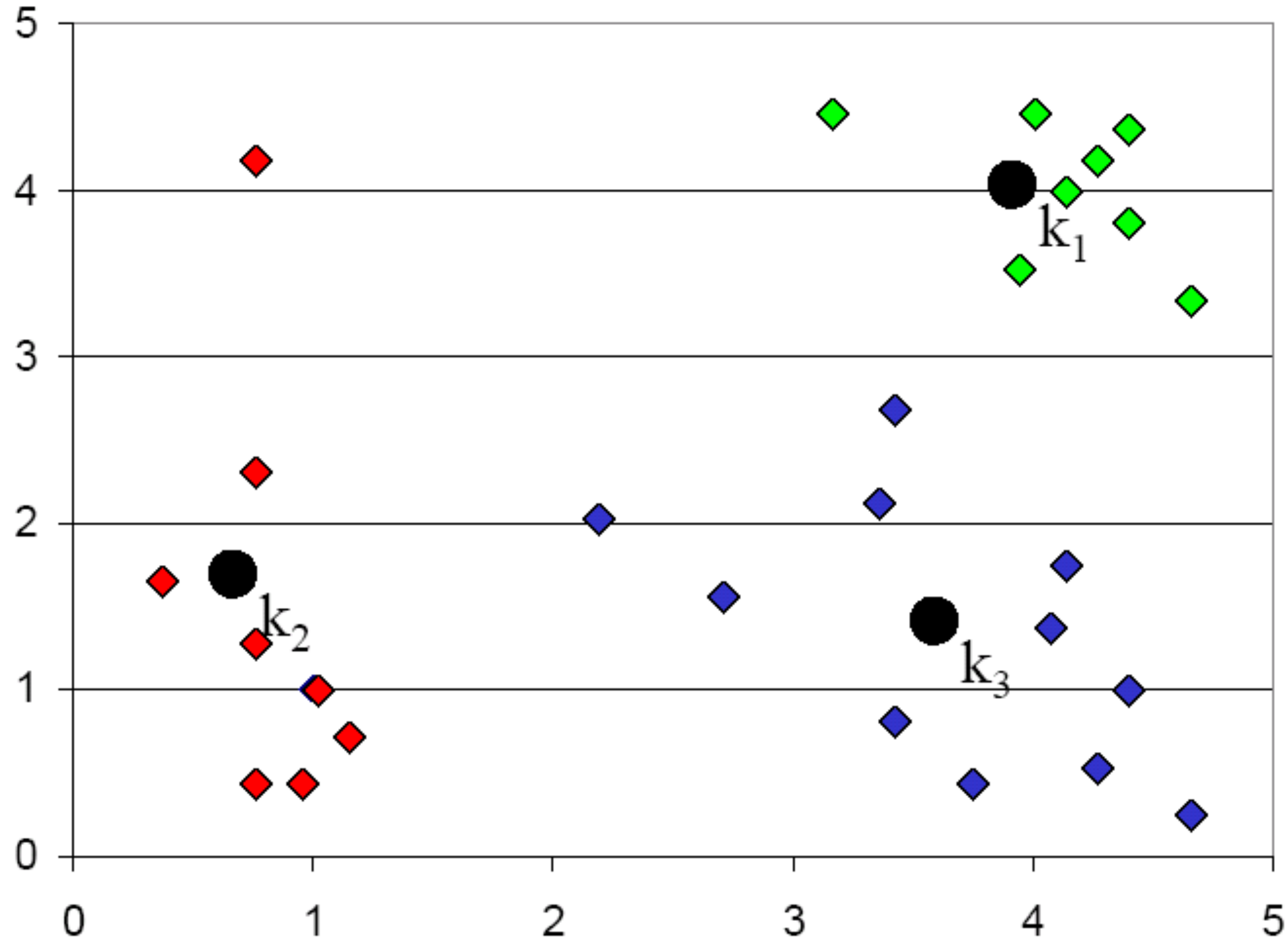
Clustering: Mixture Models

Machine Learning 10-601B

Seyoung Kim

Many of these slides are derived from Tom Mitchell, Ziv-Bar Joseph, and Eric Xing. Thanks!

Problem with K-means



Hard Assignment of Samples into Three Clusters

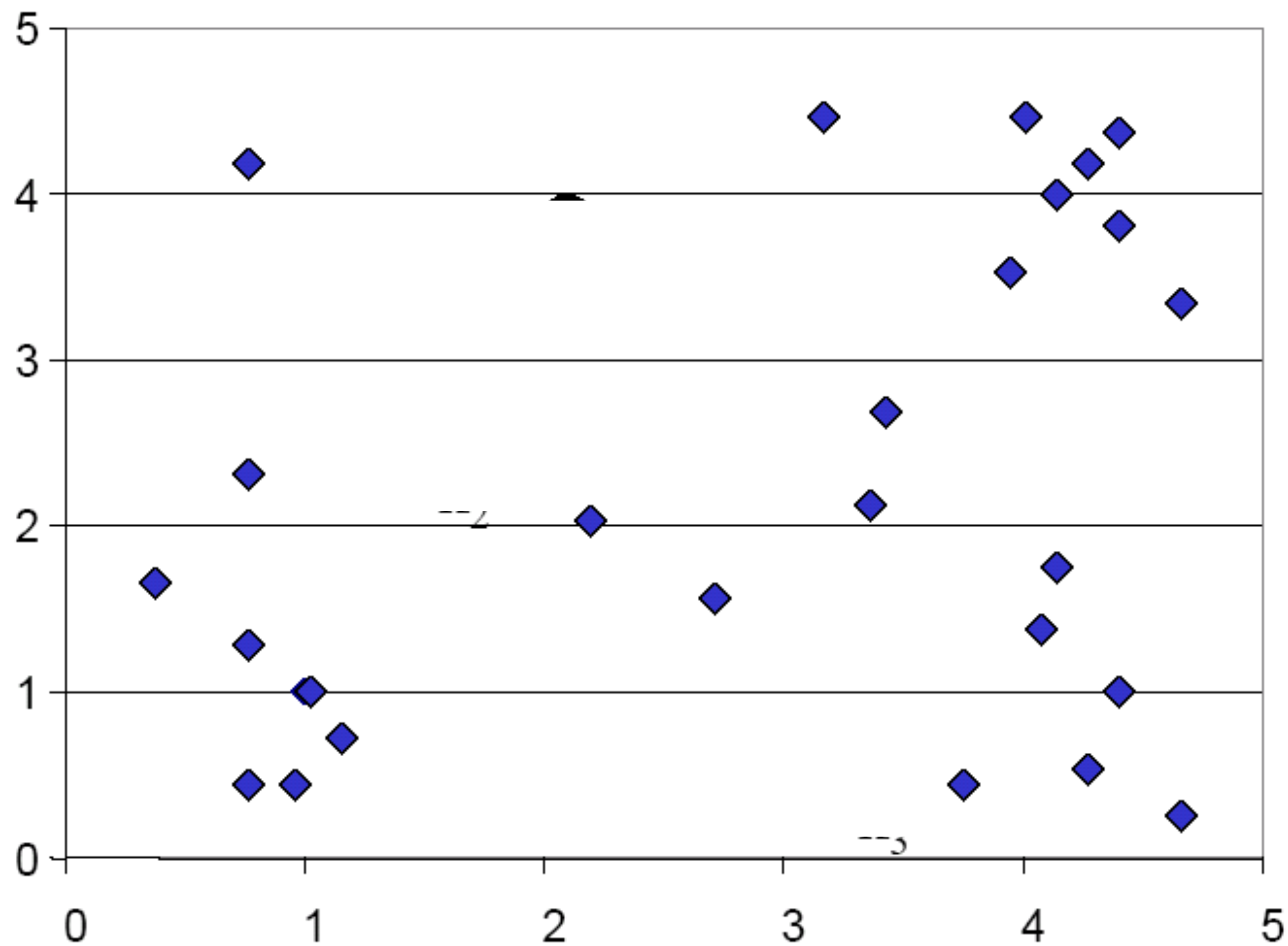
	Cluster 1	Cluster 2	Cluster 3
Individual 1	1	0	0
Individual 2	0	1	0
Individual 3	0	1	0
Individual 4	1	0	0
Individual 5	…	…	…
Individual 6	…	…	…
Individual 7	…	…	…
Individual 8	…	…	…
Individual 9	…	…	…
Individual 10	…	…	…

Probabilistic Soft-Clustering of Samples into Three Clusters

Probability of	Cluster 1	Cluster 2	Cluster 3	Sum
Individual 1	0.1	0.4	0.5	1
Individual 2	0.8	0.1	0.1	1
Individual 3	0.7	0.2	0.1	1
Individual 4	0.10	0.05	0.85	1
Individual 5	…	…	…	1
Individual 6	…	…	…	1
Individual 7	…	…	…	1
Individual 8	…	…	…	1
Individual 9	…	…	…	1
Individual 10	…	…	…	1

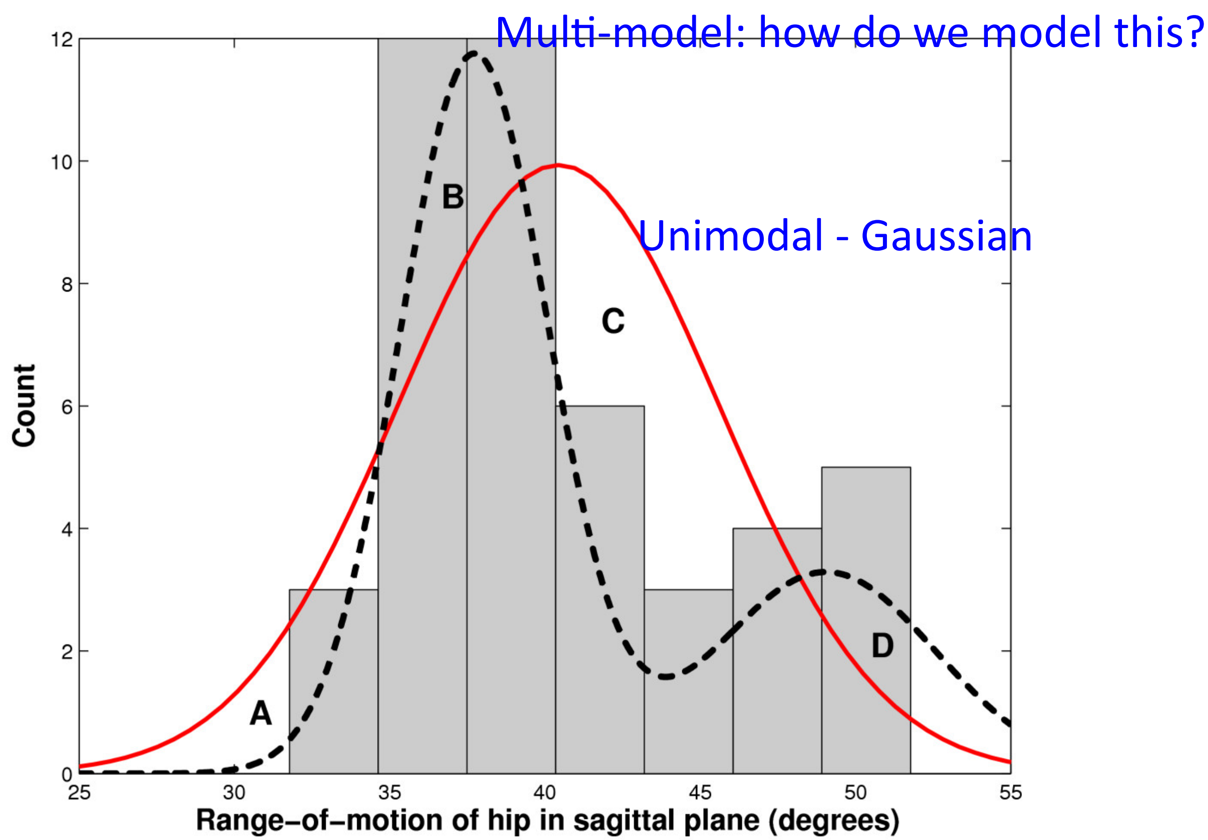
- Each sample can be assigned to more than one clusters with a certain probability.
- For each sample, the probabilities for all clusters should sum to 1. (i.e., each row should sum to 1.)
- Each cluster is explained by a cluster center variable (i.e., cluster mean)

Probability Model for Data P(X)?



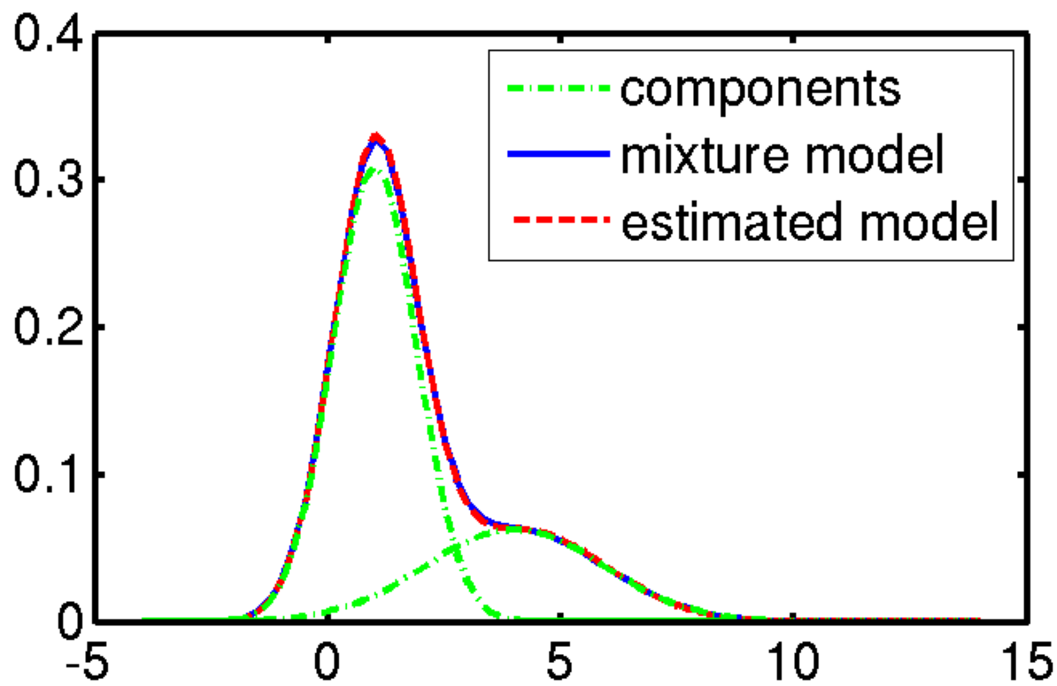
Mixture Model

- A density model $\mathbf{p}(\mathbf{x})$ may be multi-modal.



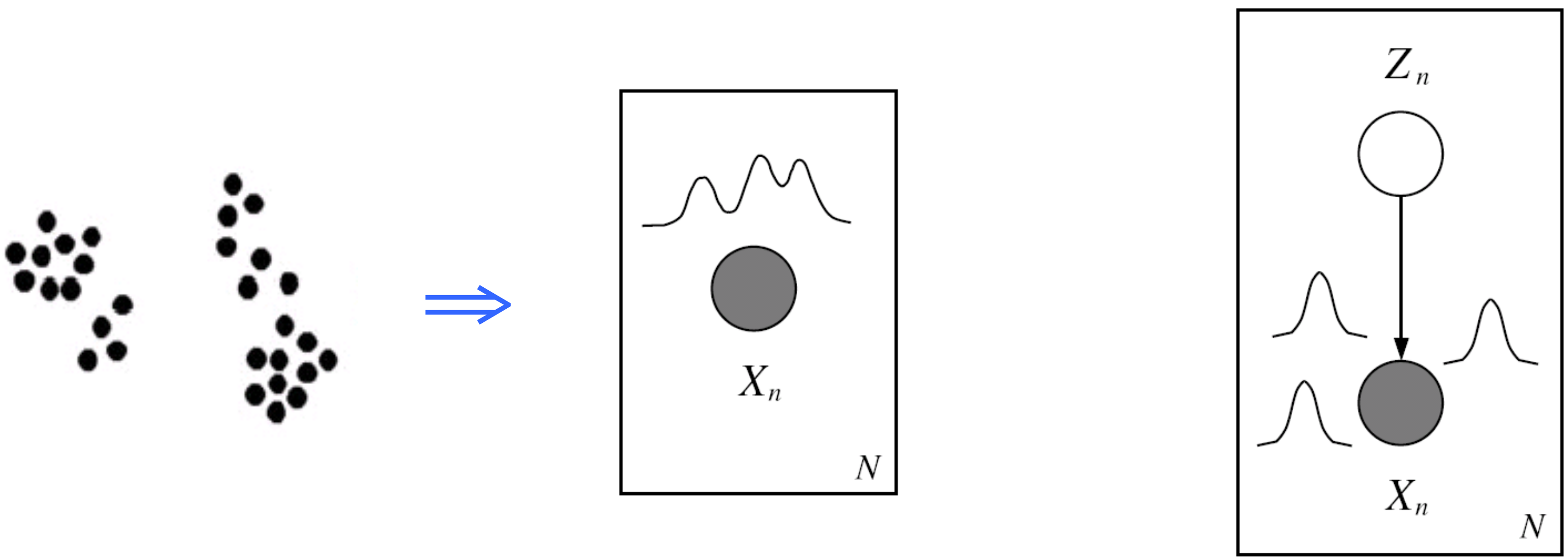
Mixture Model

- We may be able to model it as a mixture of uni-modal distributions (e.g., Gaussians).
- Each mode may correspond to a different sub-population (e.g., male and female).



Learning Mixture Models from Data

- Given data generated from multi-modal distribution, can we find a representation of the multi-model distribution as a mixture of uni-modal distributions?



(a) (b)

Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

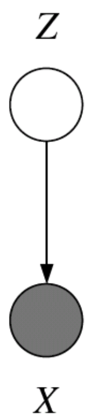
$$p(x_n) = \sum_k p(x_n \mid z_n = k) p(z_n = k)$$

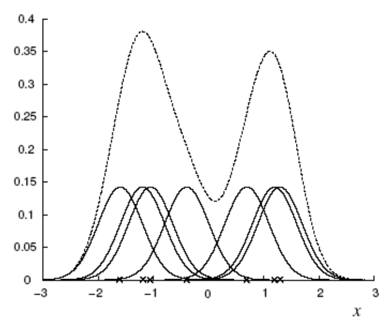
贝叶斯条件概率

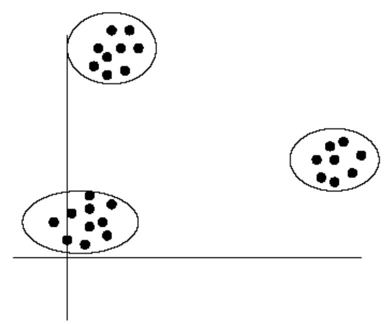
$$= \sum_k N(x_n \mid \mu_k, \Sigma_k) \pi_k$$

mixture proportion各个cluster占总样本的比例

mixture component







Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

$$p(x_n) = \sum_k p(x_n, z_n = k) p(z_n = k)$$

$$= \sum_k \underbrace{N(x_n \mid \mu_k, \Sigma_k)}_{\text{mixture component}} \underbrace{\pi_k}_{\text{mixture proportion}}$$

- This probability model describes how each data point x_n can be generated
  - Step 1: Flip a *K*-sided die (with probability π_k for the *k*-th side) to select a cluster c
  - Step 2: Generate the values of the data point from $N(\mu_c, \Sigma_c)$

Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

$$p(x_n) = \sum_k p(x_n, z_n = k)\, p(z_n = k)$$

$$= \sum_k N(x_n \mid \mu_k, \Sigma_k)\, \pi_k$$

mixture proportion mixture component

- Parameters for K clusters: $\theta = \{\mu_k, \Sigma_k, \pi_k, k = 1, \ldots, K\}$

Learning mixture models

- Latent variable model: data are only partially observed!
  - x_i: observed sample data
  - $z_i = \{z_i^1 \dots z_i^K\}$: Unobserved cluster labels (each element 0 or 1, only one of them is 1)

- MLE estimate
  - What if all data (x_i, z_i) are observed?
    - Maximize the data log likelihood for (x_i, z_i) based on $p(x_i, z_i)$
    - Easy to optimize!
  - In practice, only x_i's are observed
    - Maximize the data log likelihood for (x_i) based on $p(x_i)$
    - Difficult to optimize!
  - Maximize the expected data log likelihood for (x_i, z_i) based on $p(x_i, z_i)$
    - Expectation-Maximization (EM) algorithm

Learning mixture models: fully observed data

- In fully observed iid settings, assuming the **cluster labels z_i's were observed**, the log likelihood decomposes into a sum of local terms.

$$l_c(\theta; D) = \sum_n \log p(x_n, z_n \mid \theta) = \sum_n \log p(z_n \mid \theta) + \sum_n \log p(x_n \mid z_n, \theta)$$

Depends on π_k Depends on μ_k, Σ_k

  - The optimization problems for μ_k, Σ_k and for π_k are decoupled, and a closed-form solution for MLE exists.

MLE for GMM with fully observed data

- If we are doing MLE for completely observed data
- Data log-likelihood

$$l(\theta;D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n \mid \pi) p(x_n \mid z_n, \mu, \sigma)$$

$$= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_n^k}$$

$$= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k z_n^k \frac{1}{2\sigma^2} (x_n - \mu_k)^2 + C$$

- MLE

$$\hat{\pi}_{k,MLE} = \arg\max_{\pi} \boldsymbol{l}(\boldsymbol{\theta}; D),$$

$$\hat{\mu}_{k,MLE} = \arg\max_{\mu} \boldsymbol{l}(\boldsymbol{\theta}; D) \qquad ß \quad \hat{\mu}_{k,MLE} = \frac{\sum_n z_n^k x_n}{\sum_n z_n^k}$$

$$\hat{\sigma}_{k,MLE} = \arg\max_{\sigma} \boldsymbol{l}(\boldsymbol{\theta}; D)$$

- What if we do not know z_n?

Learning mixture models

- In fully observed iid settings, assuming the cluster labels z_i's were observed, the log likelihood decomposes into a sum of local terms.

$$l_c(\theta; D) = \sum_n \log p(x_n, z_n \mid \theta)$$

- With latent variables for cluster labels

$$l_c(\theta; D) = \sum_n \log p(x_n \mid \theta)$$

$$= \sum_n \log \sum_z p(x_n, z \mid \theta) = \sum_n \log \sum_z p(z \mid \theta) p(x_n \mid z, \theta)$$

  - all the parameters become coupled together via *marginalization*

- Are they equally difficult?

Depends on π_k

Depends on μ_k, Σ_k

Theory underlying EM

- Recall that according to MLE, we intend to learn the model parameter that would have maximized the likelihood of the data.

- But we do not observe z, so computing

$$l_c(\theta;D) = \sum_n \log \sum_z p(x_n, z \mid \theta) = \sum_n \log \sum_z p(z \mid \theta) p(x_n \mid z, \theta)$$

  is difficult!

- Optimizing the log-likelihood for MLE is difficult!

- What shall we do?

Complete vs. Expected Complete Log Likelihoods

- The complete log likelihood:

$$\boldsymbol{l}(\boldsymbol{\theta}; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n \mid \pi) p(x_n \mid z_n, \mu, \sigma)$$

$$= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_n^k}$$

$$= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k z_n^k \tfrac{1}{2\sigma^2} (x_n - \mu_k)^2 + C$$

- The expected complete log likelihood

$$\langle \boldsymbol{l}_c(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{z}) \rangle = \sum_n \langle \log p(\boldsymbol{z}_n \mid \pi) \rangle_{p(z|x)} + \sum_n \langle \log p(\boldsymbol{x}_n \mid \boldsymbol{z}_n, \mu, \Sigma) \rangle_{p(z|x)}$$

$$= \sum_n \sum_k \langle z_n^k \rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \langle z_n^k \rangle \left((x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + \log |\Sigma_k| + C \right)$$

Depends on π_k

Depends on μ_k, Σ_k

Complete vs. Expected Complete Log Likelihoods

- The complete log likelihood:

$$\boldsymbol{l}(\boldsymbol{\theta}; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n \mid \pi) p(x_n \mid z_n, \mu, \sigma)$$

$$= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_n^k}$$

$$= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k z_n^k \tfrac{1}{2\sigma^2}(x_n - \mu_k)^2 + C$$

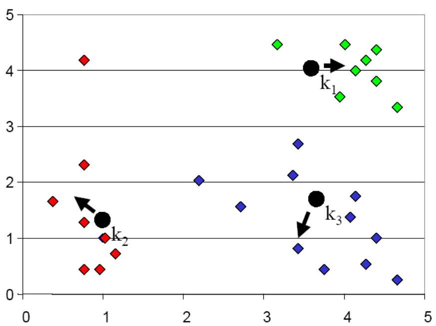
- The expected complete log likelihood

$$\left\langle \boldsymbol{l}_c(\boldsymbol{\theta}; x, z) \right\rangle = \sum_n \left\langle \log p(z_n \mid \pi) \right\rangle_{p(z|x)} + \sum_n \left\langle \log p(x_n \mid z_n, \mu, \Sigma) \right\rangle_{p(z|x)}$$

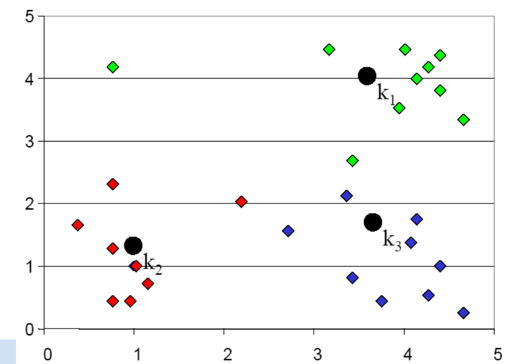
$$= \sum_n \sum_k \left\langle z_n^k \right\rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \left\langle z_n^k \right\rangle \left((x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + \log |\Sigma_k| + C \right)$$

- EM optimizes the expected complete log likelihood

EM Algorithm



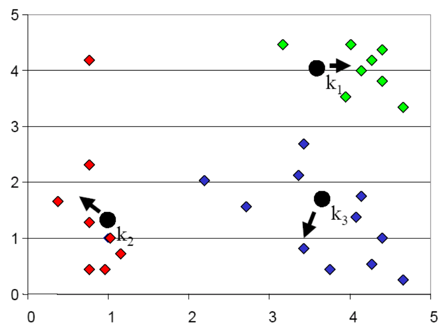
Maximization (M)-step:
- Find mixture parameters



Expectation (E)-step:
-Re-assign samples x_i's to clusters
- Impute the unobserved values z_i

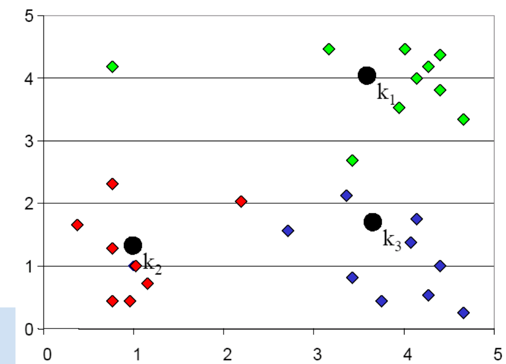
Iterate until convergence

K-Means Clustering Algorithm



Find the cluster means

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$



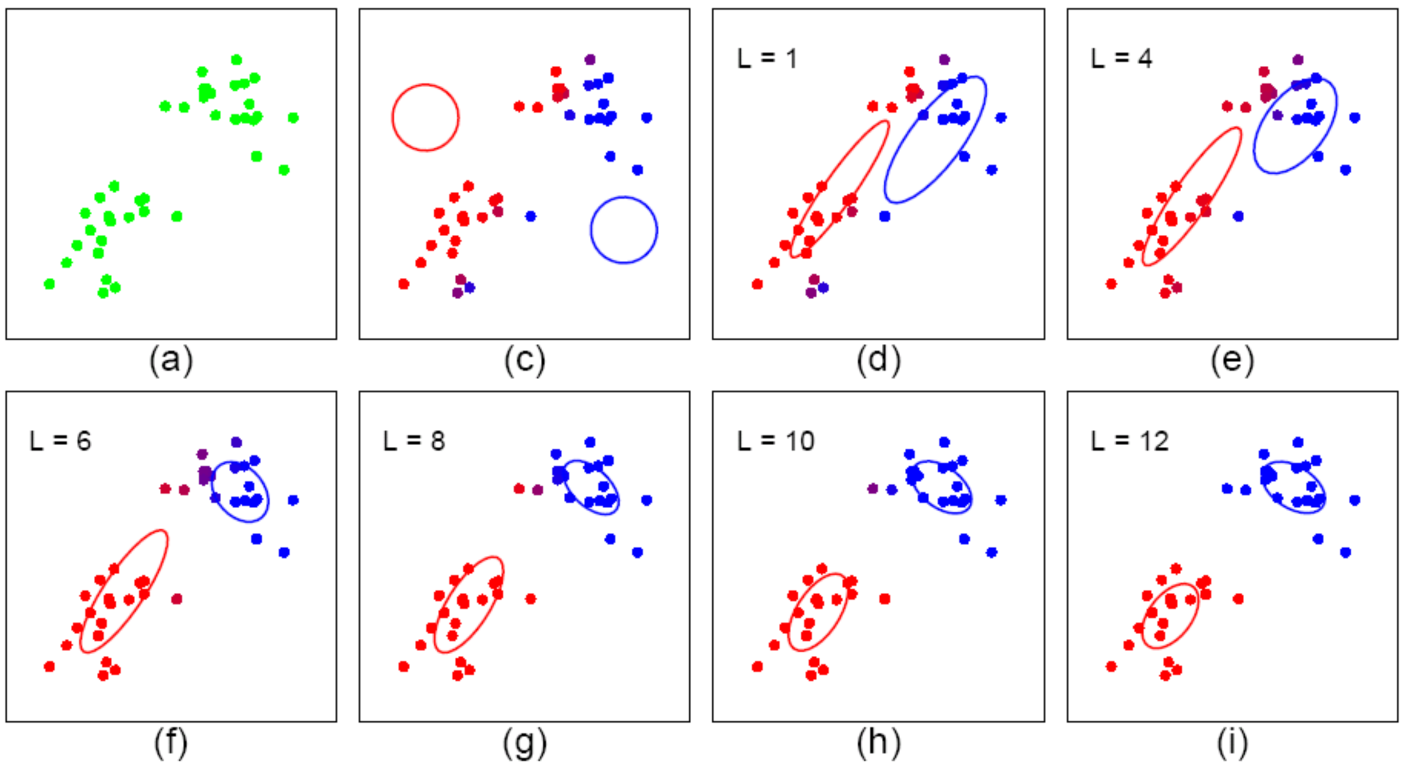
Re-assign samples x_i's to clusters

$$\arg\max_k \| x_i - \mu_k \|_2^2$$

Iterate until convergence

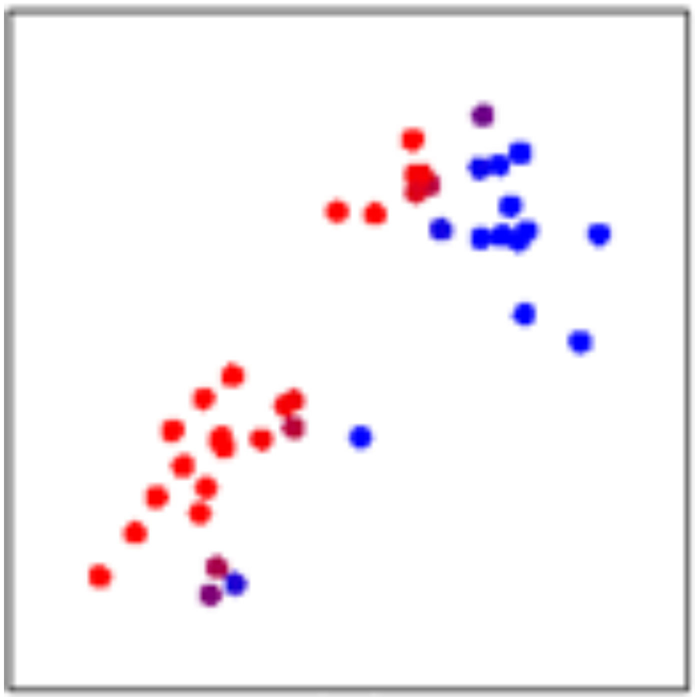
The Expectation-Maximization (EM) Algorithm

- Start:
  - "Guess" the centroid μ_k and covariance Σ_k of each of the K clusters
- Loop



The Expectation-Maximization (EM) Algorithm

- A "soft" k-means



E step:

$$\tau_n^{k(t)} = \left\langle z_n^k \right\rangle_{q^{(t)}} = p(z_n^k = 1 \mid x, \mu^{(t)}, \Sigma^{(t)})$$

M step:

$$\pi_k^{(t+1)} = \sum_n \tau_n^{k(t)} \Big/ N = \left\langle n_k \right\rangle \Big/ N$$

$$\mu_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} x_n}{\sum_n \tau_n^{k(t)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^T}{\sum_n \tau_n^{k(t)}}$$

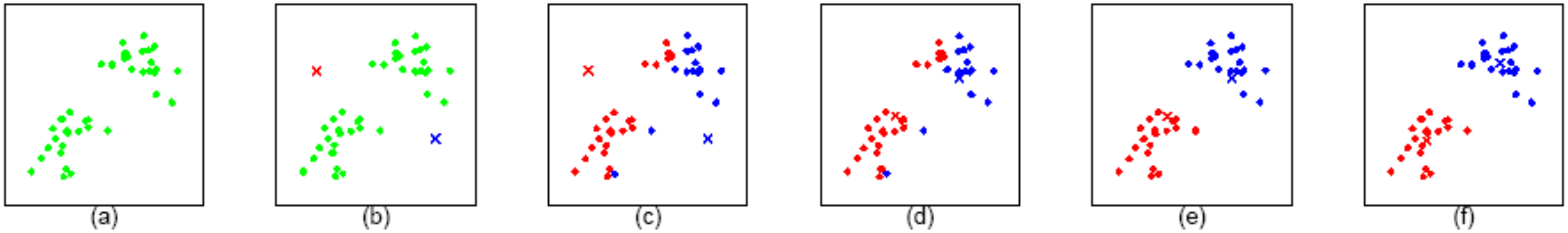
Compare: K-means

- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means “E-step” we do hard assignment:

$$z_n^{(t)} = \arg\max_k (x_n - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})$$

- In the K-means “M-step” we update the means as the weighted sum of the data, but now the weights are 0 or 1:

$$\mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k) x_n}{\sum_n \delta(z_n^{(t)}, k)}$$



Expected Complete Log Likelihood Lower-bounds Complete Log Likelihood

- For ***any*** distribution **$q(z)$**, define *expected complete log likelihood*:

$$\left\langle l_c(\theta; x, z) \right\rangle_q \stackrel{\text{def}}{=} \sum_z q(z \mid x, \theta) \log p(x, z \mid \theta)$$

  – Does maximizing this surrogate yield a maximizer of the likelihood?

- Jensen's inequality

$$\begin{aligned} l(\theta; x) &= \log p(x \mid \theta) \\ &= \log \sum_z p(x, z \mid \theta) \\ &= \log \sum_z q(z \mid x) \frac{p(x, z \mid \theta)}{q(z \mid x)} \\ &\geq \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \end{aligned} \qquad \Rightarrow \qquad l(\theta; x) \geq \left\langle l_c(\theta; x, z) \right\rangle_q + H_q$$

Closing notes

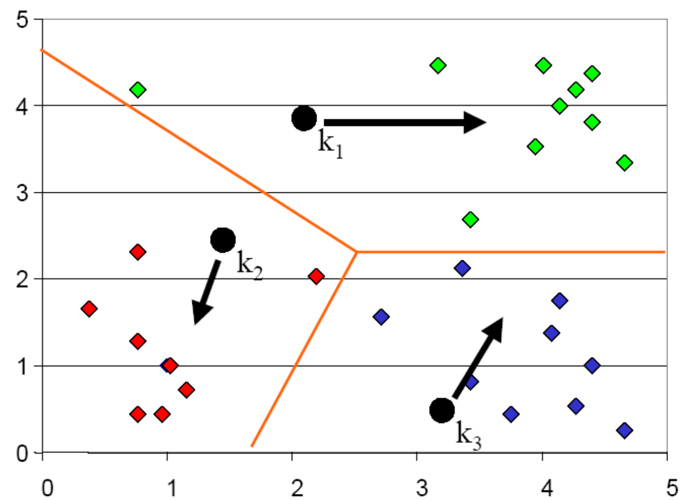
- Convergence
- Seed choice
- Quality of cluster
- How many clusters

Convergence

- Why should the K-means algorithm ever reach a fixed point?
  - -- A state in which clusters don’t change.
- K-means is a special case of a general procedure the Expectation Maximization (EM) algorithm.
  - Both are known to converge.
  - Number of iterations could be large.

Seed Choice

- Results can vary based on random seed selection.



- Some seeds can result in convergence to sub-optimal clusterings.
  – Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  – Try out multiple starting points (very important!!!)
  – Initialize with the results of another method.

What Is A Good Clustering?

- Internal criterion: A good clustering will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high
  - the inter-class similarity is low
  - The measured quality of a clustering depends on both the obj representation and the similarity measure used

- External criteria for clustering quality
  - Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
  - Assesses a clustering with respect to ground truth

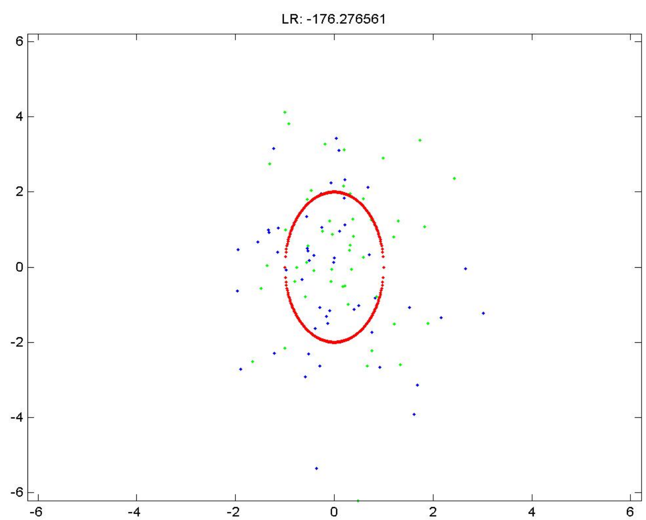
How Many Clusters?

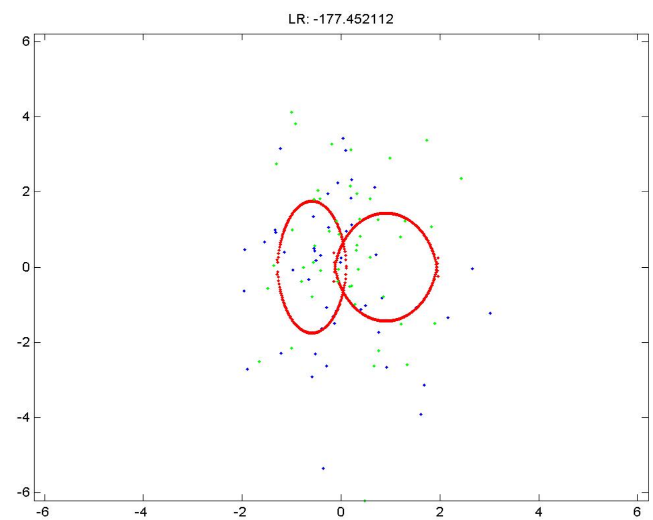
- Number of clusters K is given
  - Partition n docs into predetermined number of clusters
- Finding the “right” number of clusters is part of the problem
  - Given objs, partition into an “appropriate” number of subsets.
  - E.g., for query results - ideal value of K not known up front - though UI may impose limits.
- Tradeoff between having more clusters (better focus within each cluster) and having too many clusters
- Nonparametric Bayesian Inference

Cross validation

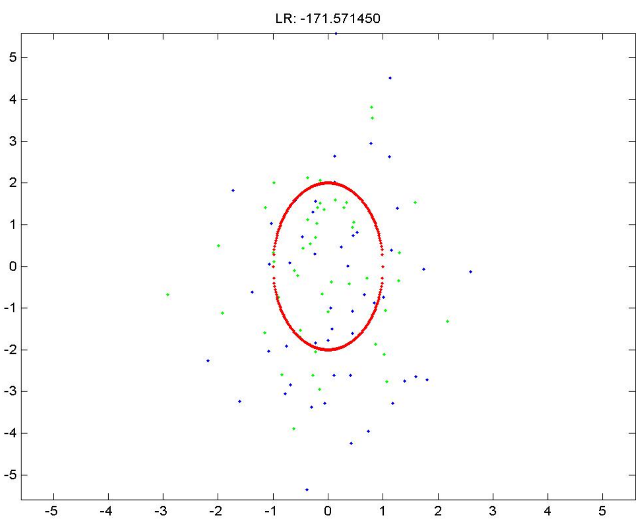
- We can also use cross validation to determine the correct number of classes
- Recall that GMMs is a generative model. We can compute the likelihood of the held-out data to determine which model (number of clusters) is more accurate

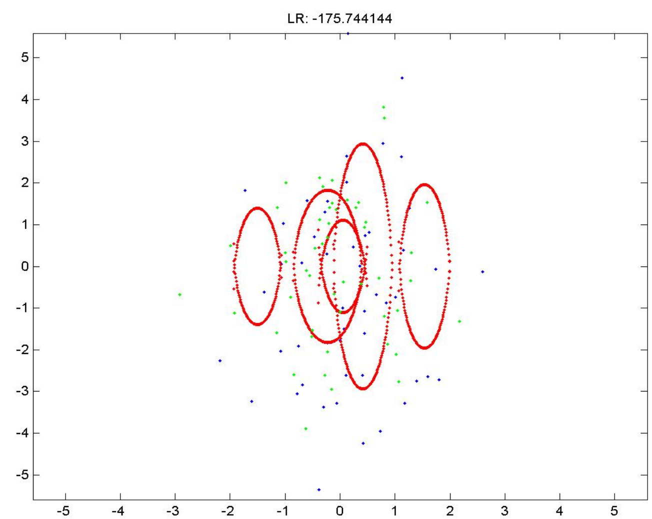
$$p(x_1 \cdots x_n \mid \theta) = \prod_{j=1}^{n} \left(\sum_{i=1}^{k} p(x_j \mid C = i) w_i \right)$$



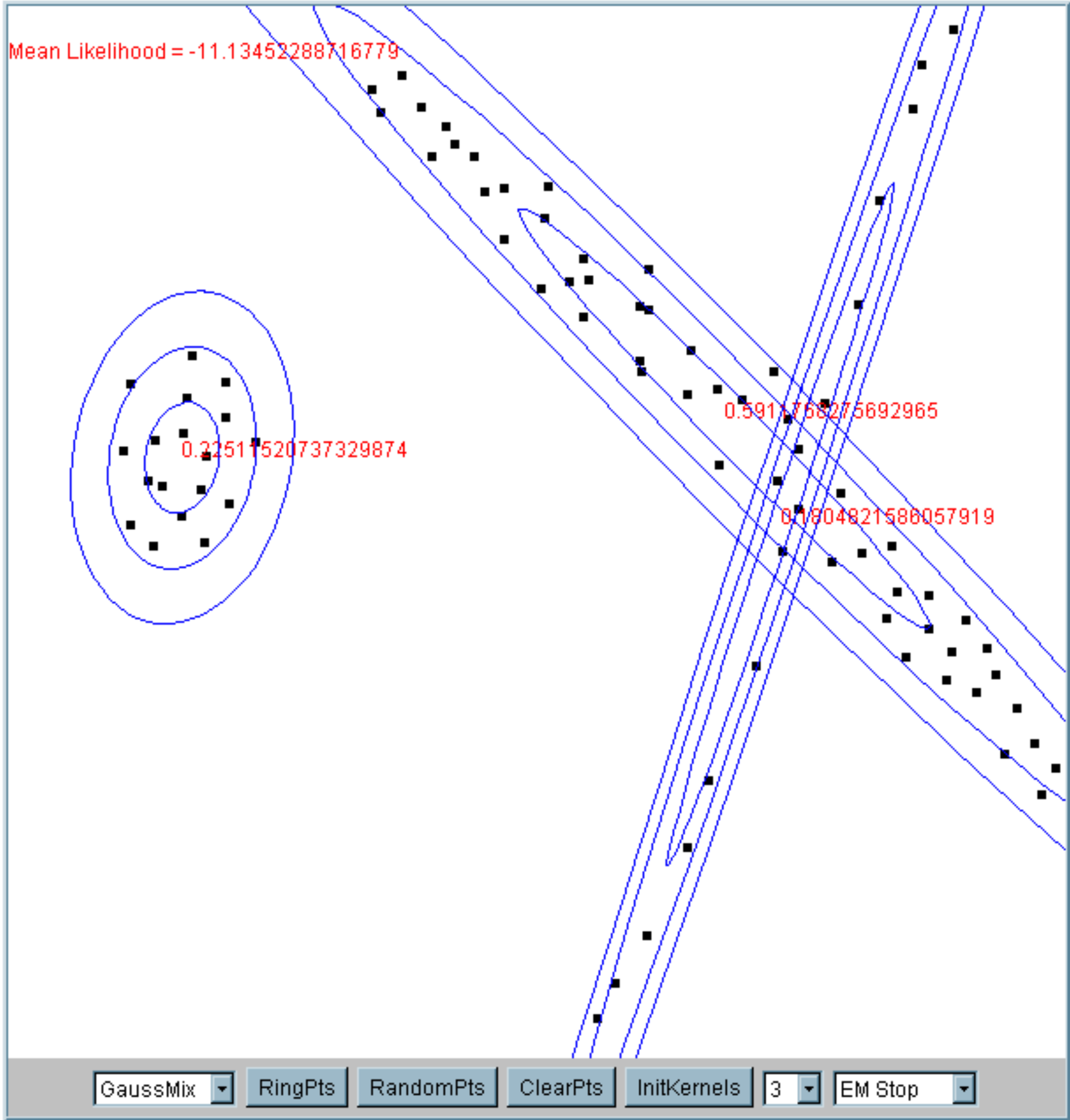


Cross validation





Gaussian mixture clustering



Clustering methods: Comparison

	Hierarchical	**K-means**	**GMM**
Running time	naively, O(N^3)	fastest (each iteration is linear)	fast (each iteration is linear)
Assumptions	requires a similarity / distance measure	strong assumptions	strongest assumptions
Input parameters	none	K (number of clusters)	K (number of clusters)
Clusters	subjective (only a tree is returned)	exactly K clusters	exactly K clusters

What you should know about Mixture Models

- Gaussian mixture models
  - Probabilistic extension of K-means for soft-clustering
  - EM algorithm for learning by assuming data are only partially observed
    - Cluster labels are treated as the unobserved part of data

- EM algorithm for learning from partly unobserved data
  - MLE of θ = $\arg\max_\theta \log P(data|\theta)$
  - EM estimate: θ = $\arg\max_\theta E_{Z|X,\theta}[\log P(X,Z|\theta)]$
    - Where X is observed part of data, Z is unobserved