CS 412 Introduction to Machine Learning

Gaussian Mixture Model (GMM)

Instructor: Wei Tang

Department of Computer Science
University of Illinois at Chicago
Chicago IL 60607

https://tangw.people.uic.edu
tangw@uic.edu

Slides credit: Sargur Srihari

Gaussian Mixture Model (GMM)

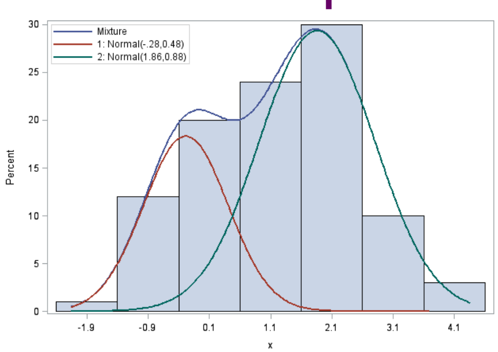
- Gaussian mixture distribution is written as
  - a linear superposition of K Gaussian components:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$$

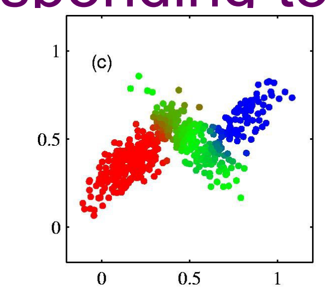
  - Represent k by a K-dimensional binary variable $\boldsymbol{z}$
    - Using 1-of-K representation (one-hot vector)
    - Let $\boldsymbol{z} = z_1,..,z_K$ whose elements are

$$z_k \in \{0,1\} \text{ and } \sum_k z_k = 1$$

    - K possible states of $\boldsymbol{z}$ corresponding to K components



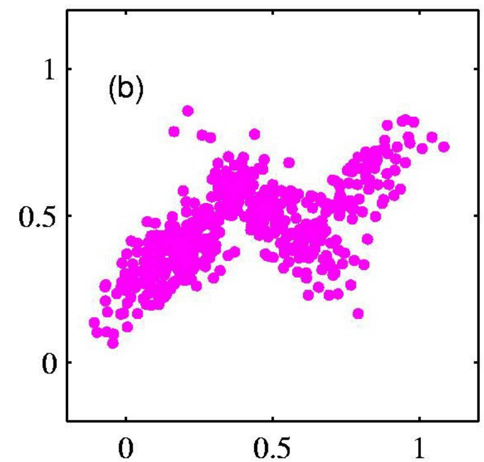
k	1	2
$\boldsymbol{z}$	10	01
π_k	0.4	0.6
μ_k	−28	1.86
σ_k	0.48	0.88



k	1	2	3
$\boldsymbol{z}$	100	010	001

Joint Distribution

- Define joint distribution of latent variable and observed variable
  - $p(x,z) = p(x|z) \cdot p(z)$
  - x is observed variable: a feature vector
  - z is the hidden variable: cluster assignment
  - Prior prob. distribution $p(z)$
  - Likelihood prob. distribution $p(x|z)$



Specifying the prior prob. $p(z)$

- Associate a probability with each component z_k
  - Denote $p(z_k = 1) = \pi_k$ where parameters $\{\pi_k\}$ satisfy

$$0 \leq \pi_k \leq 1 \text{ and } \sum_k \pi_k = 1$$

- Because z uses 1-of-K it follows that

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

With one component $p(z_1) = \pi_1^{z_1}$

With two components $p(z_1, z_2) = \pi_1^{z_1} \pi_2^{z_2}$

Specifying the Likelihood prob. $p(x|z)$

- For a particular component (value of z)

$$p(x \mid z_k = 1) = N(x \mid \mu_k, \Sigma_k)$$

- Thus $p(x|z)$ can be written in the form

$$p(x \mid z) = \prod_{k=1}^{K} N\left(x \mid \mu_k, \Sigma_k\right)^{z_k}$$

  – All product terms except for one equal one

Marginal distribution $p(\boldsymbol{x})$

- The joint distribution $p(\boldsymbol{x},\boldsymbol{z})$ is given by $p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$
- Thus marginal distribution of $\boldsymbol{x}$ is obtained by summing over all possible states of $\boldsymbol{z}$ to give

$$p(\boldsymbol{x}) = \sum_{z} p(z)p(\boldsymbol{x} \mid z) = \sum_{z} \prod_{k=1}^{K} \pi_k^{z_k} N\left(\boldsymbol{x} \mid \mu_k, \Sigma_k\right)^{z_k} = \sum_{k=1}^{K} \pi_k N\left(\boldsymbol{x} \mid \mu_k, \Sigma_k\right)$$

  - Since $z_k \in \{0,1\}$
- This is the standard form of a Gaussian mixture

Gaussian Mixture Model (GMM)

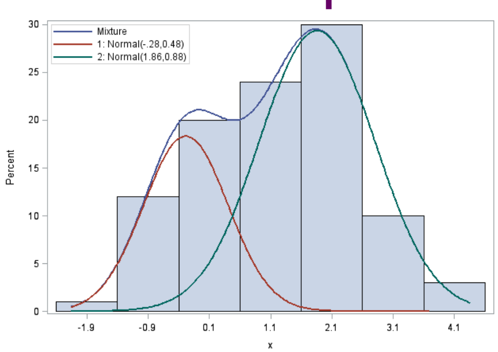
- Gaussian mixture distribution is written as
  - a linear superposition of K Gaussian components:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$$

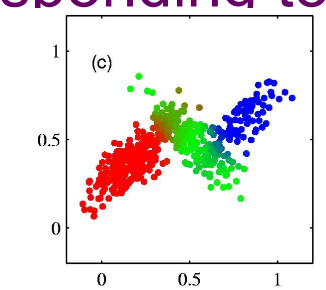
  - Represent K by a K-dimensional binary variable $\boldsymbol{z}$
    - Using 1-of-K representation (one-hot vector)
    - Let $\boldsymbol{z} = z_1,..,z_K$ whose elements are

$$z_k \in \{0,1\} \text{ and } \sum_k z_k = 1$$

    - K possible states of $\boldsymbol{z}$ corresponding to K components



k	1	2
$\boldsymbol{z}$	10	01
π_k	0.4	0.6
μ_k	–28	1.86
σ_k	0.48	0.88

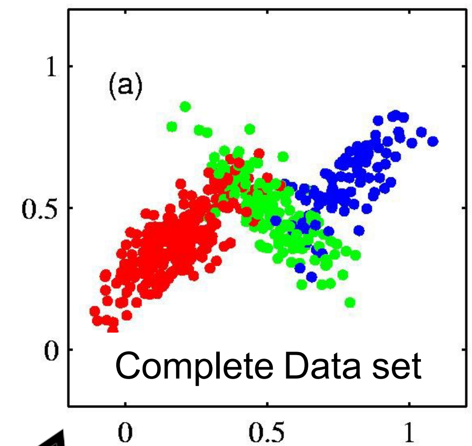


k	1	2	3
$\boldsymbol{z}$	100	010	001

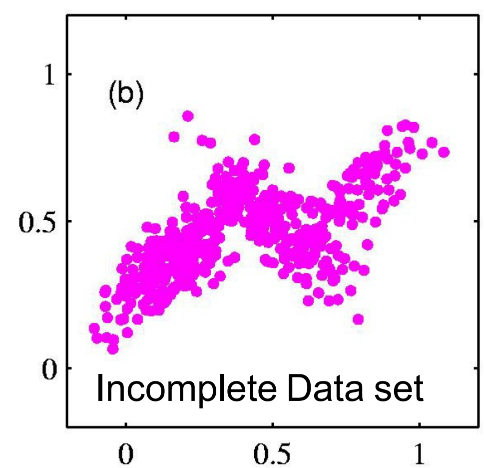
Synthesizing data from mixture

500 points from three Gaussians

- Generate sample of z, called $\hat{z}$
- Then generate a value for x from conditional p(x| zˆ)



- Samples from $p(x,z)$ are plotted according to value of x and colored with value of z
    - Samples from marginal $p(x)$ obtained by ignoring values of z



Learning: expectation maximization (EM)

Another conditional probability (Responsibility)

- In EM $p(z \mid x)$ plays a role (posterior in classification)
- The probability $p(z_k=1 \mid x)$ is denoted $\gamma(z_k)$
  - From Bayes theorem

$$\gamma(z_k) \equiv p(z_k = 1 \mid x) = \frac{p(z_k = 1)p(x \mid z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(x \mid z_j = 1)}$$

$$= \frac{\pi_k N(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x \mid \mu_k, \Sigma_j)}$$

  - View $p(z_k = 1) = \pi_k$ as prior probability of component *k*
    and $\gamma(z_k) = p(z_k = 1 \mid x)$ as the posterior probability

Maximum Likelihood for GMM

- We wish to model data set $\{\boldsymbol{x}_1, .. \boldsymbol{x}_N\}$ using a mixture of Gaussians (N items each of dimension D)

Find maximum likelihood parameters $\boldsymbol{\pi}_k$, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$

Likelihood Function for GMM

Mixture density function is

$$p(\mathrm{x}) = \sum_{\mathrm{z}} p(z)p(x \mid z) = \sum_{k=1}^{K} \pi_{\mathrm{k}} N\left(x \mid \mu_k, \Sigma_k\right)$$

Therefore Likelihood function is

$$p(X \mid \pi, \mu, \Sigma) = \prod_{n=1}^{N} \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}$$

Product is over the N i.i.d. samples

Therefore log-likelihood function is

$$\boxed{\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}}$$

Which we wish to maximize

A more difficult problem than for a single Gaussian

Maximization of Log-Likelihood

$$\boxed{\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}}$$

- Goal is to estimate the three sets of parameters

$$\pi_k, \mu_k, \Sigma_k$$

  - By taking derivatives in turn w.r.t each while keeping others constant
  - But there are no closed-form solutions
- While a gradient-based optimization is possible, we consider the iterative EM algorithm

EM for Gaussian Mixtures

- EM is a method for finding maximum likelihood solutions for models with latent variables
- Begin with log-likelihood function

$$\boxed{\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_n \mid \mu_k, \Sigma_k) \right\}}$$

  – We wish to find π, μ, Σ that maximize this quantity

$$Q(\theta, \theta^0) = E_{Z|X,\theta^0}[\log(P(X, Z|\theta))] = \sum_Z P(Z|X, \theta^0) \log(P(X, Z|\theta))$$

Instead take derivatives in turn of Q w.r.t Θ

  – Means μ_k and set to zero
  – covariance matrices Σ_k and set to zero
  – mixing coefficients π_k and set to zero

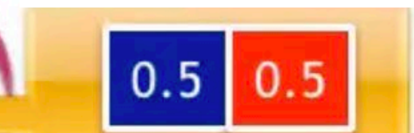
http://www.columbia.edu/~mh2078/MachineLearningORFE/EM_Algorithm.pdf

K-MEANS ALGORITHM REMINDER

1. Initialize means μ_k
   2. E Step: Assign each point to a cluster
   3. M Step: Given clusters, refine mean μ_k of each cluster k
4. Stop when change in means is small

EXPECTATION MAXIMIZATION (EM) FOR GAUSSIAN MIXTURES

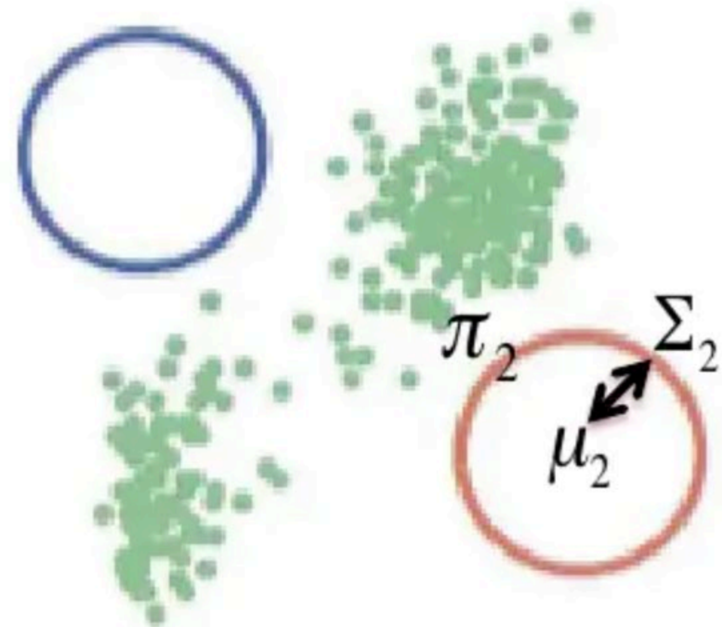
1. Initialize Gaussian* parameters: means μ_k , covariances Σ_k and mixing coefficients π_k
2. **E Step:** Assign each point X_n an assignment score $\gamma(z_{nk})$ for each cluster k
3. **M Step:** Given scores, adjust μ_k , π_k , Σ_k for each cluster k
4. Evaluate likelihood. If likelihood or parameters converge, stop.



*There are k Gaussians

EM FOR GAUSSIAN MIXTURES

1. Initialize μ_k , Σ_k π_k , one for each Gaussian k



- Tip! Use K-means result to initialize:

$$\mu_k \leftarrow \mu_k$$

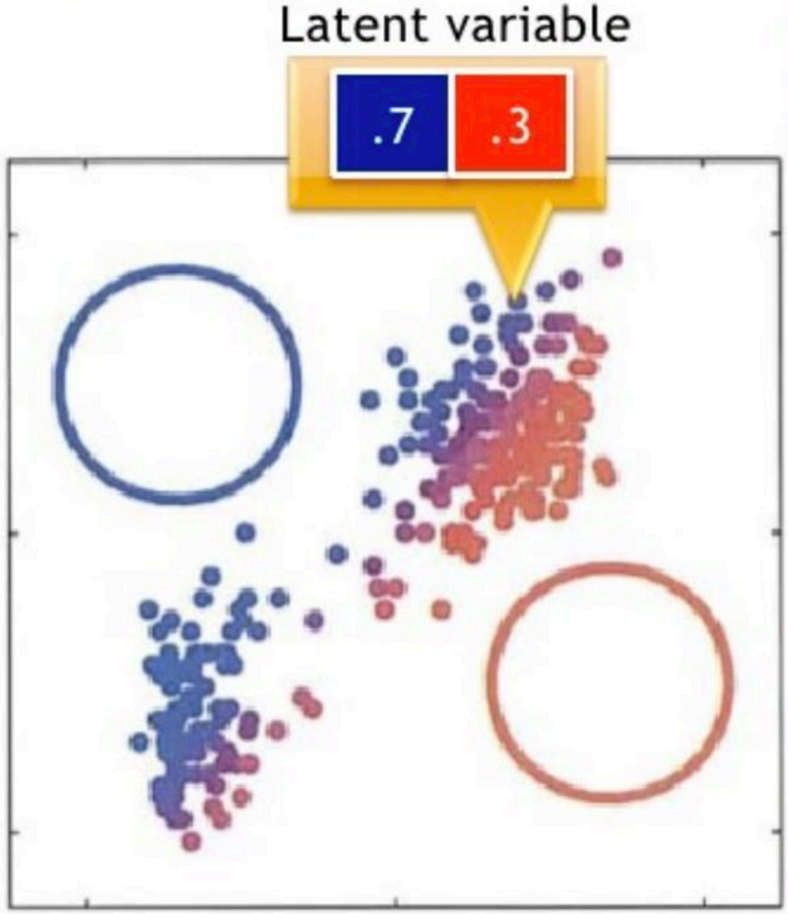
$$\Sigma_k \leftarrow \text{cov}(cluster(K))$$

$$\pi_k \leftarrow \frac{\text{Number of points in k}}{\text{Total number of points}}$$

EM FOR GAUSSIAN MIXTURES

2. **E Step:** For each point X_n, determine its assignment score to each Gaussian k:

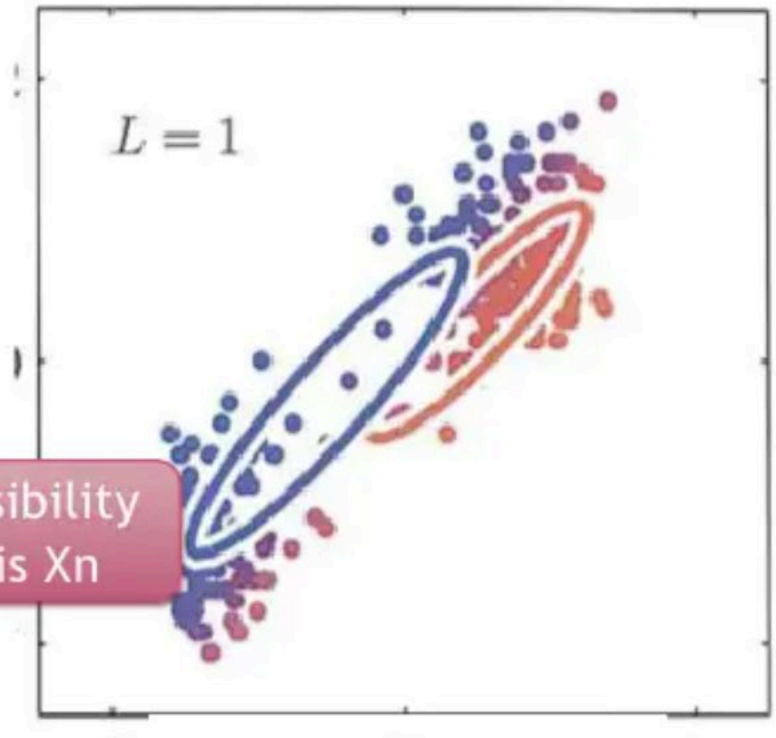
$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$



$\gamma(z_{nk})$ is called a "responsibility": how much is this Gaussian k responsible for this point X_n?

EM FOR GAUSSIAN MIXTURES

3. **M Step:** For each Gaussian k, update parameters using new $\gamma(z_{nk})$



Responsibility for this Xn

Mean of Gaussian k

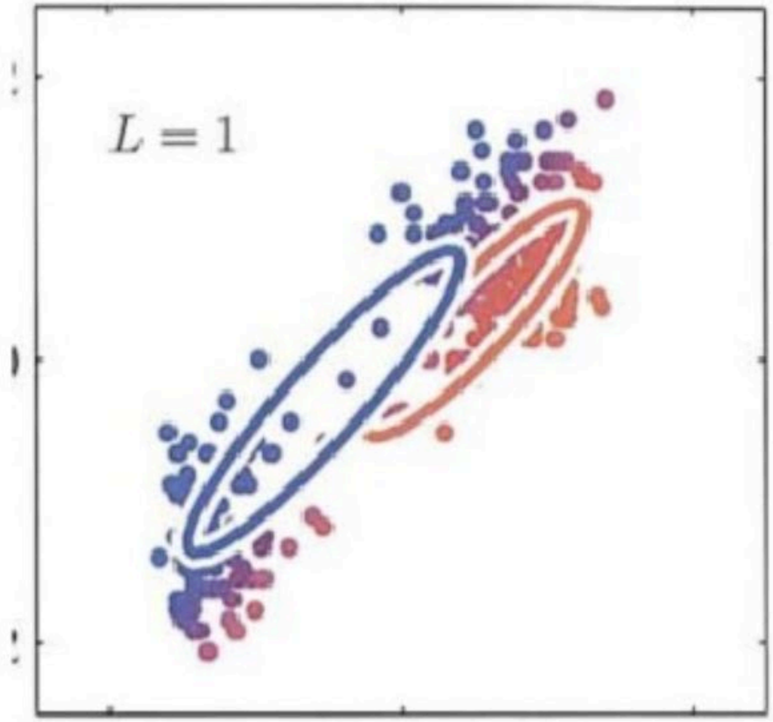
$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Find the mean that “fits” the assignment scores best

EM FOR GAUSSIAN MIXTURES

3. **M Step:** For each Gaussian k, update parameters using new $\gamma(z_{nk})$



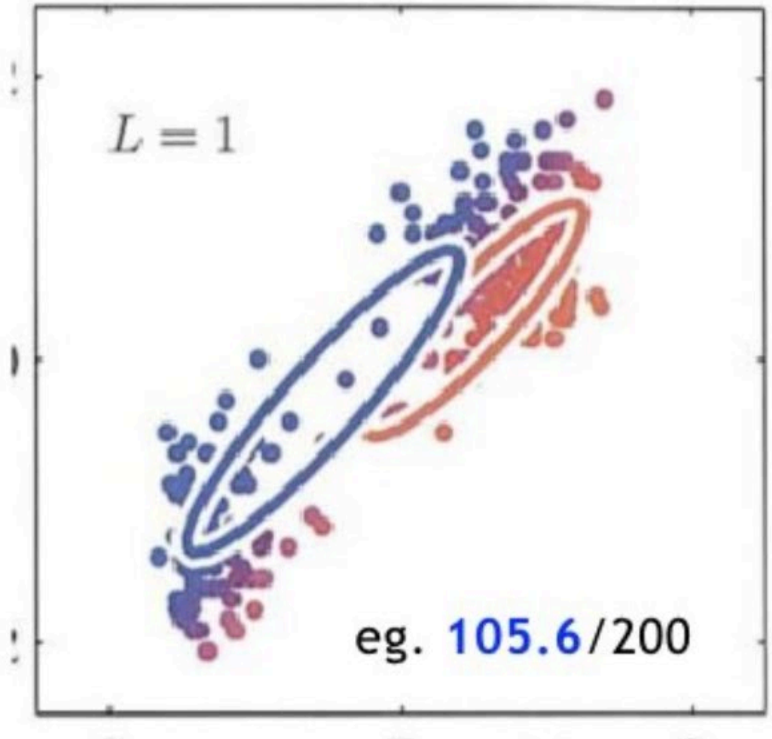
Covariance matrix of Gaussian k

$$\Sigma_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

Just calculated this!

EM FOR GAUSSIAN MIXTURES

3. **M Step:** For each Gaussian k, update parameters using new $\gamma(z_{nk})$



Mixing Coefficient for Gaussian k

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

Total # of points

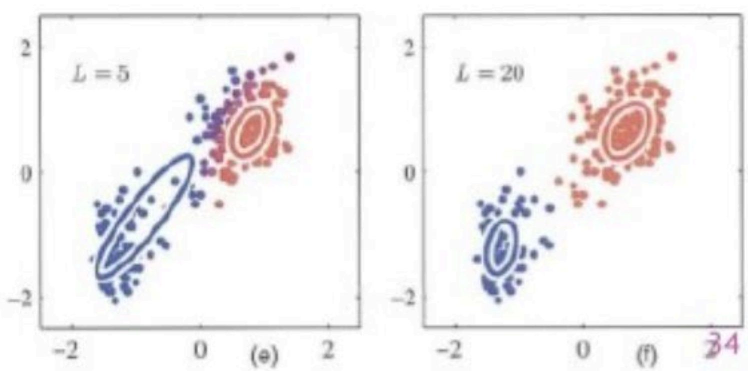
$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

EM FOR GAUSSIAN MIXTURES

4. Evaluate log **likelihood**. If likelihood or parameters converge, stop. Else go to Step 2 (**E step**).

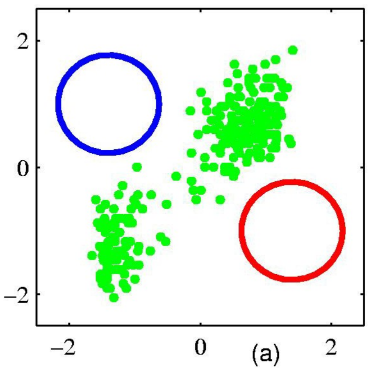
$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \underline{\pi_k} \mathcal{N}(\mathbf{x}_n | \underline{\boldsymbol{\mu}_k}, \underline{\boldsymbol{\Sigma}_k}) \right\}$$

Likelihood is the probability that the data X was generated by the parameters you found. ie. Correctness!

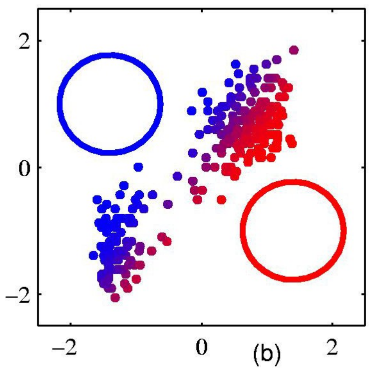


EM Example

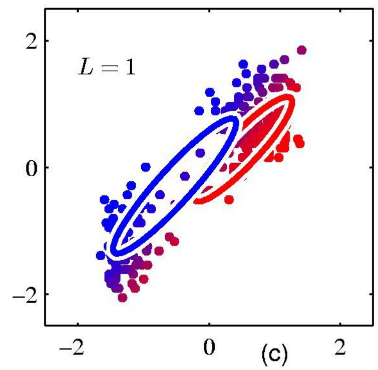
Data points and
Initial mixture model



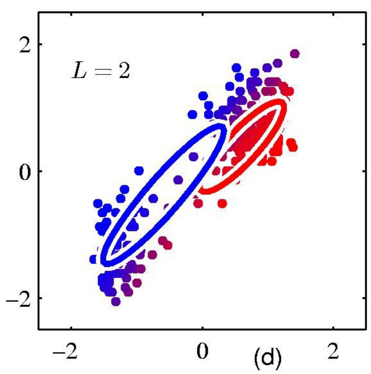
Initial E step
Determine
responsibilities



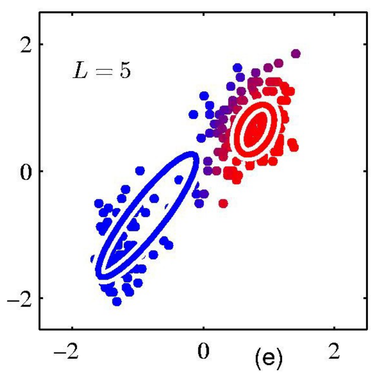
After first M step
Re-evaluate Parameters



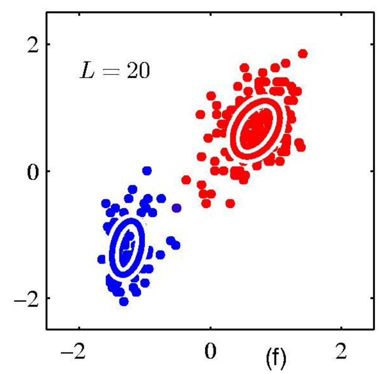
After 2 cycles



After 5 cycles

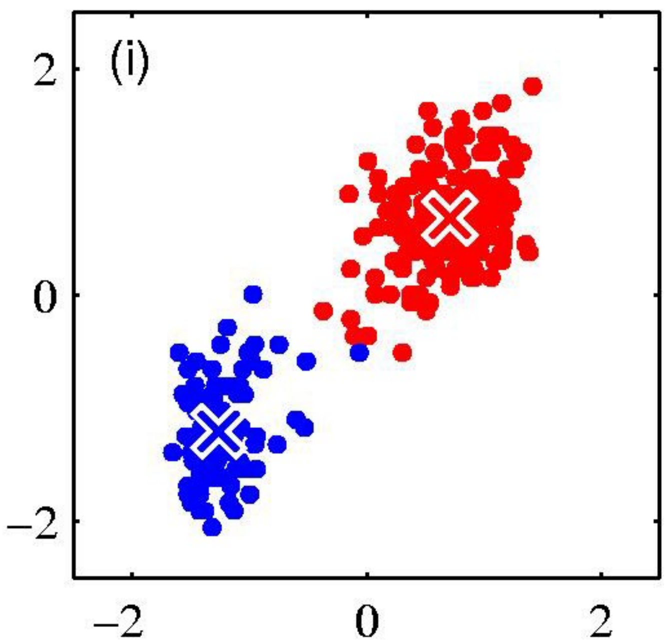


After 20 cycles

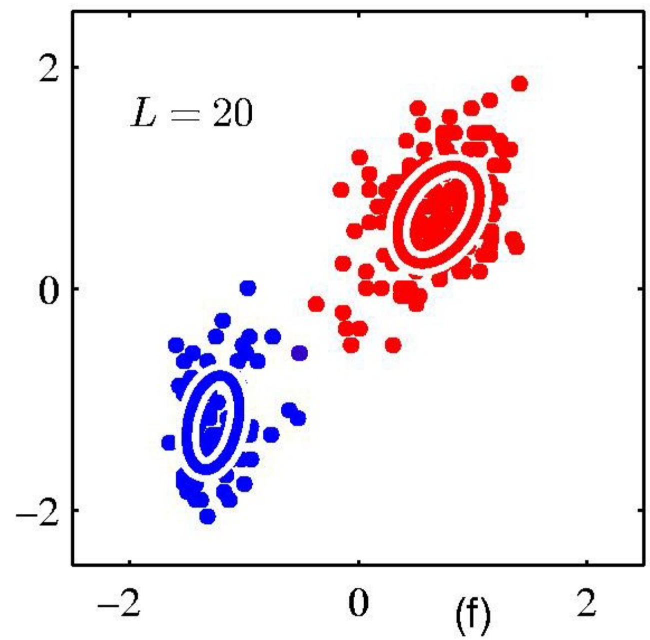


Comparison with K-Means

K-means result



E-M result



Practical Issues with EM

- Takes many more iterations than *K*-means
  - Each cycle requires significantly more comparison
- Common to run *K*-means first in order to find suitable initialization
- Covariance matrices can be initialized to covariances of clusters found by *K*-means
- EM is not guaranteed to find global maximum of log likelihood function

```
>>> import numpy as np
>>> from sklearn.mixture import GaussianMixture
>>> X = np.array([[1, 2], [1, 4], [1, 0], [10, 2], [10, 4], [10, 0]])
>>> gm = GaussianMixture(n_components=2, random_state=0).fit(X)
>>> gm.means_
array([[10.,  2.],
       [ 1.,  2.]])
>>> gm.predict([[0, 0], [12, 3]])
array([1, 0])
```

Attributes:	**weights_ : *array-like of shape (n_components,)*** The weights of each mixture components. **means_ : *array-like of shape (n_components, n_features)*** The mean of each mixture component. **covariances_ : *array-like*** The covariance of each mixture component.

https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

Summary of EM for GMM

- Given a Gaussian mixture model
- Goal is to maximize the likelihood function w.r.t. the parameters (means, covariances and mixing coefficients)

Step1: Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k and evaluate initial value of log-likelihood

EM continued

- Step 2: E step: Evaluate responsibilities using current parameter values

$$\boxed{\gamma(z_k) = \frac{\pi_k N(\boldsymbol{x}_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(\boldsymbol{x}_n \mid \mu_j, \Sigma_j))}}$$

- Step 3: M Step: Re-estimate parameters using current responsibilities

$$\boxed{\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathrm{x}_n}$$

$$\boxed{\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathrm{x}_n - \mu_k^{\text{new}})(\mathrm{x}_n - \mu_k^{\text{new}})^T}$$

$$\boxed{\pi_k^{\text{new}} = \frac{N_k}{N}}$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$

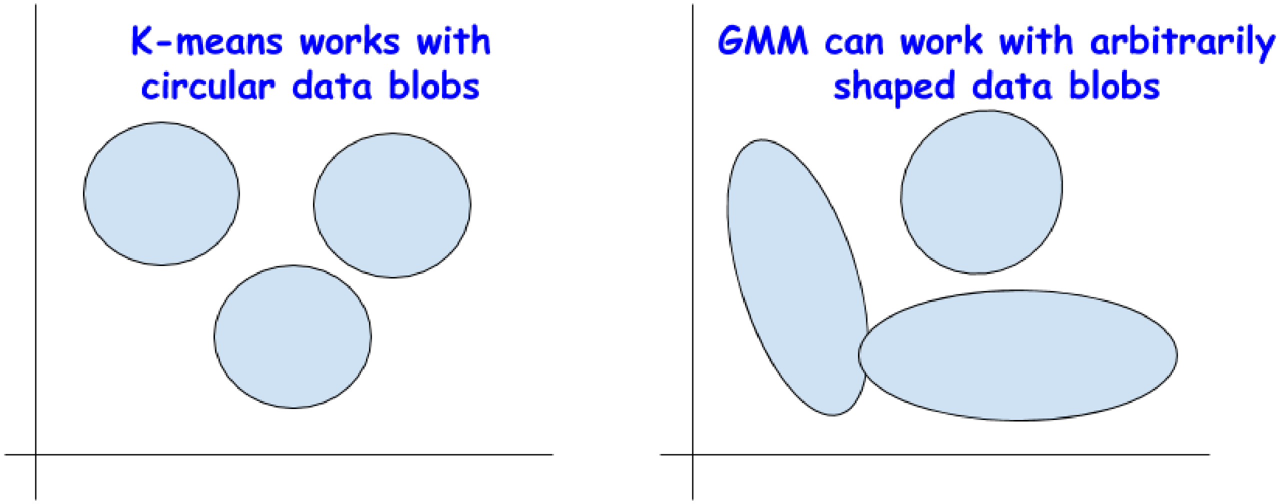
EM Continued

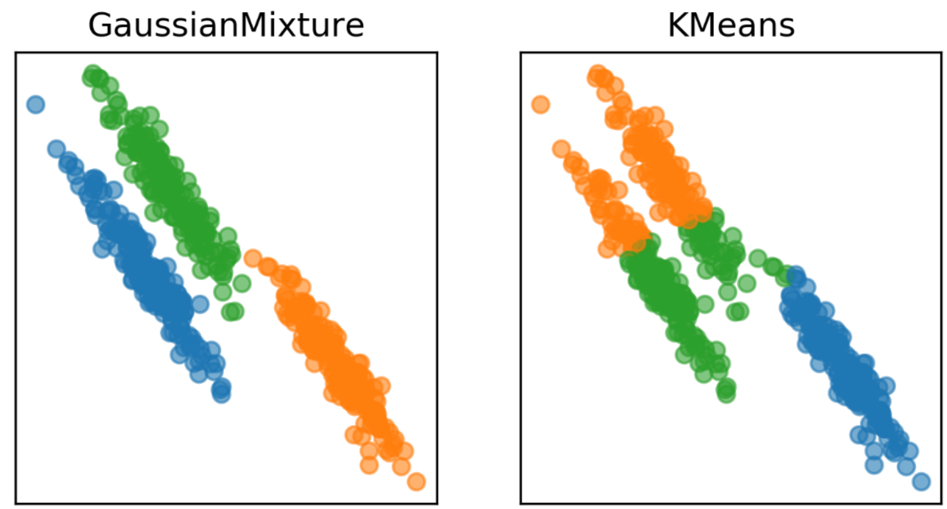
- Step 4: Evaluate the log likelihood

$$\boxed{\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(\mathrm{x}_n \mid \mu_k, \Sigma_k) \right\}}$$

  – And check for convergence of either parameters or log likelihood

- If convergence not satisfied return to Step 2





Different cluster analysis results on "mouse" data set: