Lecture 23
EM Algorithm for Clustering

ECEN 5283 Computer Vision

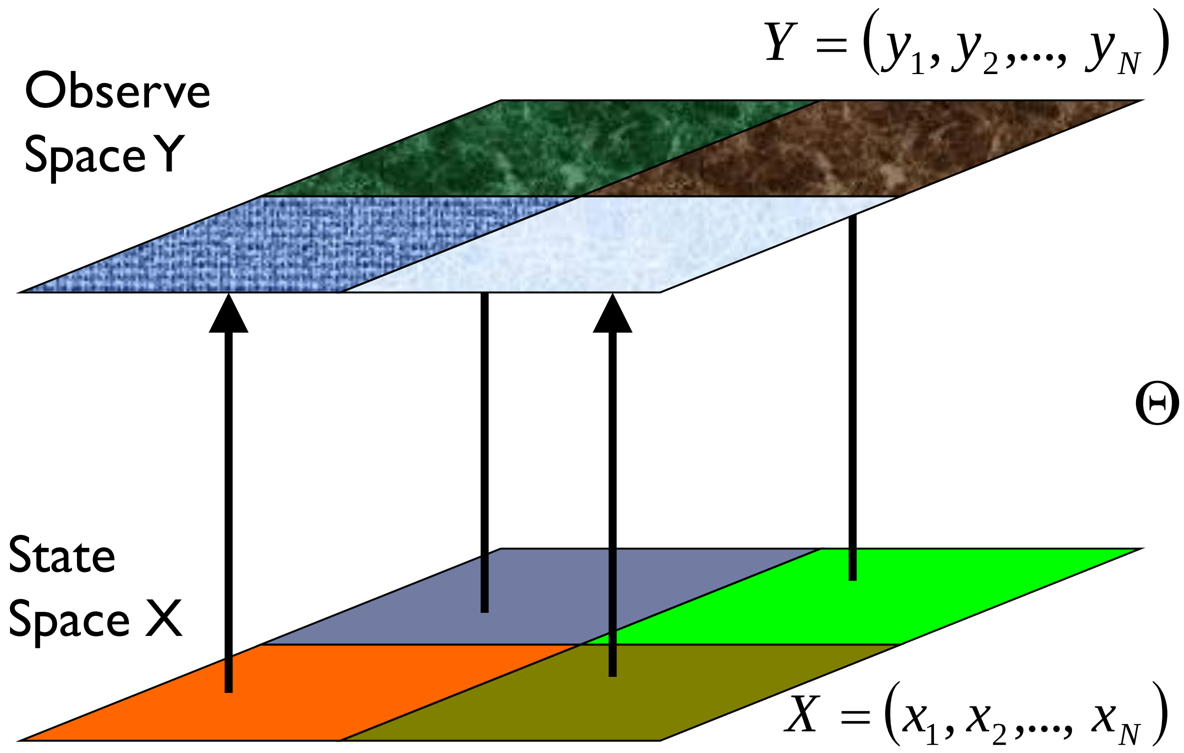
Dr. Guoliang Fan
School of Electrical and Computer Engineering
Oklahoma State University

Goals

- To revisit the missing data problem that involves two data likelihood functions

- To revisit a soft-clustering EM algorithm by looking into the likelihood function and the initialization

Missing Data Problem Revisited



Parameter Space

$$\Theta = \{\alpha_i, \boldsymbol{\mu}_i, \Sigma_i \mid i = 1,2,3,4\}$$

$$\Theta^* = \arg\max_{\Theta} \log p(Y \mid \Theta) \quad \text{(log-likelihood of incomplete data)}$$

$$\Theta^* = \arg\max_{\Theta} \log p(X, Y \mid \Theta) \quad \text{(log-likelihood of complete data)}$$

Two Log-likelihood Functions

- Incomplete data log-likelihood (without the labels X)

$$\log p(Y \mid \Theta) = \sum_{j=1}^{N} \log\left(\sum_{i=1}^{k} p(y_j \mid x_j = i, \Theta)\alpha_i \right)$$

Can be computed but not easy to optimize

- Complete data log-likelihood (with the labels X)

$$\log p(X, Y \mid \Theta) = \sum_{j=1}^{N} \log\left(\sum_{i=1}^{k} x_{ji}\, p(y_j \mid x_j = i, \Theta)\alpha_i \right)$$

$$= \sum_{j=1}^{N} \sum_{i=1}^{k} x_{ji} \log p(y_j \mid x_j = i, \Theta)\alpha_i$$

$$x_{ji} = \begin{cases} 1 & \text{if state } i \text{ produces observation } j \\ 0 & \text{Otherwise} \end{cases}$$

Cannot be computed but can be optimized via a lower bound

EM Algorithm: E-step

- **Initialization**: set s=0 and

$$\Theta^0 = (\alpha_1^{(0)}, \alpha_2^{(0)}, \ldots, \alpha_k^{(0)}, \theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_k^{(0)}). \qquad \theta_i = \{\mu_i, \Sigma_i\}$$

- **Expectation** (E-step):
  - Compute an expected value of for the complete data using the incomplete data and the current parameters.

$$\mathbf{I}(l,m) = \frac{\alpha_m^{(s)} p(y_l \mid \theta_m^{(s)})}{\sum_{i=1}^{k} \alpha_i^{(s)} p(y_l \mid \theta_i^{(s)})} = p(x_l = m \mid y_l, \Theta^{(s)})$$

Posterior probability

$$p(x \mid y) = \frac{p(x,y)}{p(y)} = \frac{p(y \mid x) p(x)}{\sum_x p(y \mid x) p(x)}$$

$$p(y_l \mid \theta_i^{(s)}) = \frac{\exp\left\{-\frac{1}{2}(y_l - \mu_i)^T \Sigma_i^{-1} (y_l - \mu_i)\right\}}{(2\pi)^{d/2} \det(\Sigma_i)^{1/2}}$$

Likelihood function

EM Algorithm: M-step

- Maximization (M-step):
  - Find the parameters by using the estimate of missing data.

$$\alpha_i^{(s+1)} = \frac{1}{N}\sum_{l=1}^{N} p(x_l = i \mid y_l, \Theta^{(s)})$$

$$\mu_i^{(s+1)} = \frac{\sum_{l=1}^{N} y_l\, p(x_l = i \mid y_l, \Theta^{(s)})}{\sum_{l=1}^{N} p(x_l = i \mid y_l, \Theta^{(s)})}$$

$$\Sigma_i^{(s+1)} = \frac{\sum_{l=1}^{N} p(x_l = i \mid y_l, \Theta^{(s)})\left\{(y_l - \mu_i^{(s)})(y_l - \mu_i^{(s)})^T\right\}}{\sum_{l=1}^{N} p(x_l = i \mid y_l, \Theta^{(s)})}$$

Gaussian Likelihood Function

- In the case of single variable *x*, the Gaussian distribution can be written as

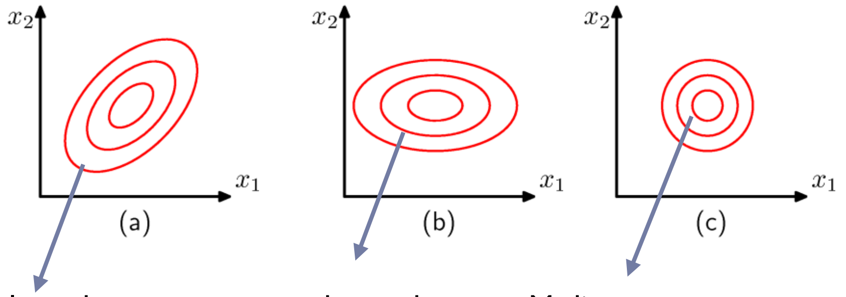
$$N(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2 \right)$$

- In a D-dimensional case, the multivariate Gaussian distribution is defined

$$N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2} \underline{(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})} \right)$$

  - Mahalanobis distance

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})$$



All points along the same contour shares the same M-distance.

Covariance Matrix

- Given the covariance, eigen-value decomposition is defined as

$$\Sigma = \mathbf{U}\boldsymbol{\Lambda}\,\mathbf{U}^{\mathrm{T}} \quad \Longrightarrow \quad \Sigma\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

$$\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2) \qquad \mathbf{U}^{\mathrm{T}} = \mathbf{U}^{-1} \qquad \text{where } \mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

  - Consider the 2D case, we can represent the covariance matrix by its eigenvectors as

$$\Sigma = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^{\mathrm{T}} \\ \mathbf{u}_2^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \lambda_1\mathbf{u}_1 & \lambda_2\mathbf{u}_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^{\mathrm{T}} \\ \mathbf{u}_2^{\mathrm{T}} \end{pmatrix} = \lambda_1 \mathbf{u}_1 \mathbf{u}_1^{\mathrm{T}} + \lambda_2 \mathbf{u}_2 \mathbf{u}_2^{\mathrm{T}}$$

Outer product

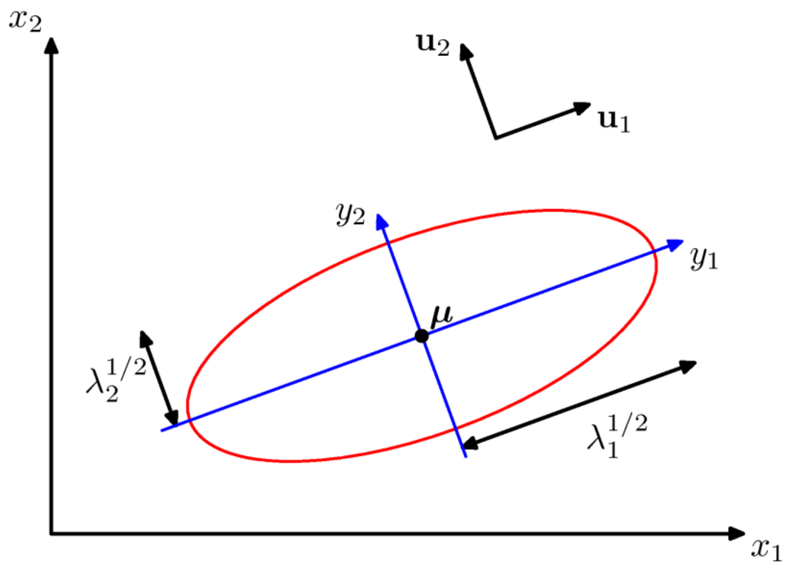
- Then we represent the covariance matrix by its eigenvectors as

$$\Sigma = \mathbf{U}\boldsymbol{\Lambda}\,\mathbf{U}^{\mathrm{T}} = \sum_{i=1}^{D} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \rightarrow \Sigma^{-1} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\,\mathbf{U}^{\mathrm{T}} = \sum_{i=1}^{D} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T$$

Mahalanobis Distance

- Given a vector **x**, the M-distance is obtained as

$$\begin{aligned}\Delta^2 &= (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\\ &= (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\left(\sum_{i=1}^{D}\frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T\right)(\mathbf{x}-\boldsymbol{\mu})\\ &= \sum_{i=1}^{D}\frac{1}{\lambda_i}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{u}_i\mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu})\\ &= \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i} \quad \text{where } y_i = \mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu}).\end{aligned}$$



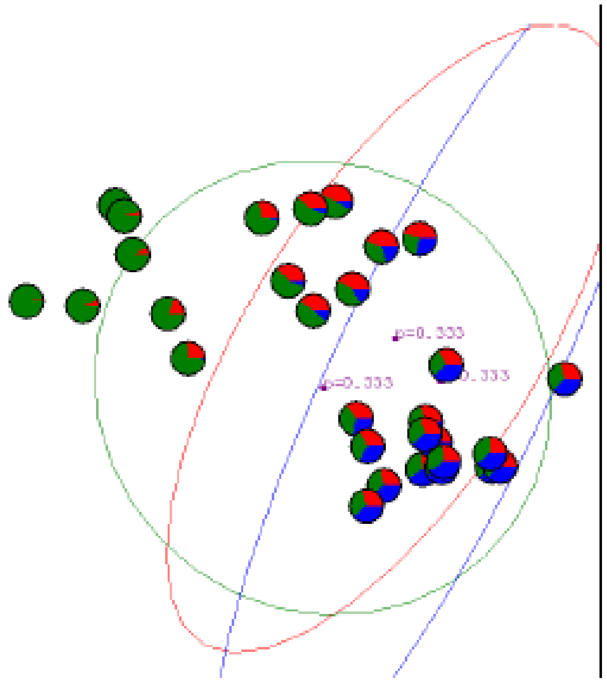
- We interpret $\{y_i \mid i = 1, 2, \ldots, D\}$ as a new coordinate system defined by the orthonormal vectors $\mathbf{u}_i^T$.
  - Then in the 2D case, a new vector system is obtained

What is the good thing about this new coordinate system?

$$\mathbf{y} = \mathbf{U}^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}) \qquad \text{where } \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \text{ and } \mathbf{U}^{\mathrm{T}} = \begin{pmatrix} \mathbf{u}_1^{\mathrm{T}} \\ \mathbf{u}_2^{\mathrm{T}} \end{pmatrix}$$

Demo: Initialization

Example: EM for GMM

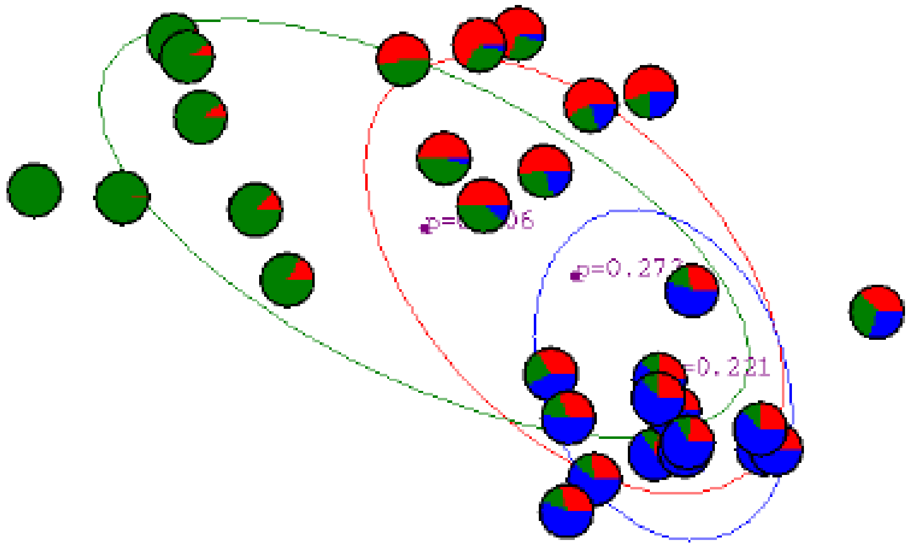


Initial model parameters.

https://www2.cs.duke.edu/courses/fall07/cps271/EM.pdf

Demo: 1st iteration

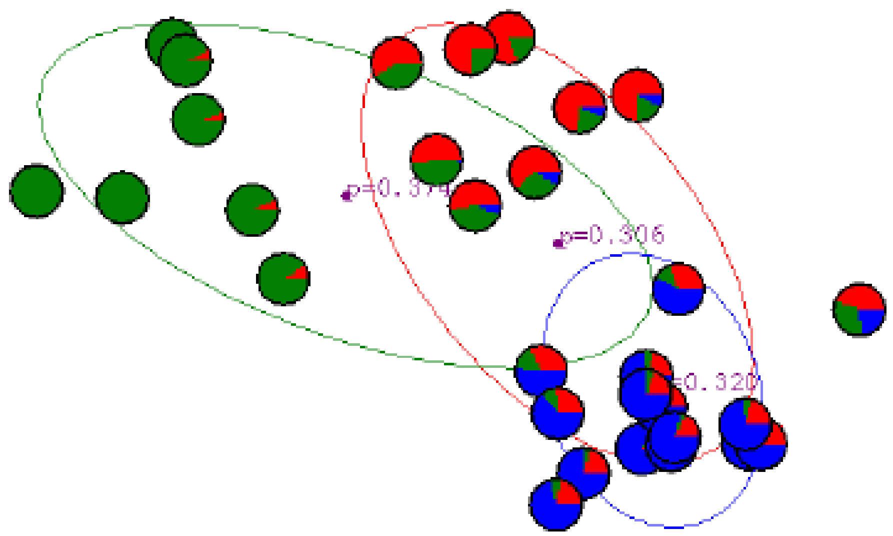
Example: EM for GMM



After first iteration

Demo: 2nd iteration

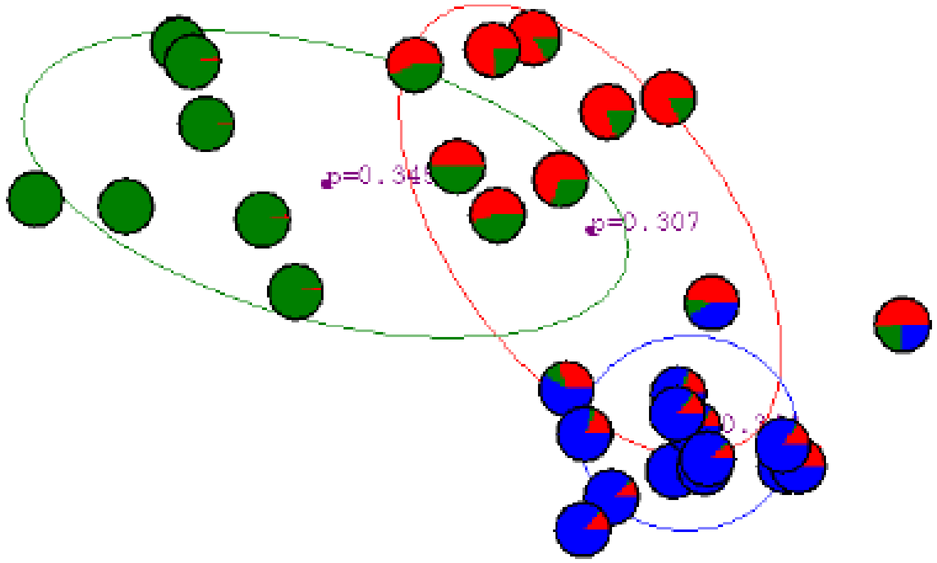
Example: EM for GMM



After second iteration

Demo: 3rd iteration

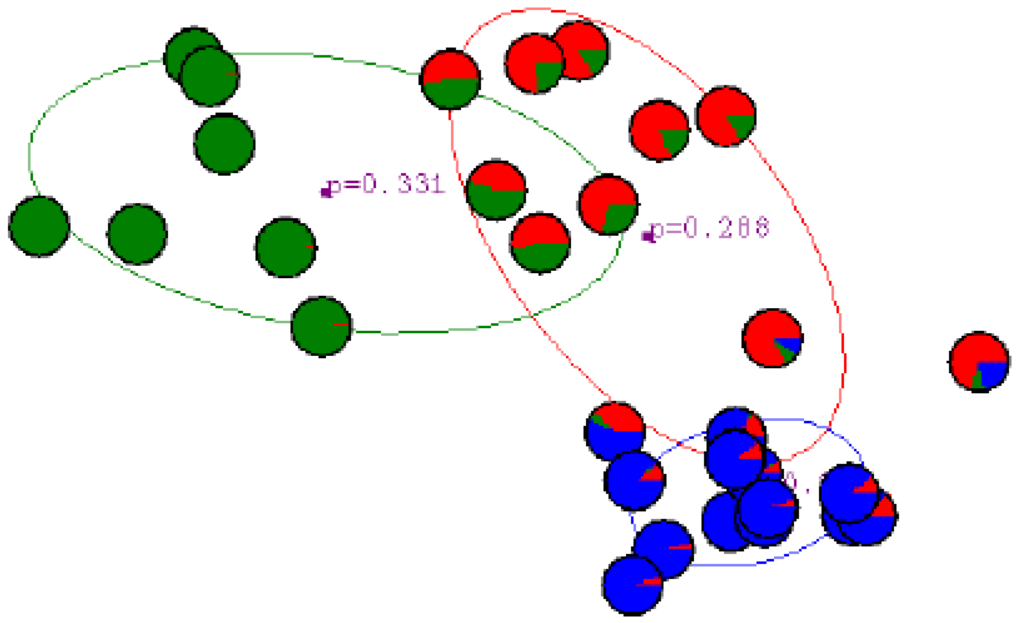
Example: EM for GMM



After third iteration

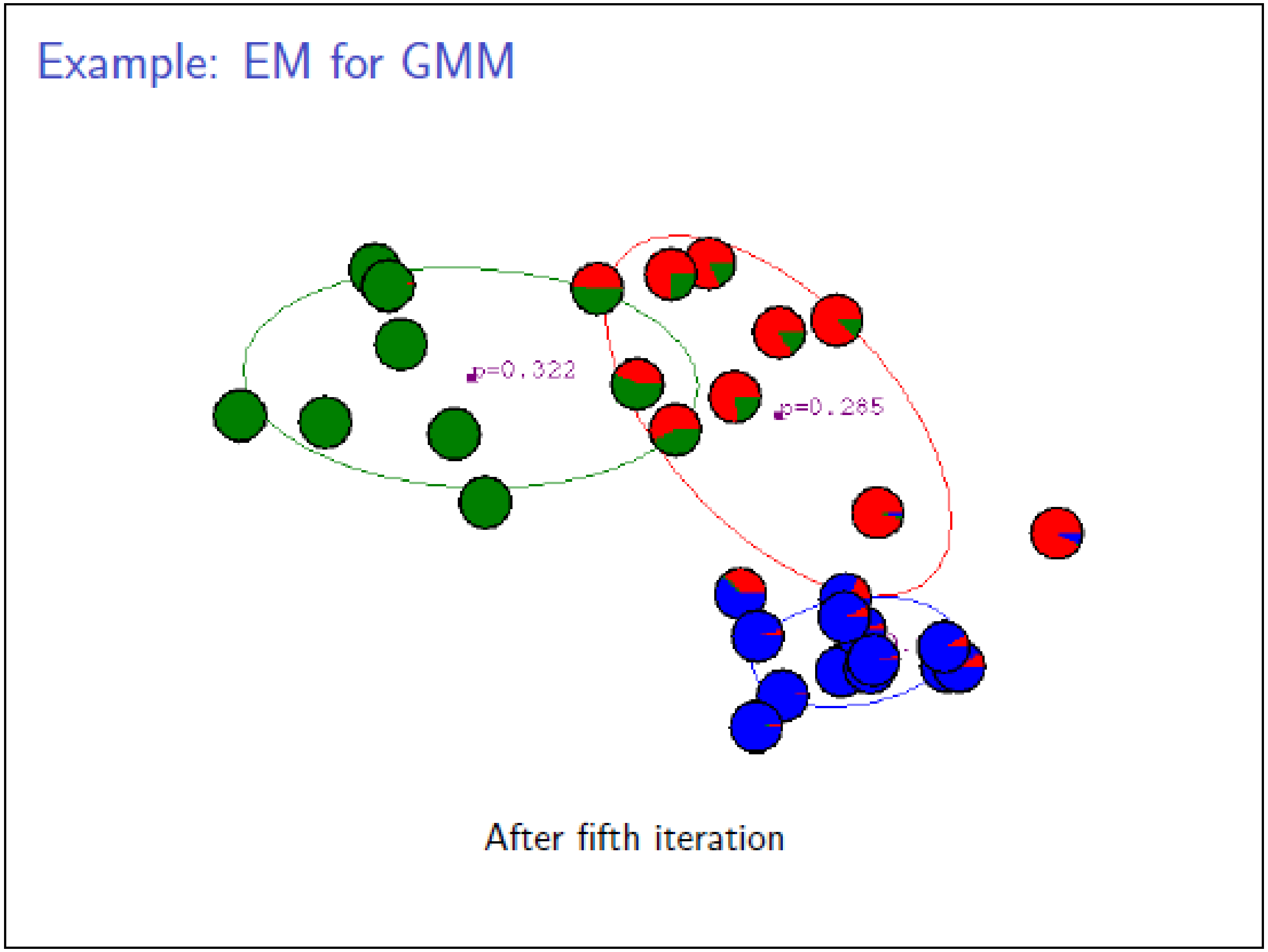
Demo: 4th iteration

Example: EM for GMM



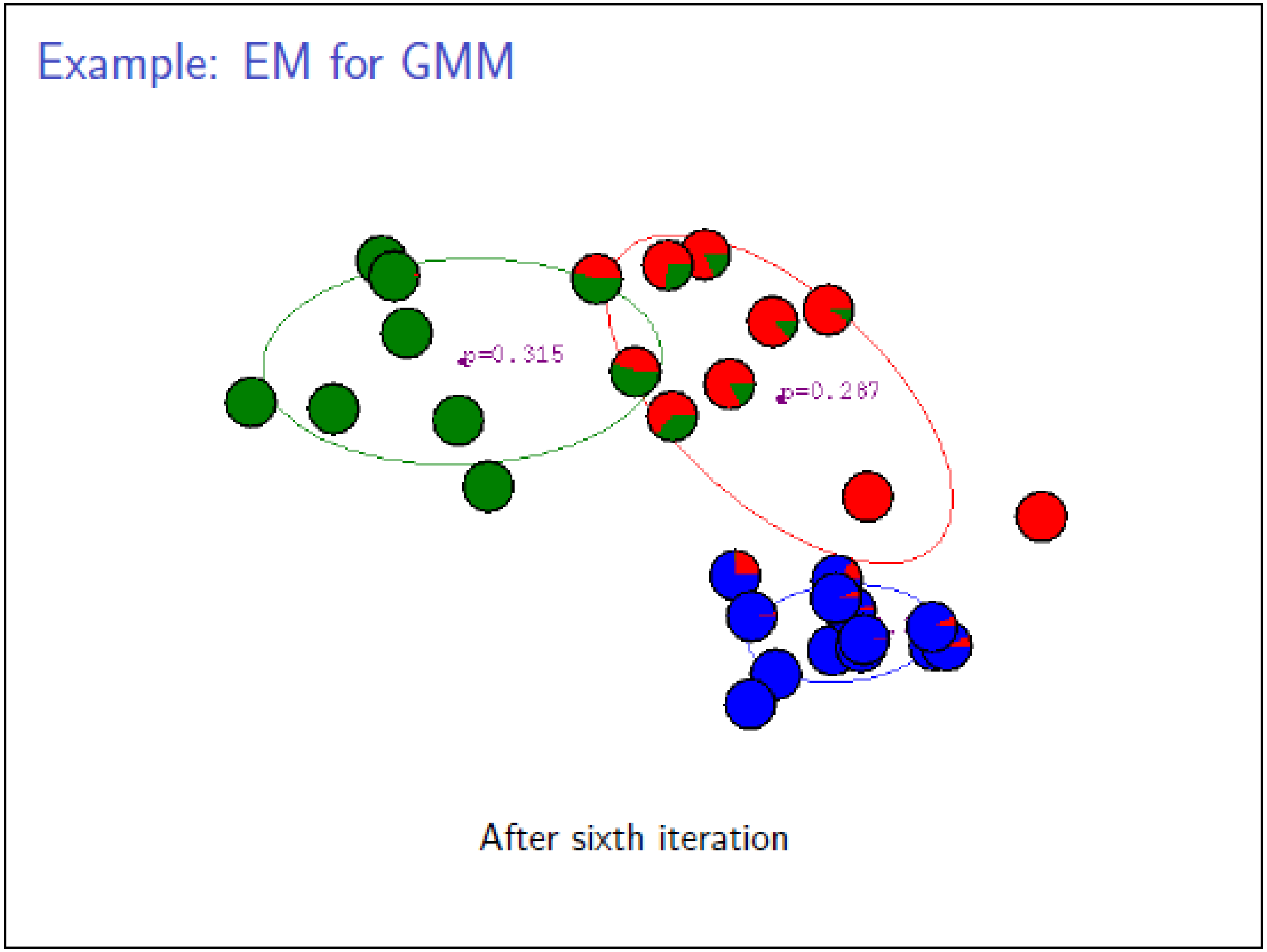
After fourth iteration

Demo: 5th iteration


Example: EM for GMM

After fifth iteration

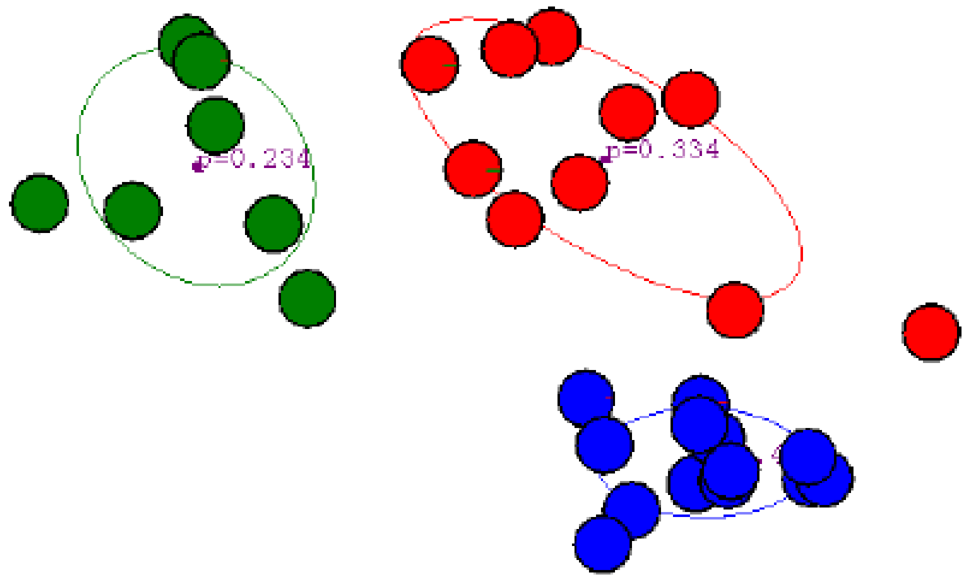
Demo: 6th iteration


Example: EM for GMM

After sixth iteration

Demo: Convergence

Example: EM for GMM



After convergence

EM Algorithm: Stop Condition

- Iteration still the stop criteria is satisfied, e.g., no much change of the incomplete data log-likelihood

$$p(\mathbf{Y}|\Theta^{(s+1)}) - p(\mathbf{Y}|\Theta^{(s)}) < \Delta \quad \rightarrow \log p(\mathbf{Y}|\Theta^{(s+1)}) - \log p(\mathbf{Y}|\Theta^{(s)}) < \Delta'$$

$$\log p(Y|\Theta) = \sum_{j=1}^{N} \log\left(\sum_{i=1}^{k} p(y_j | x_j = i, \Theta)\alpha_i \right)$$

(or an easy way to fix the iteration number)

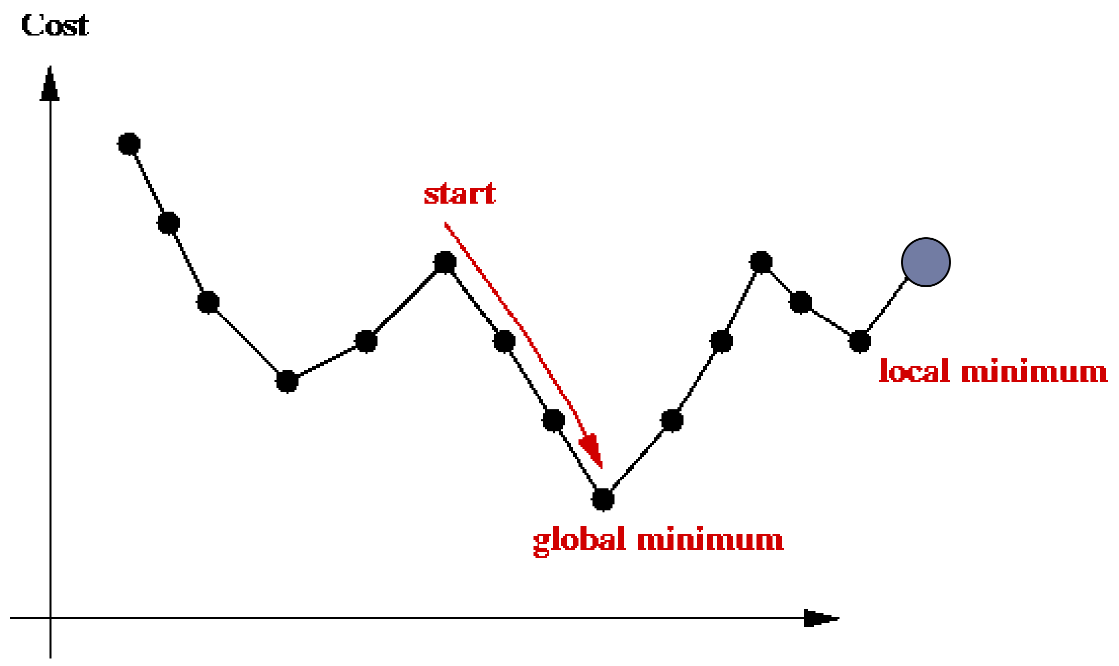
- Decide the class label for data point

$$\mathbf{I}(l,m) = p(x_l = m | y_l, \Theta^{(s)})$$

$$x_l = \arg_{m\in\{1,\ldots,g\}} \max \mathbf{I}(l,m)$$

EM Algorithm: Local Optimality

- Both K-mean and EM can only converge to the local optimum of the objective function.
- In other words, both methods are sensitive to the initialization, especially EM.



EM Algorithm: K-means Initialization

- Random initialize k centers for the k-means algorithm.

$$C^0 = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$$

- Run the k-means algorithm until converge or with certain iteration number.

$$\Phi(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{k} \left\{ \sum_{x_j = i} \left| y_j - \mathbf{c}_i \right|^2 \right\}$$

- According to the class label of all samples $\{x_1, x_2, \ldots, x_N\}$, initialize the multivariate Gaussian models for the EM.

$$\alpha_i = \frac{\#(x_j = i \mid j = 1, \ldots, N)}{N}$$

$$\Sigma_i = \frac{\sum_{x_j = i} (y_j - \mu_i)(y_j - \mu_i)^T}{\#(x_j = i \mid j = 1, \ldots, N)}$$

$$\mu_i = \frac{\sum_{x_j = i} y_j}{\#(x_j = i \mid j = 1, \ldots, N)}$$

K-mean vs. EM

- Both K-mean and EM need to be initialized and involve two major steps during iteration.

	K-mean	EM
Initialization	Initialize k means (cluster centers) $C^0 = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$	Initialize k Gaussian models that have equal weights. $\Theta^0 = \{\alpha_i, \mu_i, \Sigma_i \mid i = 1, \ldots, k\}$
Step 1.	Assume the cluster centers are known, and classify each data sample to the closest cluster center.	Given the model parameters, estimate the missing data in terms of the posterior probability of each data sample.
Step 2.	Assume the allocation (the class label of each sample) is known, and choose a new set of cluster centers. Each center is the mean of all points allocated to that cluster.	From the estimated missing data, to obtain the maximum likelihood estimate of the model parameters.