Statistical Methods in Artificial Intelligence
CSE471 - Monsoon 2015 : Lecture 20

Avinash Sharma

CVIT, IIIT Hyderabad

Lecture Plan

- Revision from Previous Lecture
- Kmeans Clustering
- Probabilistic Kmeans Clustering (GMM fitting)
- Variants of K-means Clustering
  - Fuzzy Kmeans
  - Kernel Kmeans
  - Kmedians and Kmedoids (Self Study/ Tutorial)

Introduction to Data Clustering

- Given a set of points, with a notion of distance between points, group the points into some number of clusters, so that
  - **Members of a cluster are close/similar to each other.**
  - **Members of different clusters are dissimilar.**

- Clustering is generally an ***unsupervised learning*** task as it attempts to recover the natural grouping of the data.

- Typically:
  - Points are sampled in a high dimensional space.
  - Generative Model assumption (with clusters having identical model parameters) rarely holds.

Similarity Measures

- Vectors: Cosine distance.

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{\|\mathbf{x}\| \ \|\mathbf{x}'\|}$$

- Sets: Jaccard distance.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

(If A and B are both empty, we define $J(A,B) = 1$.)

$$0 \leq J(A,B) \leq 1.$$

- Points: Minkowski distance
  - q=2: Euclidean distance
  - q=1: City-block distance

$$d(\mathbf{x}, \mathbf{x}') = \left(\sum_{k=1}^{d} |x_k - x'_k|^q \right)^{1/q},$$

- Points: Mahalanobis metric
  - Data dependent

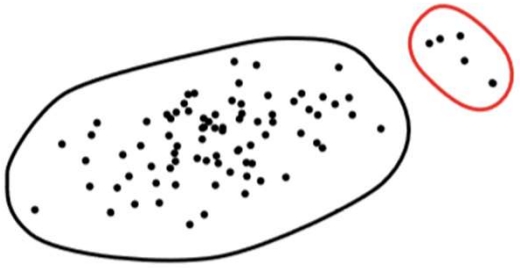
$$d(\mathbf{x}, \mathbf{y})^2 = (\mathbf{x} - \mathbf{y})^T \mathrm{S}^{-1} (\mathbf{x} - \mathbf{y})$$

Criterion Functions for Clustering

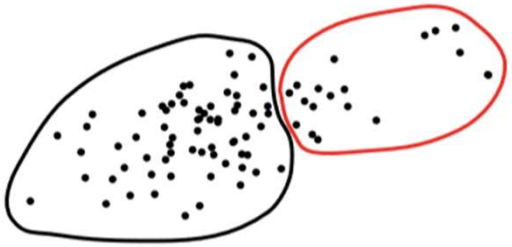
- The Sum-of-Squared-Error Criterion:
  - Achieves minimum variance clustering

$$J_e = \sum_{i=1}^{c} \sum_{\mathbf{x} \in \mathcal{D}_i} \|\mathbf{x} - \mathbf{m}_i\|^2. \qquad \mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}.$$

  - Not always best criterion



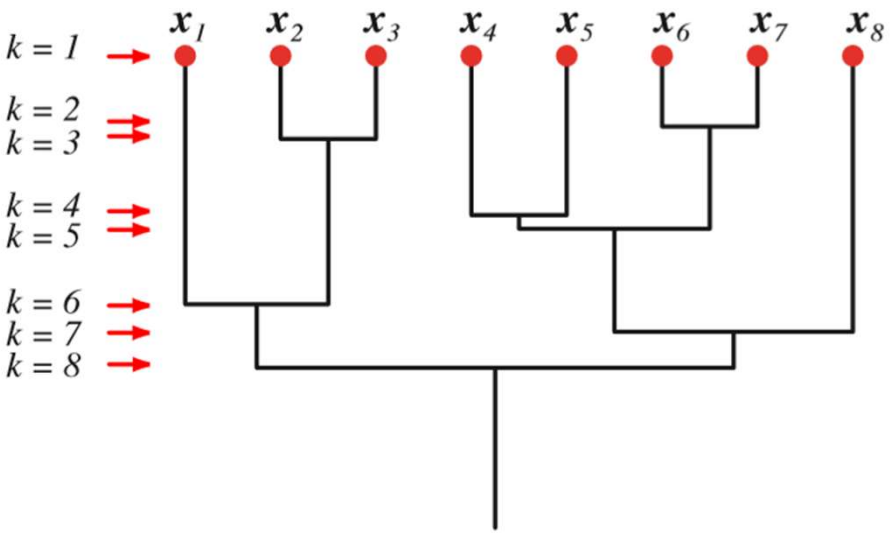
J_e = *large*

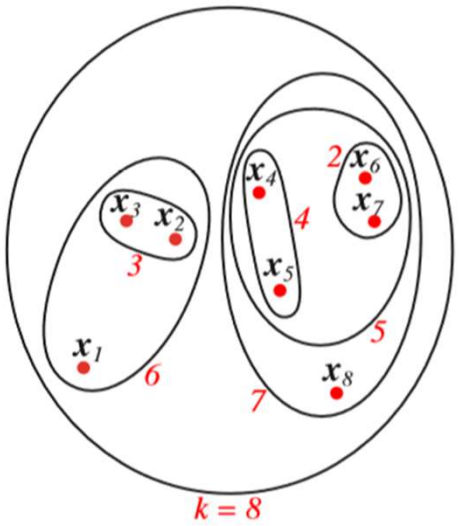


J_e = *small*

Hierarchical Clustering

- Combining two points/clusters at a time based on nearness of points/clusters until a fix number of clusters are remained as long as
  - *any two points put into a single cluster remains in the same cluster all the way till final solution.*





Agglomerative Clustering

- Agglomerative clustering is a bottom-up procedure that combines nearest cluster in each iteration until desired number of clusters are obtained.

Algorithm 4 (Agglomerative hierarchical clustering)

1 **begin initialize** $c, \hat{c} \leftarrow n, \mathcal{D}_i \leftarrow \{\mathbf{x}_i\}, i = 1, \ldots, n$
2 **do** $\hat{c} \leftarrow \hat{c} - 1$
3 Find nearest clusters, say, $\mathcal{D}_i$ and $\mathcal{D}_j$
4 Merge $\mathcal{D}_i$ and $\mathcal{D}_j$
5 **until** $c = \hat{c}$
6 **return** c clusters
7 **end**

Agglomerative Clustering

- The measures of distance between clusters:

$$
\begin{aligned}
d_{min}(\mathcal{D}_i, \mathcal{D}_j) &= \min_{\substack{\mathbf{x} \in \mathcal{D}_i \\ \mathbf{x}' \in \mathcal{D}_j}} \|\mathbf{x} - \mathbf{x}'\| \\
d_{max}(\mathcal{D}_i, \mathcal{D}_j) &= \max_{\substack{\mathbf{x} \in \mathcal{D}_i \\ \mathbf{x}' \in \mathcal{D}_j}} \|\mathbf{x} - \mathbf{x}'\| \\
d_{avg}(\mathcal{D}_i, \mathcal{D}_j) &= \frac{1}{n_i n_j} \sum_{\mathbf{x} \in \mathcal{D}_i} \sum_{\mathbf{x}' \in \mathcal{D}_j} \|\mathbf{x} - \mathbf{x}'\| \\
d_{mean}(\mathcal{D}_i, \mathcal{D}_j) &= \|\mathbf{m}_i - \mathbf{m}_j\|.
\end{aligned}
$$

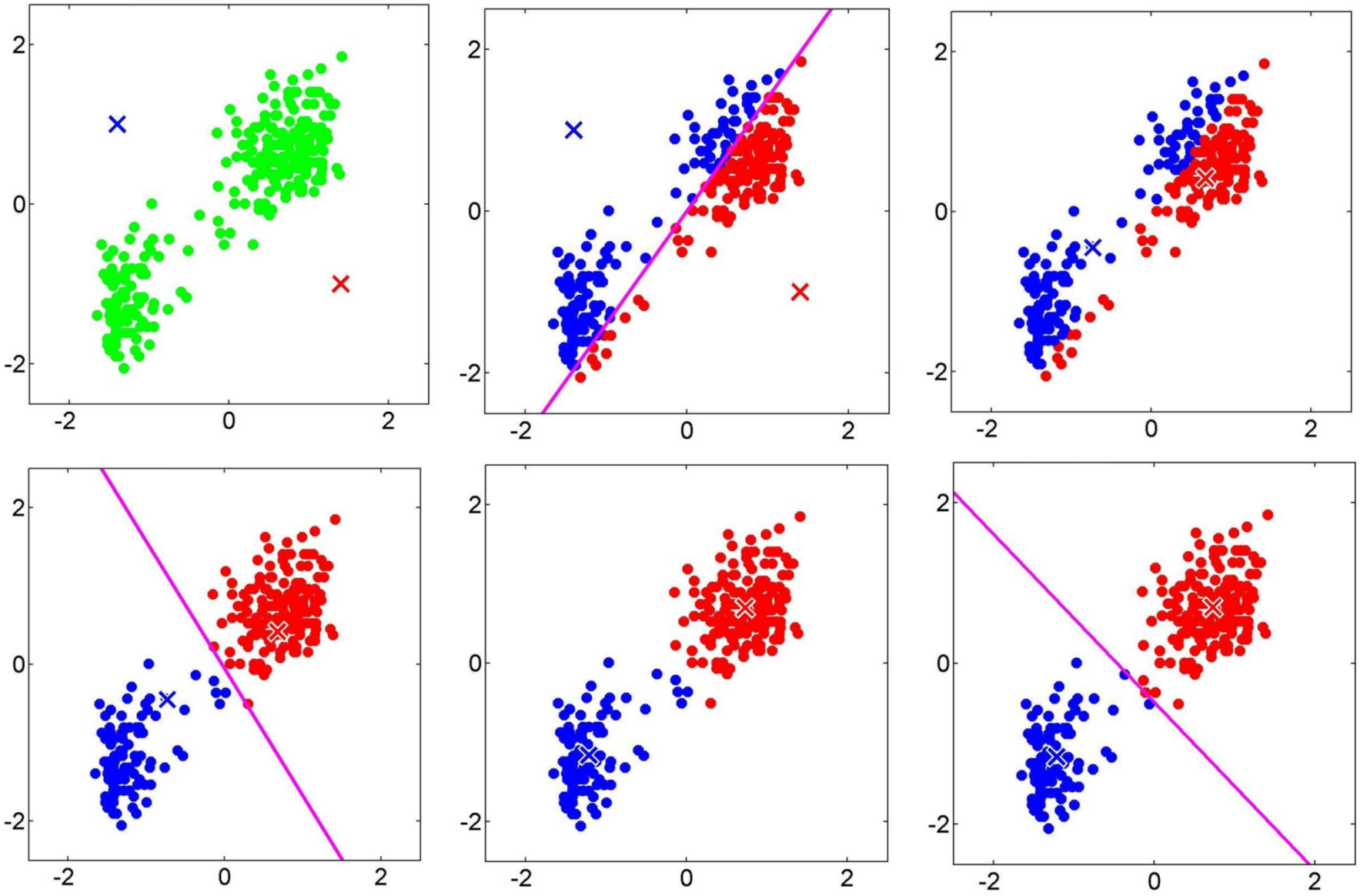
Kmeans Clustering

- Goal is to represent a data set in terms of K clusters using the respective cluster means

- Initialize means randomly
- Iterate between two phases:
  - E-step: assign each data point to nearest mean
  - M-step: update cluster means

- Simplest version is based on Euclidean distance

Kmeans Clustering



Kmeans Clustering

Minimize $J = \sum_{i=1}^{N} \sum_{j=1}^{K} a_{ij} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$

such that $a_{ij} \epsilon \{0,1\}$ and $\sum_{j=1}^{K} a_{ij} = 1$

- Initialize the K mean-vector $\boldsymbol{\mu}_j$ randomly (e.g., choosing any K data points as the mean vectors)

- E-step: minimize J w.r.t. a_{ij}
  - Set $a_{ij} = 1$ for cluster index j corresponding to the smallest $\left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$ i.e., closes cluster mean (centroid)

- M-step: minimize J w.r.t. $\boldsymbol{\mu}_j$
  - Set $\frac{\partial J}{\partial \boldsymbol{\mu}_j} = 0 \Rightarrow \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} a_{ij} \mathbf{x}_i}{\sum_{i=1}^{N} a_{ij}}$ i.e., re-computing the mean.

Limitations of Kmeans Clustering

- Convergence to local minima –sensitive to initialization.
- Applicable to data when mean is defined.
- Sensitive to outliers and data noise
- Suitable only to cases when clusters are convex shapes.
- Number of clusters (K) needs to be explicitly set.
- Computational complexity ($O(NKdL)$).

Probabilistic Kmeans

- Gaussian mixture models (GMM) trained with expectation-maximization (EM) algorithm implements:
  - Probabilistic assignments to clusters.
  - Multivariate Gaussian distributions instead of means.

- Representing the probability distribution of the data as a Gaussian mixture model enables
  - Capturing the uncertainty in the cluster assignments
  - giving model for data distribution
  - determining K using Bayesian mixture model (not covered here)

Probabilistic Kmeans

- Maximum Likelihood Estimation of Multivariate Gaussian distribution with unknown $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

$$\ln\, p(\mathbf{x}_k|\boldsymbol{\mu}) = -\frac{1}{2}\ln\left[(2\pi)^d|\boldsymbol{\Sigma}|\right] - \frac{1}{2}(\mathbf{x}_k-\boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}_k-\boldsymbol{\mu})$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k \qquad \widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k-\hat{\boldsymbol{\mu}})(\mathbf{x}_k-\hat{\boldsymbol{\mu}})^t.$$

Probabilistic Kmeans

Gaussian Mixture Model

- Linear super-position of Gaussians

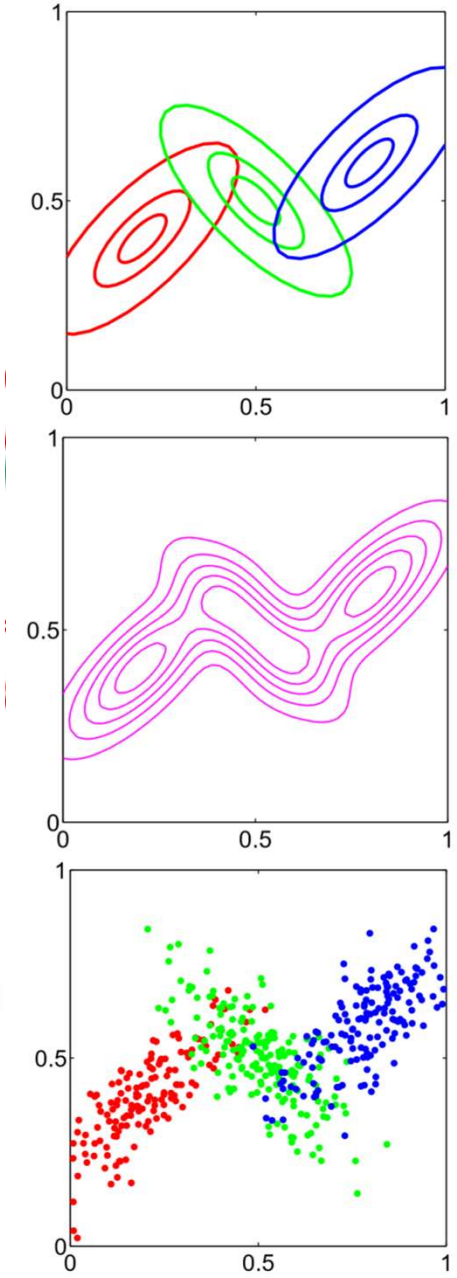
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

- Can interpret the mixing coefficients as prior probabilities

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



Probabilistic Kmeans

- GMM training (fitting)
  - Given a data set, find the corresponding GMM parameters, namely, mixing coefficients, means and covariances.
  - The maximum likelihood solution would involve fitting each component to the corresponding cluster only if component to data point assignment is known
  - These assignment labels are known as the latent/ hidden variables

Probabilistic Kmeans

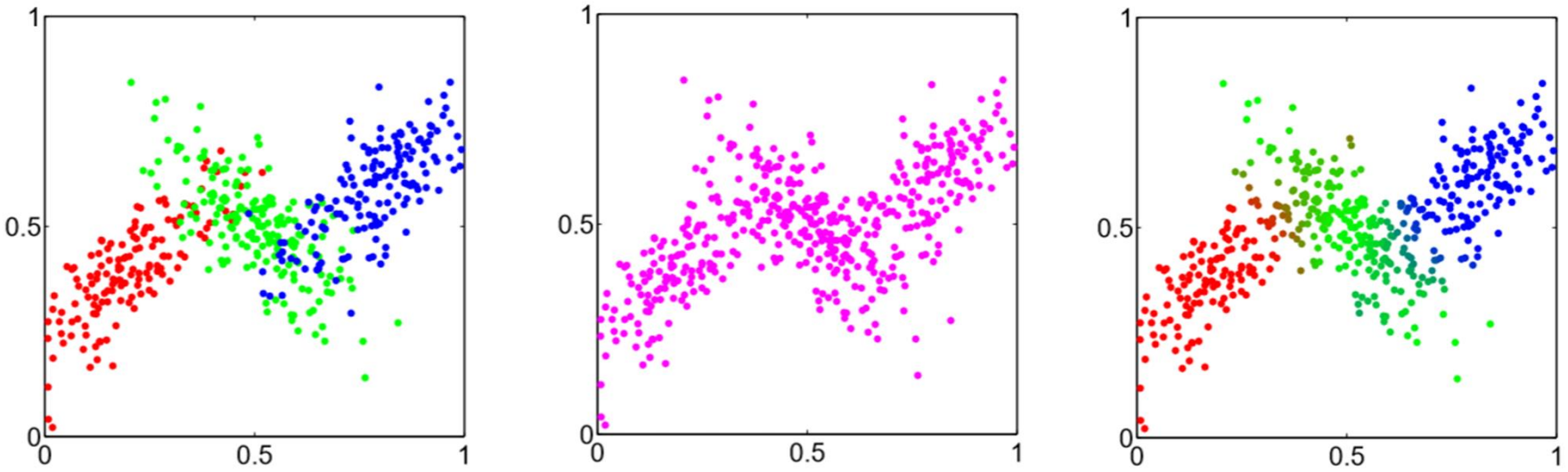
- Mixing coefficients can be seen as the prior probabilities for the components
- For a given data point we can evaluate the corresponding posterior probabilities, called soft assignments or responsibilities.
- These are given from Bayes' Theorem :

$$\gamma_k(\mathbf{x}) \equiv p(k|\mathbf{x}) = \frac{p(k)p(\mathbf{x}|k)}{p(\mathbf{x})}$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Probabilistic Kmeans

- Mixing coefficients can be seen as the prior probabilities for the components
- For a given data point we can evaluate the corresponding posterior probabilities, called soft assignments or responsibilities.
- These are given from Bayes' Theorem :



Probabilistic Kmeans

Maximum Likelihood for the GMM:

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

- Since there is no closed form solution of this likelihood function, an iterative expectation-maximization (EM) algorithm is used to maximize it

Probabilistic Kmeans

EM Algorithm –Informal Derivation

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\, \mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^{\mathsf{T}}}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

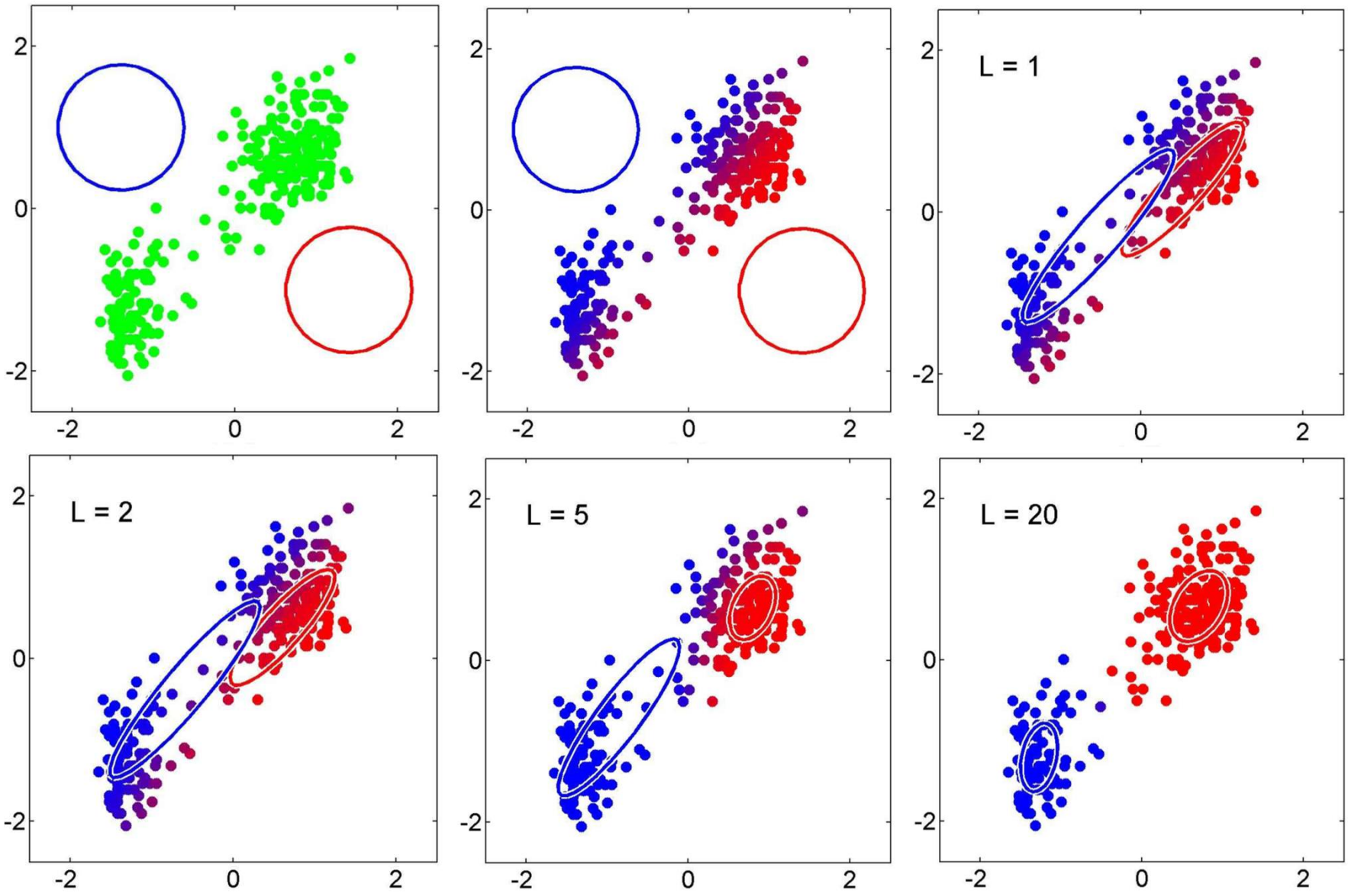
Probabilistic Kmeans

EM Algorithm –Informal Derivation

- make initial guesses for the parameters
- alternate between the following two stages:
  - E-step: evaluate assignments
  - M-step: update parameters using ML results

- This is a generalized version of the simple kmeans.
- In simple kmeans the Gaussians is considered to have spherical covariance matrices, identical for all components(clusters) and with hard assignments of points to clusters.

Probabilistic Kmeans



Fuzzy Kmeans

- In fuzzy clustering, every point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster.
- Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster.

- Minimize $J = \sum_{i=1}^{N} \sum_{j=1}^{K} {}^{m}a_{ij} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$

  such that ${}^{m}a_{ij} \epsilon [0,1]$ and $\sum_{j=1}^{K} {}^{m}a_{ij} = 1$

  where, ${}^{m}a_{ij} = \dfrac{1}{\sum_{k=1}^{K} \left(\dfrac{\left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|}{\left\| \mathbf{x}_i - \boldsymbol{\mu}_k \right\|} \right)^{2/(m-1)}}$ for $m \geq 1$

Kernel Kmeans

- Minimize $J = \sum_{i=1}^{N} \sum_{j=1}^{K} a_{ij} \left\| \varphi(\mathbf{x}_i) - \widetilde{\boldsymbol{\mu}_j} \right\|^2$

  such that $a_{ij} \epsilon \{0,1\}$ and $\sum_{j=1}^{K} a_{ij} = 1$

$$\widetilde{\boldsymbol{\mu}_j} = \frac{\sum_{i=1}^{N} a_{ij} \varphi(\mathbf{x}_i)}{\sum_{i=1}^{N} a_{ij}}$$

- We can rewrite the criterion function as:

  Minimize $J = \text{trace}(G) - \text{trace}(AGA^T)$

  Or, Maximize $\text{trace}(AGA^T)$

where, G is an $N \times N$ kernel matrix and A is the optimal normalized cluster membership matrix

(Solution will be discussed in the next class)

References

- Gaussian Mixture Models and the EM Algorithm. Jens Rittscher and Chuck Stewart
- Mixture Models and the EM Algorithm. Christopher M. Bishop