac{\sum_i w_{ij}^{t+1} ||x_{i,1} - \mu_{j,1}^t|||^2}{\sum_i w_{ij}^{t+1}}; \left(\sigma_{j,2}^2\right)^{t+1} = \frac{\sum_i w_{ij}^{t+1} ||x_{i,2} - \mu_{j,2}^t\|^2}{\sum_i w_{ij}^{t+1}};$$

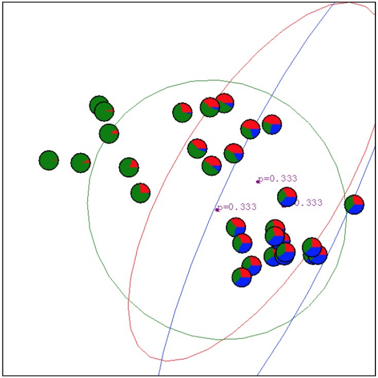
$$\left(\sigma\left(X_1, X_2\right)_j\right)^{t+1} = \frac{\sum_i w_{ij}^{t+1}\left(x_{i,1} - \mu_{j,1}^t\right)\left(x_{i,2} - \mu_{j,2}^t\right)}{\sum_i w_{ij}^{t+1}}; w_j^{t+1} \propto \sum_i w_{ij}^{t+1}$$

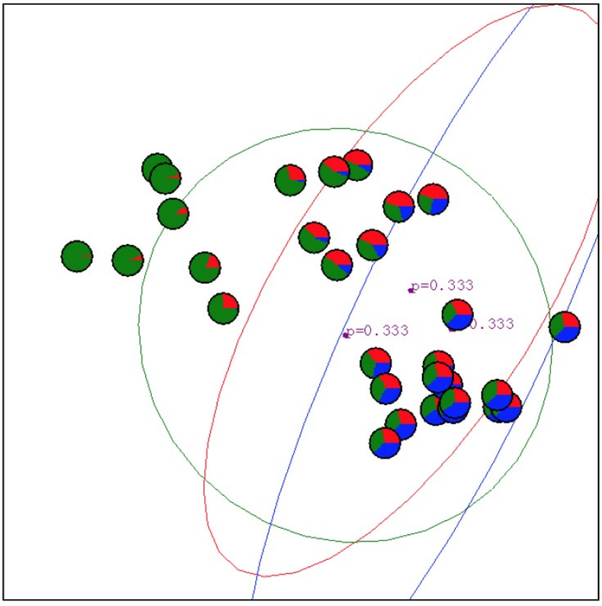
K-Means: A Special Case of Gaussian Mixture Model

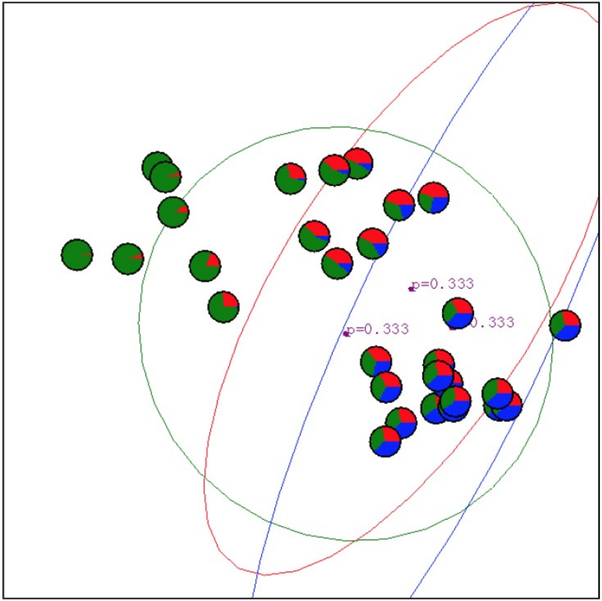
- When each Gaussian component with covariance matrix $\sigma^2 \boldsymbol{I}$
  - Soft K-means $\cdot w_{ij} = p\left(x_i \mid \mu_j, \sigma^2\right) w_j \propto \exp\left\{-\frac{\left(x_i - \mu_j\right)^2}{2\sigma^2}\right\} w_j$
  - When $\sigma^2 \to 0$
    - Soft assignment becomes hard assignment
    - $w_{ij} \to 1,$ if x_i is closest to μ_j

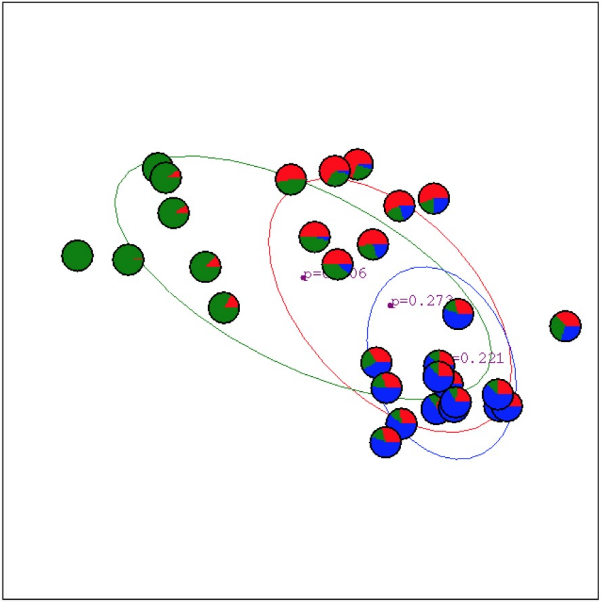
Example

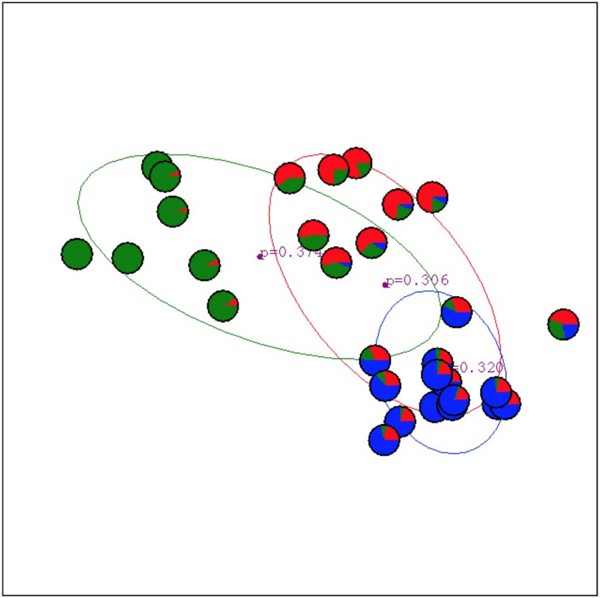
- Each circle (mini pie-chart) is an observation
- Large ovals in the background represent initial $\hat{\mu}_k, \hat{\Sigma}_k$ $\hat{\pi}_k = 1/3$ for all 3 classes
- Pie chart segments correspond to responsibilities estimates from current $\hat{\mu}_k, \hat{\Sigma}_k, \hat{\pi}_k$

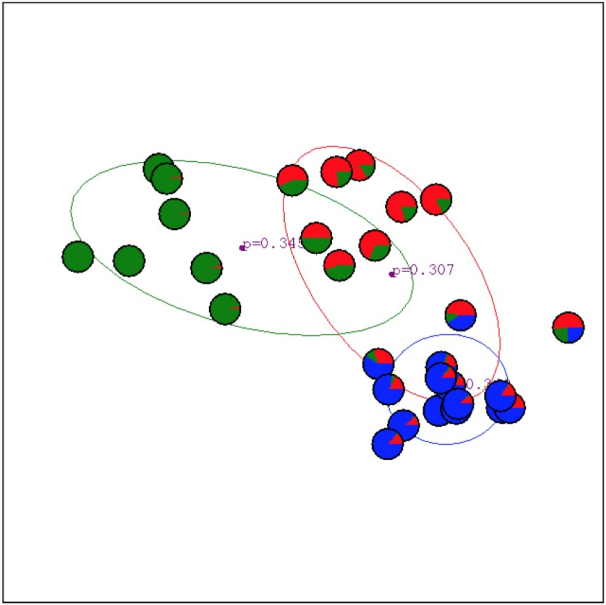


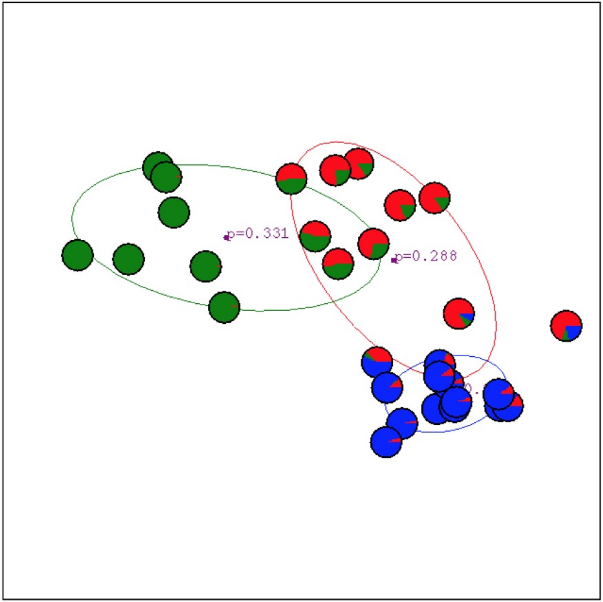

p=0.333
p=0.333
0.333

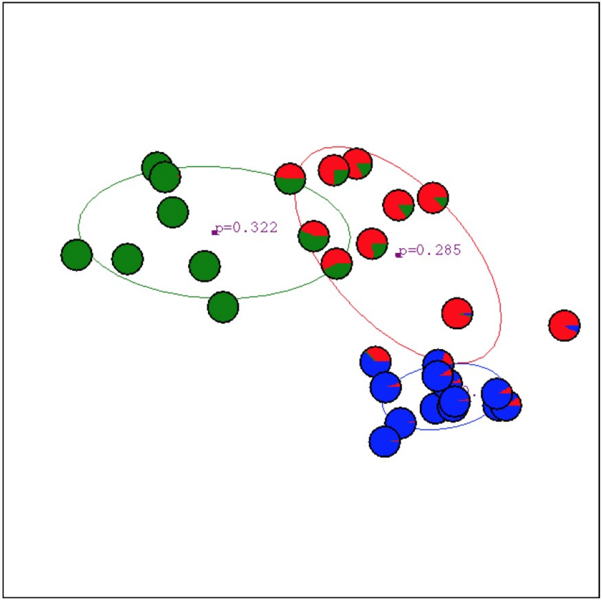

p=0.333
p=0.333
0.333

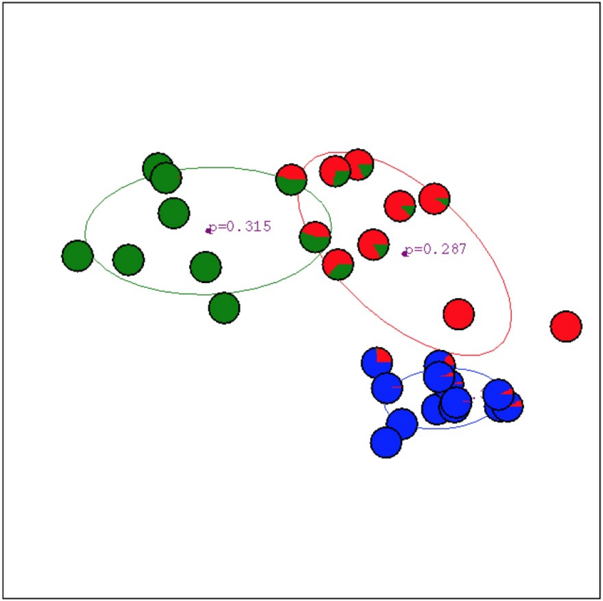


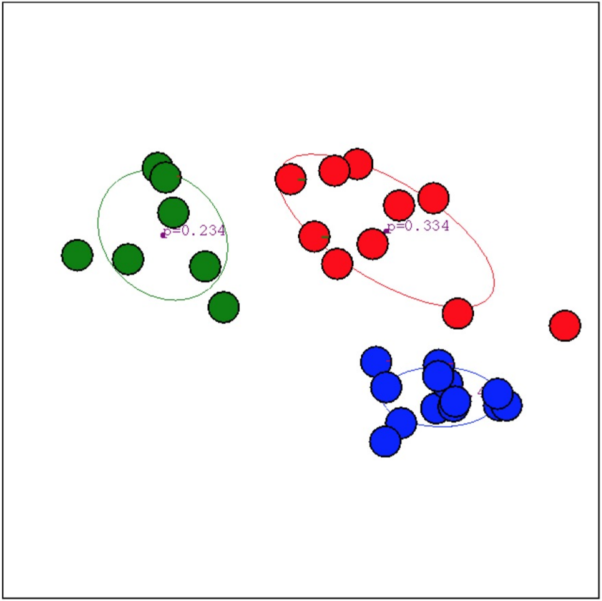

p=0.374
p=0.306
=0.320

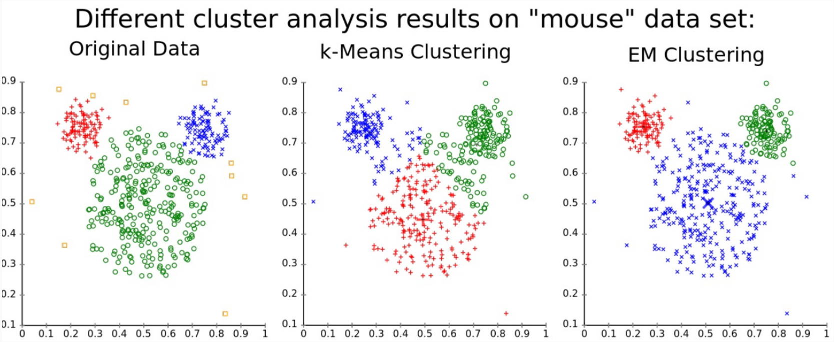

p=0.307


p=0.331
p=0.288


p=0.322
p=0.285


p=0.315
p=0.287


p=0.234
p=0.334


Different cluster analysis results on "mouse" data set:
Original Data
k-Means Clustering
EM Clustering