Pattern Recognition

Gaussian Mixture Model Classification

Outline

- Review
  - Clustering
  - K-Means
- Gaussian Mixture Model
- Expectation Maximization

K Means Clustering

- Clustering algorithms are used to find groups of "similar" data points among the input patterns.
- K-means clustering is an effective algorithm to extract a given number of clusters of patterns from a training set.
- The cluster locations can be used to classify data into distinct classes.

K-Means Algorithm

- Initialize the number of cluster centers selected (generally randomly selected from the training set)
- Classify the entire training data set. Use the distance of each data sample from the cluster centers
- For each cluster, find re-compute the center of the cluster using

$$M_k = \frac{1}{N_k} \cdot \sum_{j=1}^{N_k} X_{jk}$$

Parameter K

- K (the number of clusters) can be provided by the user, or estimated from the training data set.
- If K is provided by the user:
  - Start with the user specified number of clusters
  - If the standard deviation in a cluster is too large, split it into two clusters

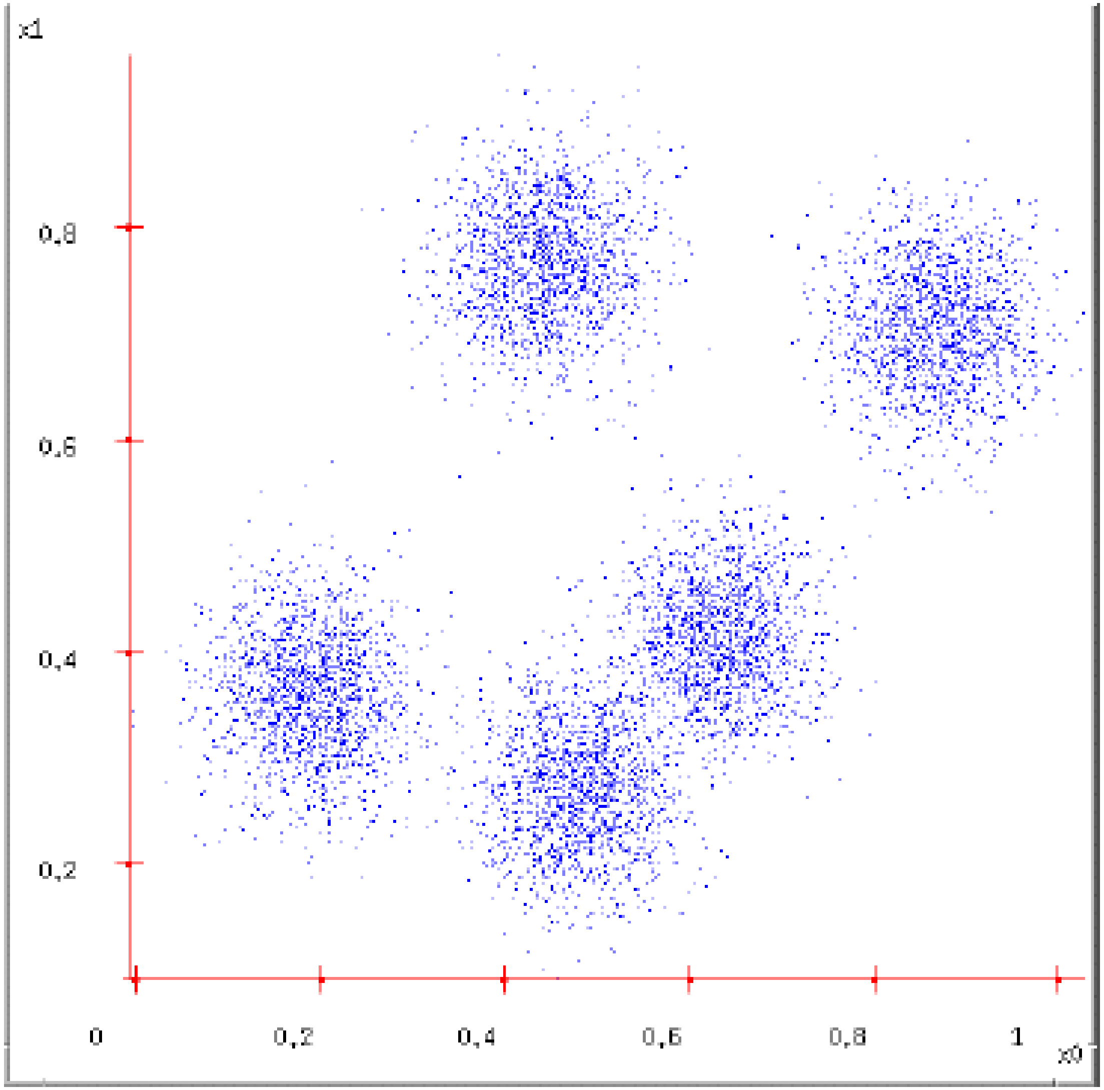
$$\sigma_i = \sum_{j=1}^{N_k} (X_{ij} - M_{ij})^2$$

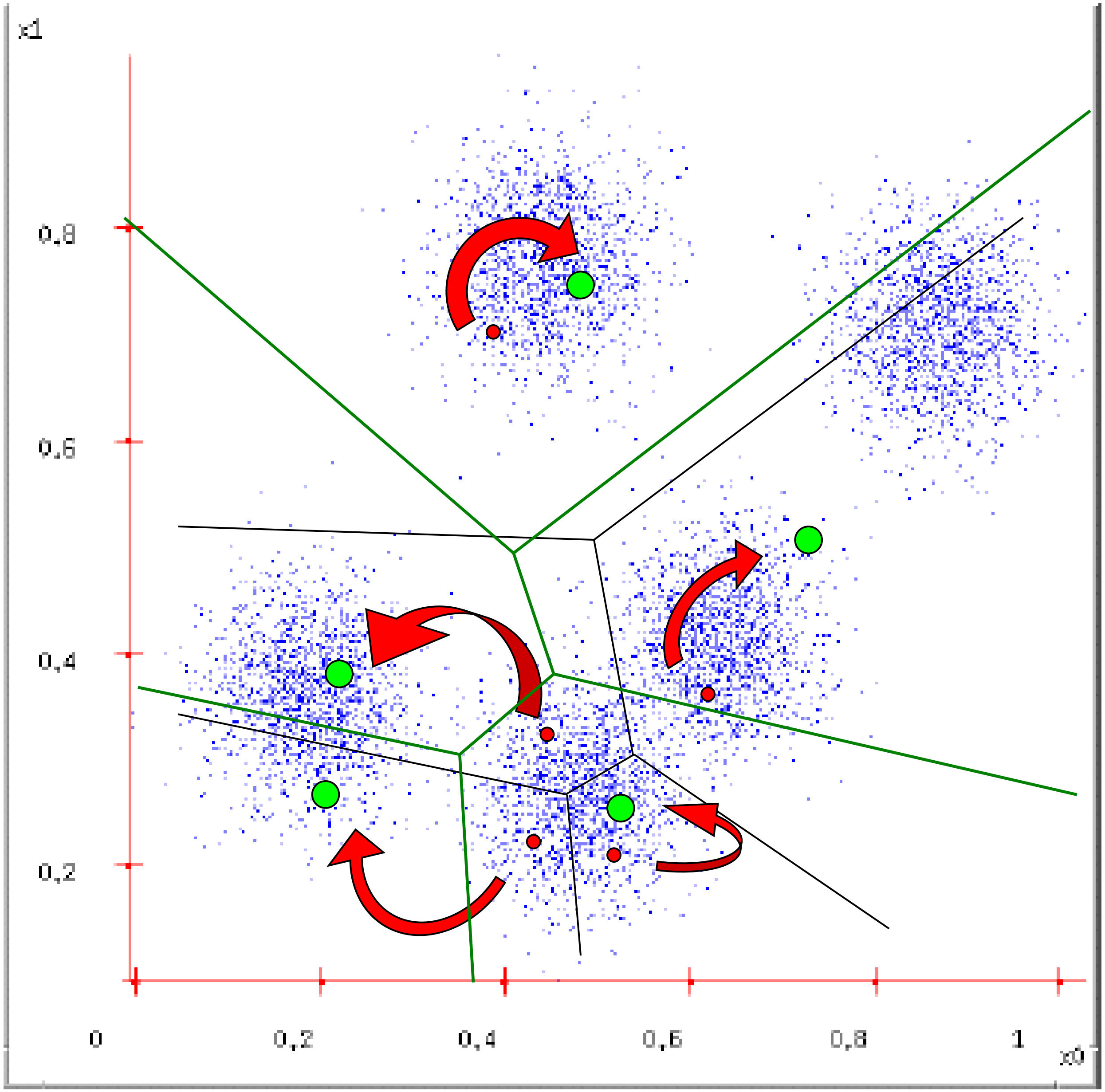
  - If the centers of two clusters are too close, merge them

Parameter K

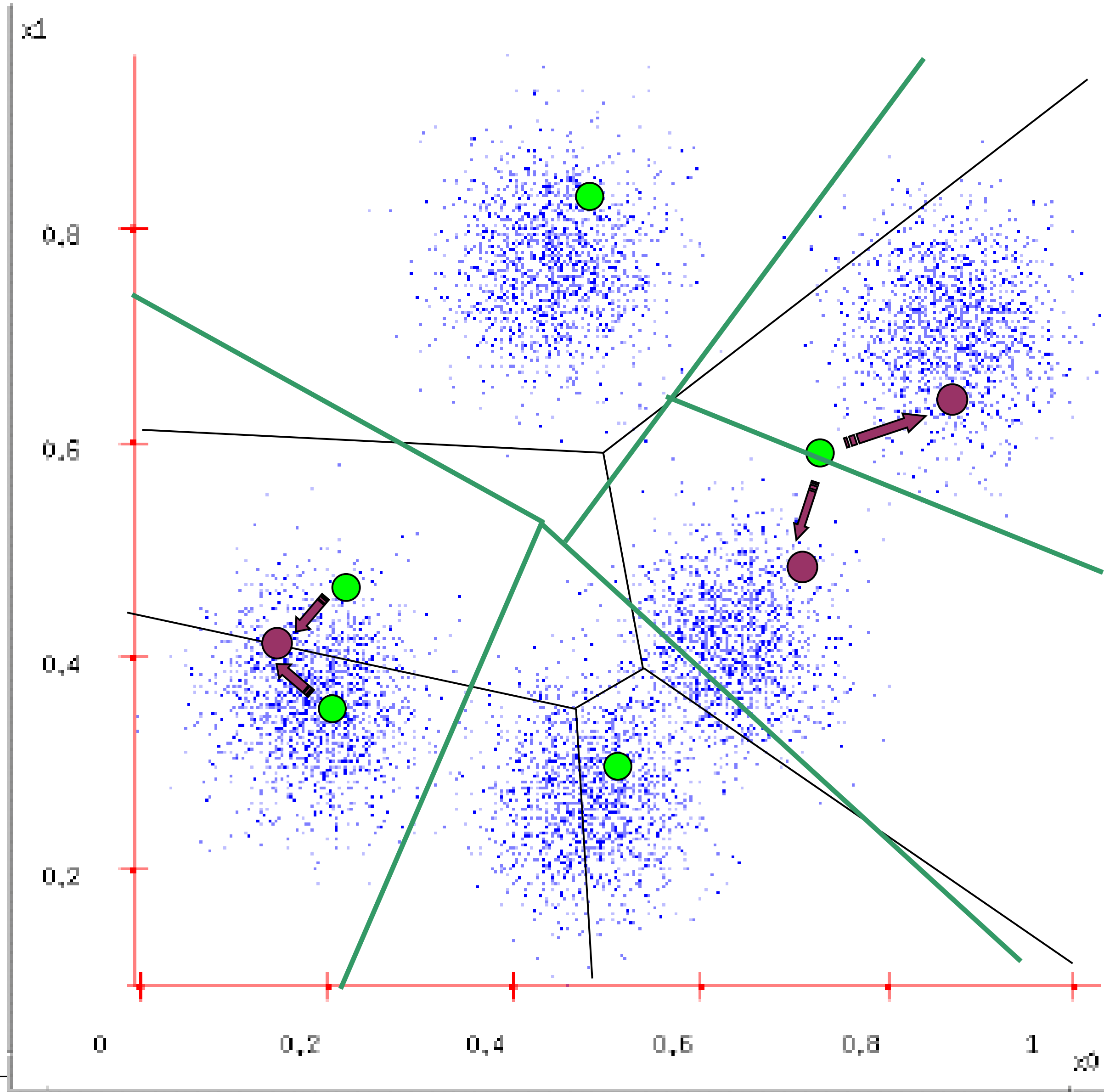
- If K is not provided by the user:
  - Start with the minimum number of clusters (two for instance)
  - Split the cluster with the largest standard deviation value which is larger than a threshold too.
  - Merge two clusters with the smallest distance between their centers. The distance should be less than a threshold too.
  - Alternatively, the algorithm can be started with the maximum number of clusters (too slow)

Example




x1
0,8
0,6
0,4
0,2
0
0,2
0,4
0,6
0,8
1
x0

Splitting & Merging



Gaussian Mixture

- The Gaussian mixture architecture estimates probability density functions (PDF) for each class, and then performs classification based on Bayes' rule:

$$P(C_i \mid X) = P(X \mid C_i) \cdot \frac{P(C_i)}{P(X)}$$

Where $P(X \mid C_i)$ is the PDF of class j, evaluated at X, $P(C_j)$ is the prior probability for class j, and $P(X)$ is the overall PDF, evaluated at X.

Gaussian Mixture

- Unlike the unimodal Gaussian architecture, which assumes $P(X \mid C_j)$ to be in the form of a Gaussian, the Gaussian mixture model estimates $P(X \mid C_j)$ as a weighted average of multiple Gaussians.

$$P(X \mid C_j) = \sum_{k=1}^{Nc} w_k G_k$$

Where w_k is the weight of the k-th Gaussian G_k and the weights sum to one. One such PDF model is produced for each class.

Gaussian Mixture

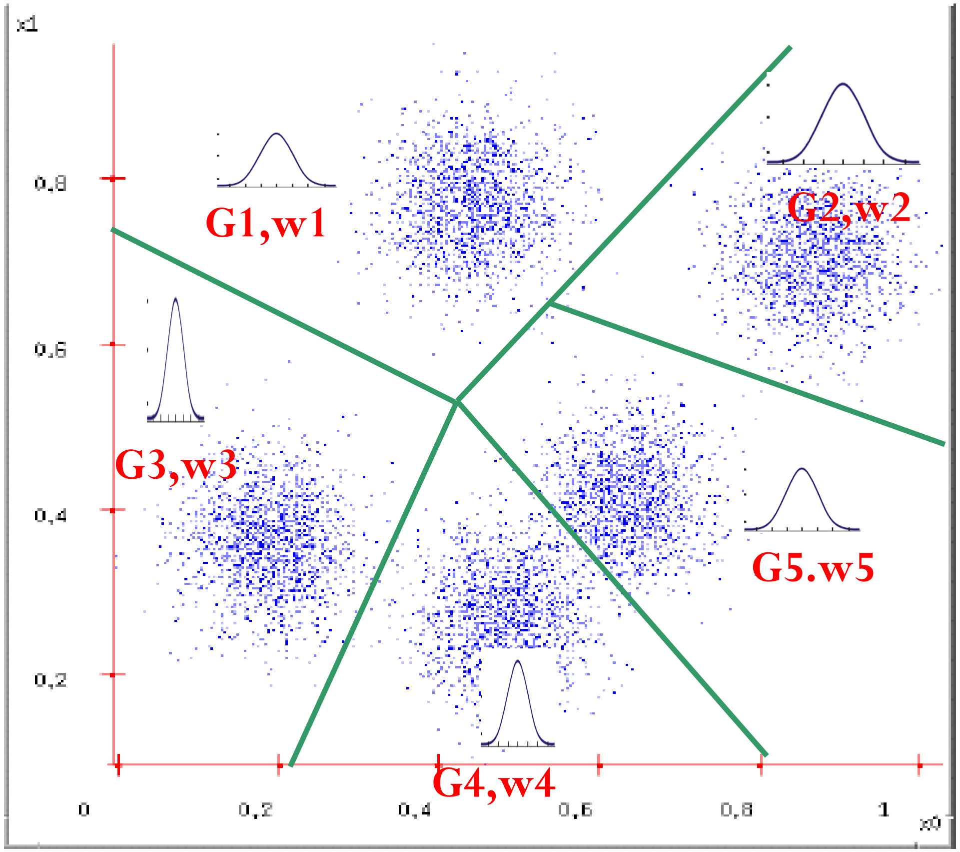
- Each Gaussian component is defined as:

$$G_k = \frac{1}{(2\pi)^{n/2} \, | \Sigma_k |^{1/2}} \cdot e^{[-1/2 (X - \mu_k)^T \Sigma_k^{-1} (X - \mu_k)]}$$

Gaussian Mixture

- Free parameters of the Gaussian mixture model consist of the means and covariance matrices of the Gaussian components
- The weights indicating the contribution of each Gaussian to the approximation of $P(X \mid C_j)$.

Composition of Gaussian Mixture



Gaussian Mixture

- These parameters are tuned using an iterative procedure called the Expectation-Maximization (EM) algorithm, that aims at maximizing the likelihood of the training set generated by the estimated PDF.
- The likelihood function L for each class j can be defined as:

$$L_j = \prod_{i=0}^{N_{train}} P(X_i \mid C_j) \longrightarrow \ln(L_j) = \sum_{i=0}^{N_{train}} \ln(P(X_i \mid C_j))$$

Gaussian Mixture Training Flow Chart (1)

Initialize the initial **Gaussian means μ_i**, i=1,…G using the K means clustering algorithm

Initialize the **covariance matrices, Σ_i**, to the distance to the nearest cluster.

Initialize the **weights $\pi_i = 1 / G$** so that all Gaussian are equally likely.

Gaussian Mixture Training Flow Chart (2)

- Present each pattern X of the training set and model each of the classes K as a weighted sum of Gaussians:

$$p(X \mid \theta_s) = \sum_{i=1}^{G} \pi_i \, p(X \mid G_i)$$

- Where **G** is the number of Gaussians, the **π_i'**s are the weights, and

$$p(X \mid G_i) = \frac{1}{(2\pi)^{d/2} \, |\Sigma_i|^{1/2}} \cdot e^{[-1/2(X-\mu_i)^T \Sigma i^{-1} (X-\mu_i)]}$$

- Where **Σ_i** is the covariance matrix.

Gaussian Mixture Training Flow Chart (3)

Compute: The probability of cluster-i given X

$$\tau_{ip} \equiv P(G_i \mid X) = \frac{\pi_i p(X \mid G_i, C_k)}{p(X)} = \frac{\pi_i p(X \mid G_i, C_k)}{\sum_{j=1}^{G} \pi_j p(X \mid \theta_j, C_k)}$$

Gaussian Mixture Training Flow Chart (4)

- Iteratively update the **weights**, **means** and **covariances**:

$$\pi_i(t+1) = \frac{1}{N_c} \sum_{p=1}^{N_c} \tau_{ip}(t)$$

$$\mu i(t+1) = \frac{1}{N_c \pi_i(t)} \sum_{p=1}^{N_c} \tau_{ip}(t) X_p$$

$$\Sigma_i(t+1) = \frac{1}{N_c \pi_i(t)} \sum_{p=1}^{N_c} \tau_{ip}(t)((X_p - \mu_i(t))(X_p - \mu_i(t))^T)$$

Gaussian Mixture Training Flow Chart (5)

Re-compute τ_{ip} using the new weights, means and covariances. Stop training if

$$\Delta\tau_{i\,p} \equiv \tau_{i\,p}(t+1) - \tau_{i\,p}(t) \leq threshold$$

Or the number of epochs reach the specified value. Otherwise, continue the iterative updates.

Gaussian Mixture Test Flow Chart

Present each input pattern X and compute the confidence for each class j:

$$P(C_j)P(X \mid \theta_x, C_j)$$

Where $P(C_j) = \frac{N_{ci}}{N}$ is the prior probability of class $\mathbf{C_j}$ estimated by counting the number of training patterns.

Classify pattern X as the class with the highest confidence.

Questions?