Expectation Maximization

Administrivia

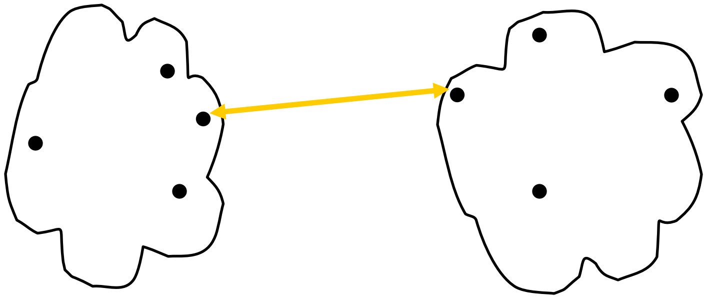
- NEW CHANGE on 16th Novemeber, 2022

```
Add the line below
'''
torch.cuda.empty_cache()
torch.manual_seed(0)
'''
Add the above line
'''
```

From last class

Hierarchical Clustering

How to Define Inter-Cluster Similarity

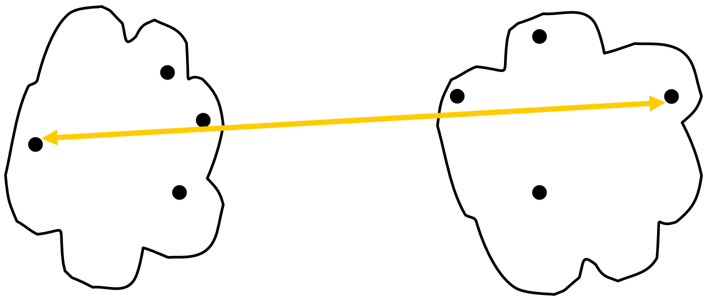


- MIN
- MAX
- Group Average

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity

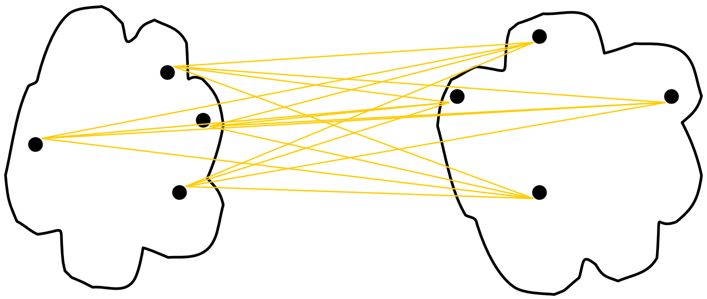


- MIN
- MAX
- Group Average

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

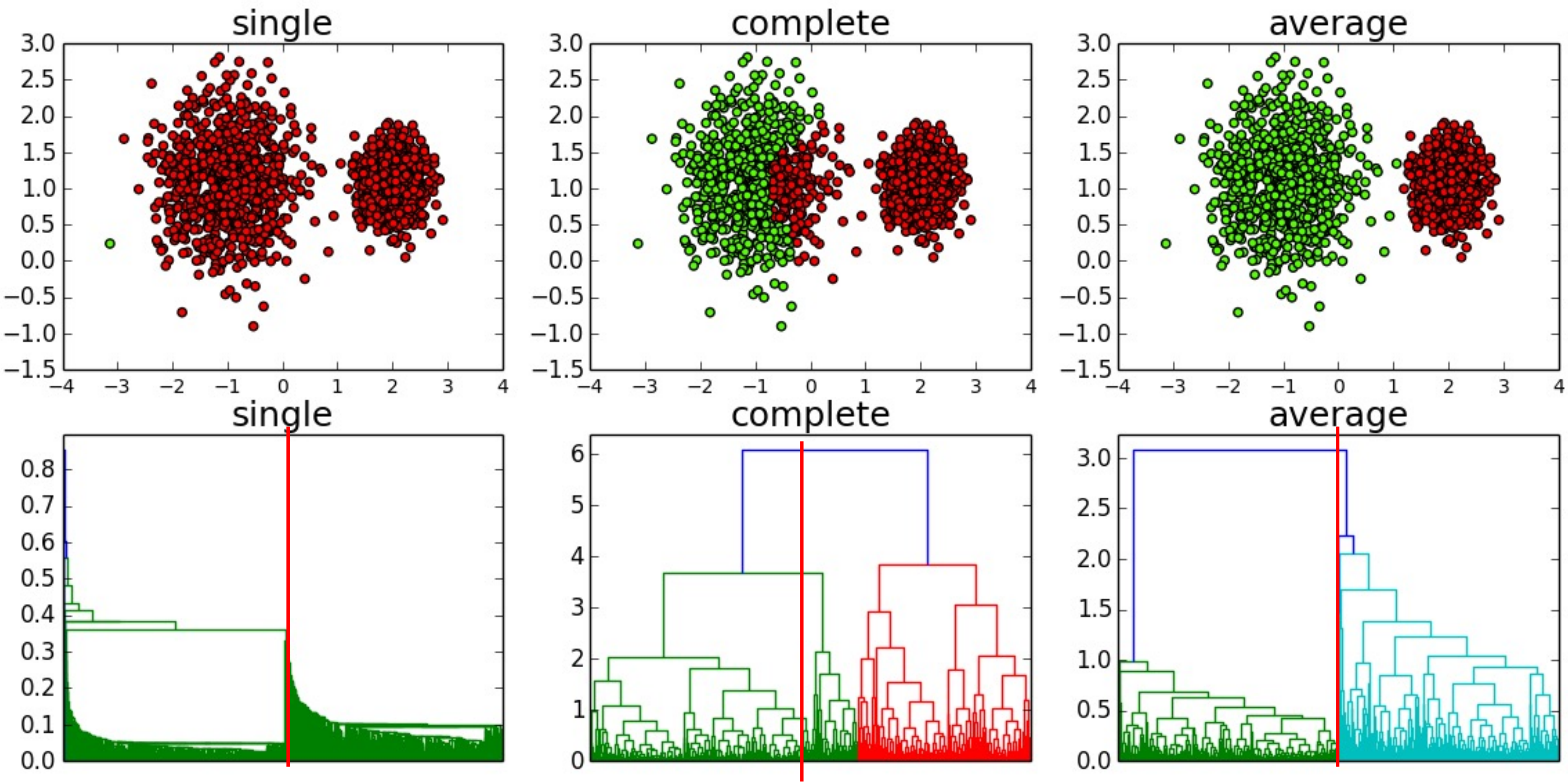
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



The dendrogram colors don’t match the cluster colors
(and specifically the “complete” one is backwards)

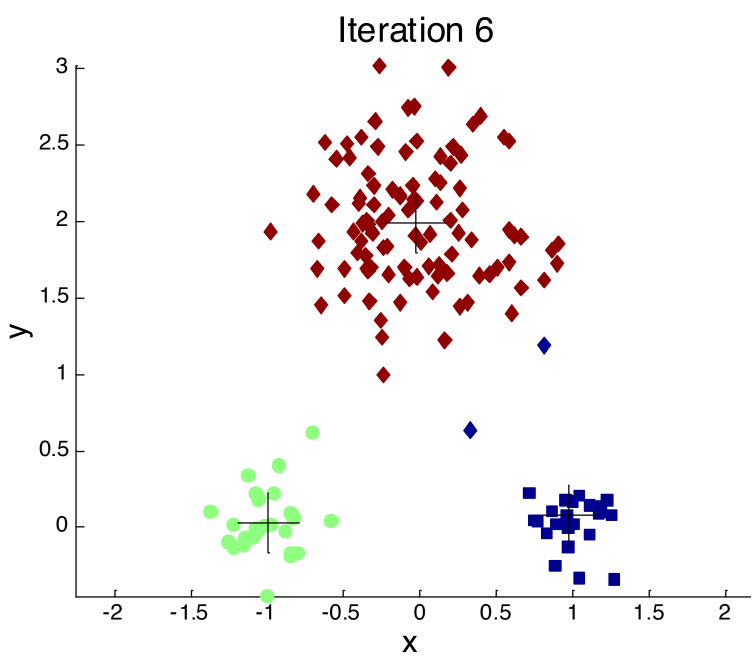
K-Means Clustering

K-means Clustering

- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, *K*, must be specified
- Basic algorithm is very simple

1: Select K points as the initial centroids.
2: **repeat**
3:     Form K clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.
5: **until** The centroids don't change

Example

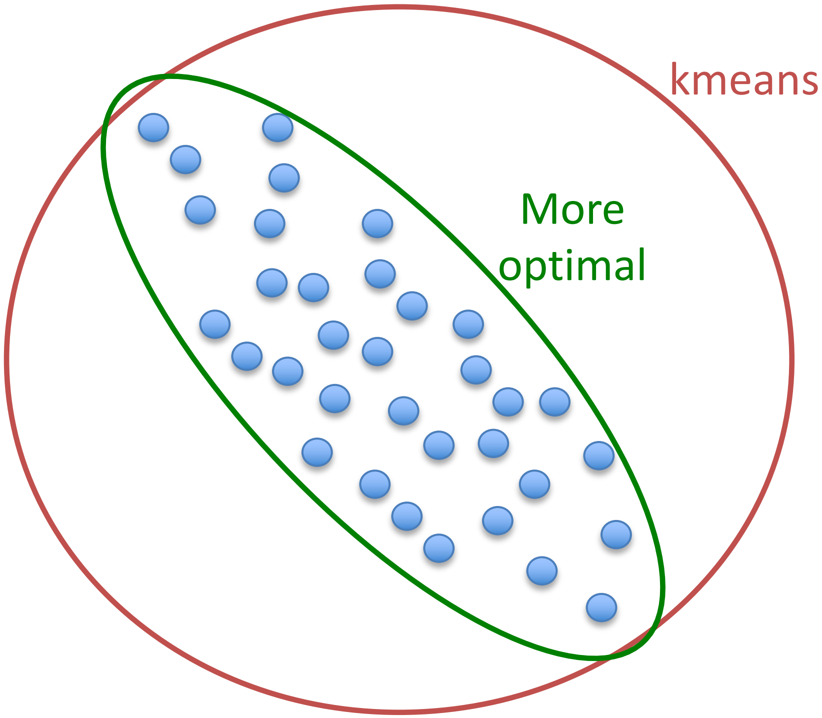


Super cool visualization

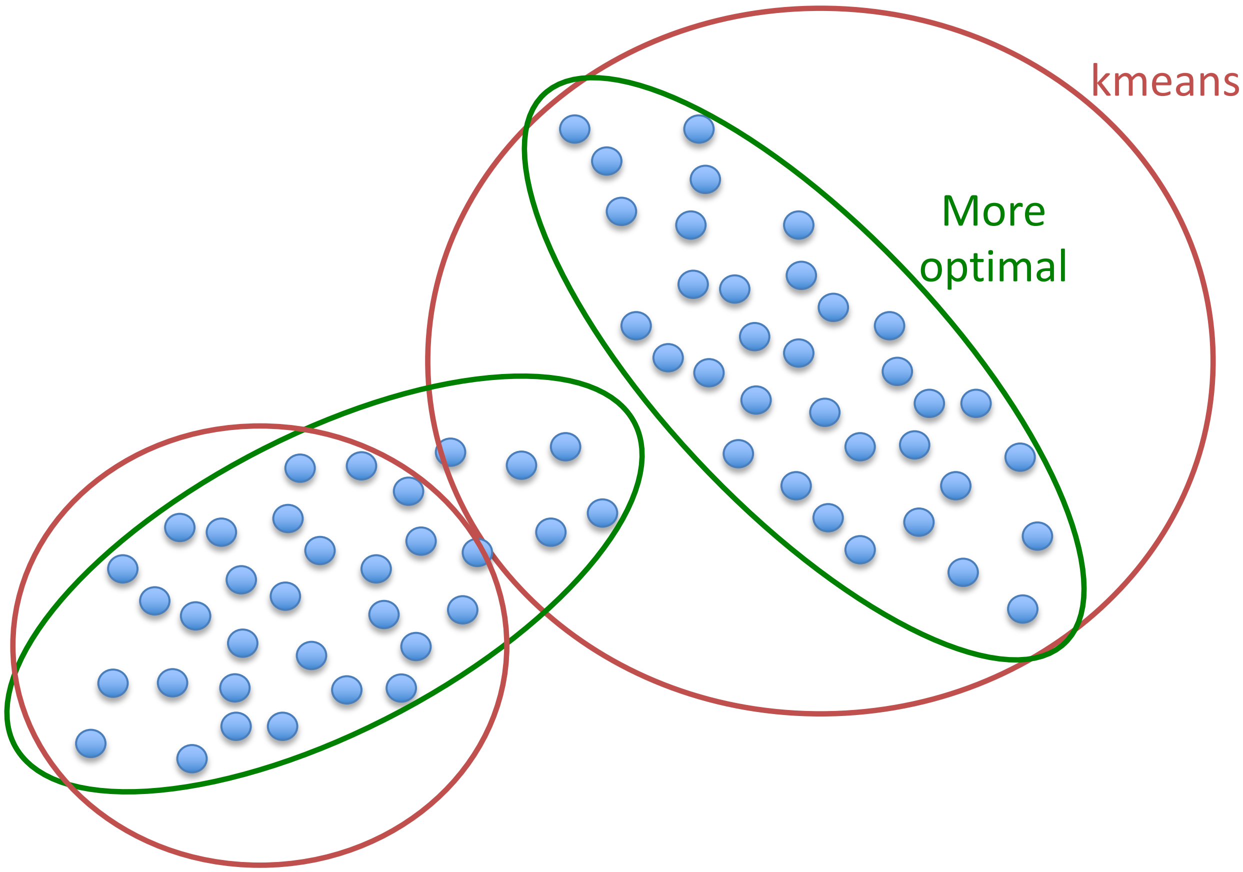
- https://educlust.dbvis.de/#

Disadvantages of Kmeans

- Assumes equal variance in all dims



Easy to see how kmeans could make a mistake here



Soft Clustering

- K means does a hard assignment of points to clusters.
  - Point 1 is in cluster A
- Might also like to know the probability of belonging to a cluster
  - Point 1 has probability 0.5 in cluster A
- Model each cluster with a probability distribution
  - Normal with params μ,Σ

Likelihood for unsupervised case

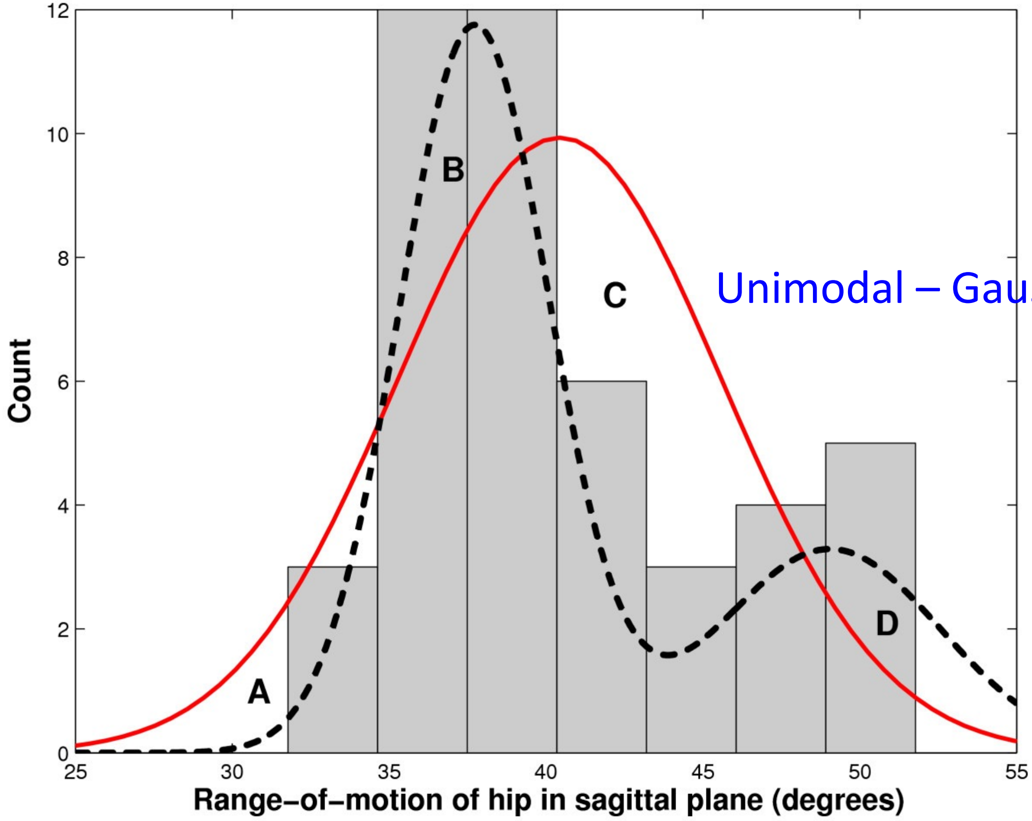
- In supervised learning we want to maximize
  - P(x,y) = p(x|y)p(y)
- This is *unsupervised* learning
  - We have no y!
- We wish to maximize p(x)
  - Find the params of p that maximize p(x)
  - Y can be latent, and we maximize the marginal dist.

$$p(x) = \sum_y p(x|y)p(y)$$

Mixture Model

- A density model $p(x)$ may be multi-modal.

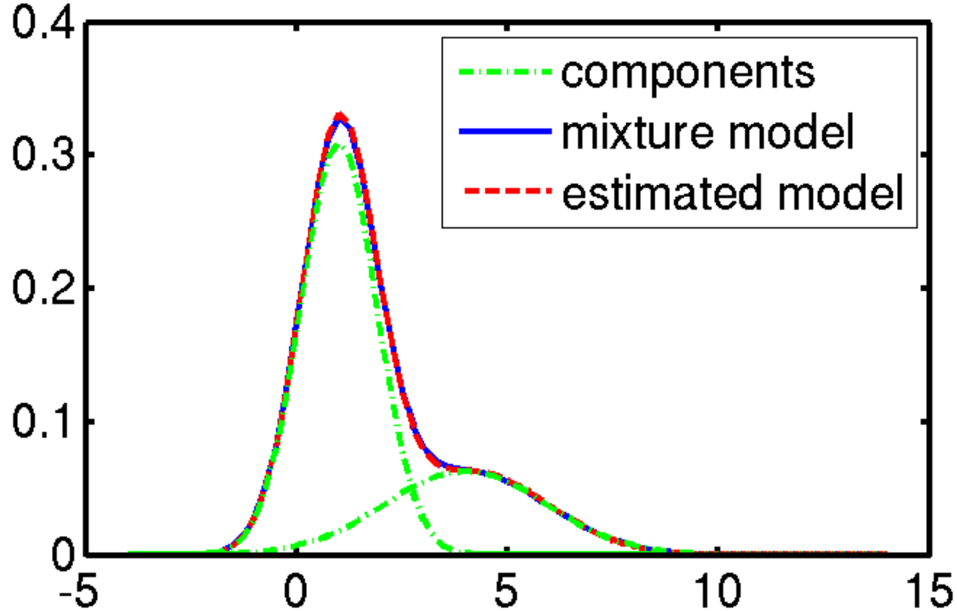
Multi-modal: how do we model this?



Unimodal – Gaussian(μ,Σ)

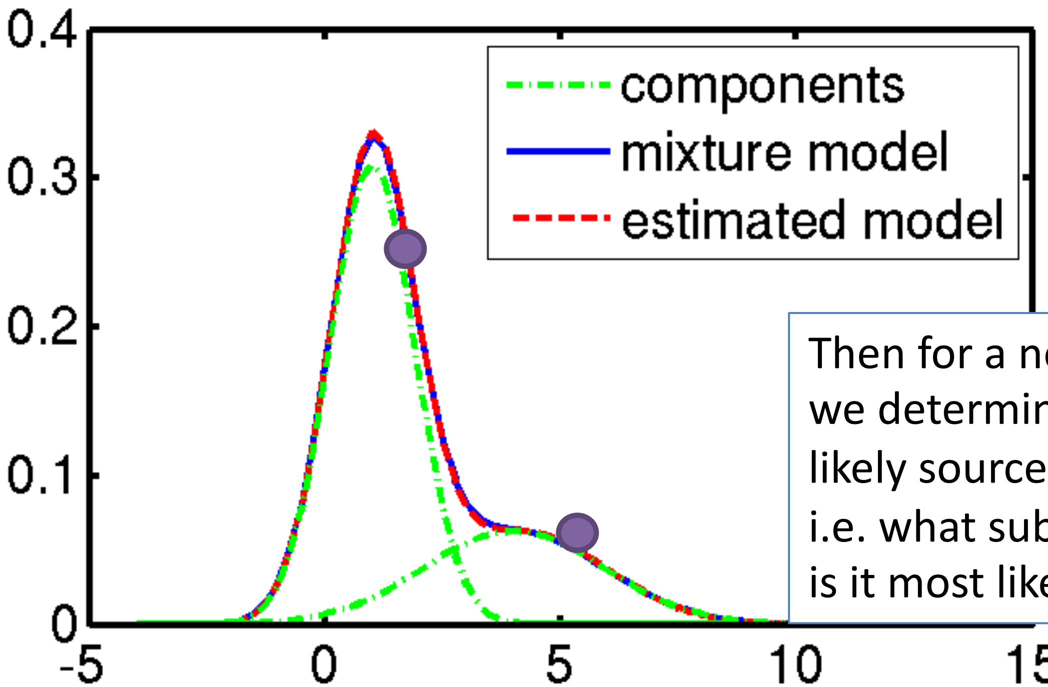
Mixture Model

- 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., hip dysplasia vs normal hips).



Mixture Model

- We observe the mixture (blue)
- Can we recover the components? (green)



Then for a new point, can we determine its most likely source?
i.e. what sub-population is it most likely from?

Likelihood for unsupervised case

- In supervised learning we want to maximize
  - P(x,y) = p(x|y)p(y)
- This is *unsupervised* learning
  - We have no y!
- We wish to maximize p(x)
  - We can marginalize over y

$$p(x) = \sum_y p(x|y)p(y)$$

Likelihood for unsupervised case

- In clustering, what does y represent?

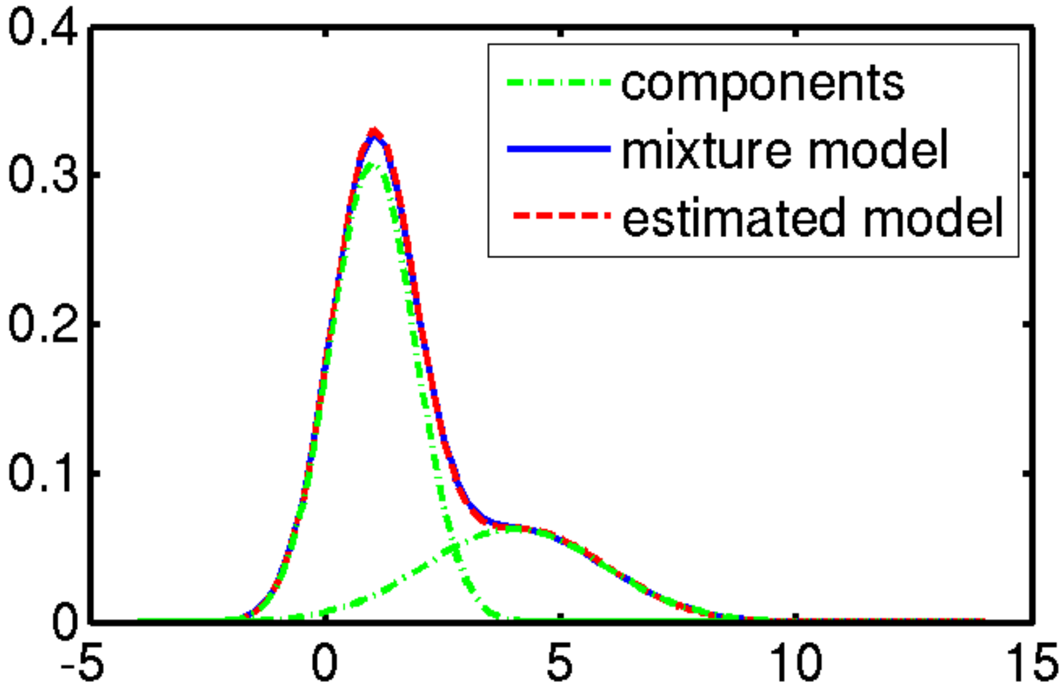
$$p(x) = \sum_{y} p(x|y)p(y)$$

- Y is the cluster!

$$p(x|\{y_1 \ldots y_K\}) = \sum_{j=1}^{K} p(x|y_j)p(y_j)$$

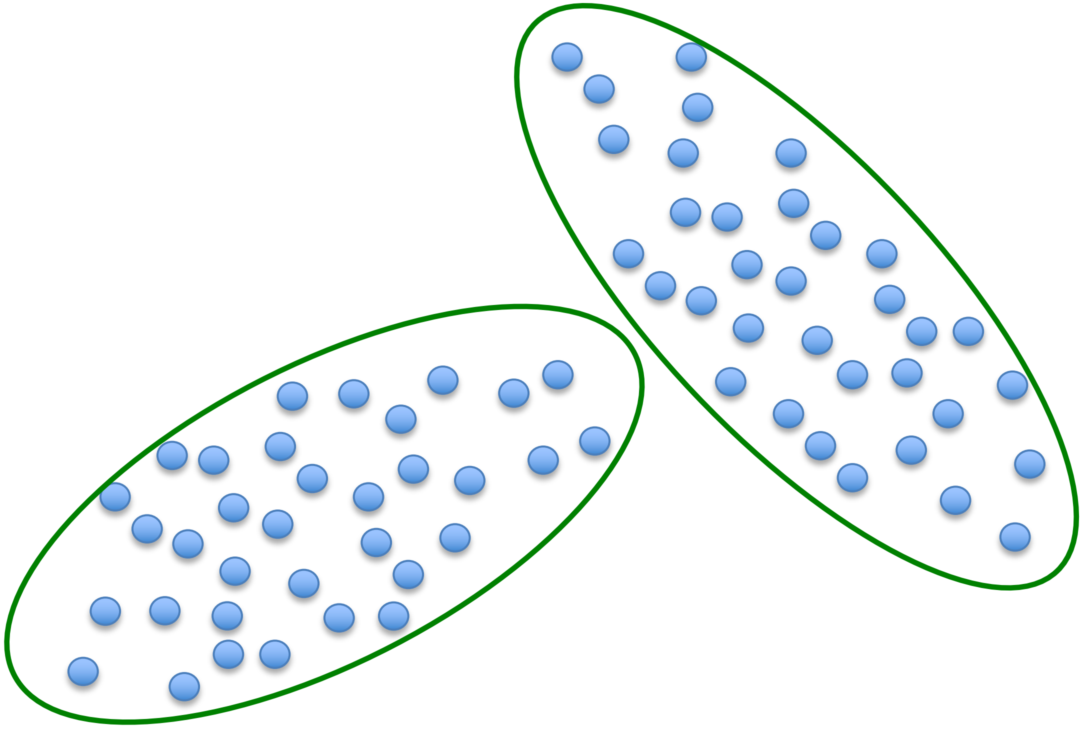
The Model

- Each cluster can be represented with some parameters
  - What could they be (1D case)?



The Model

- Each cluster can be represented with some parameters
  - What could they be here (2D case)?



The Model

- Probability of a point x given

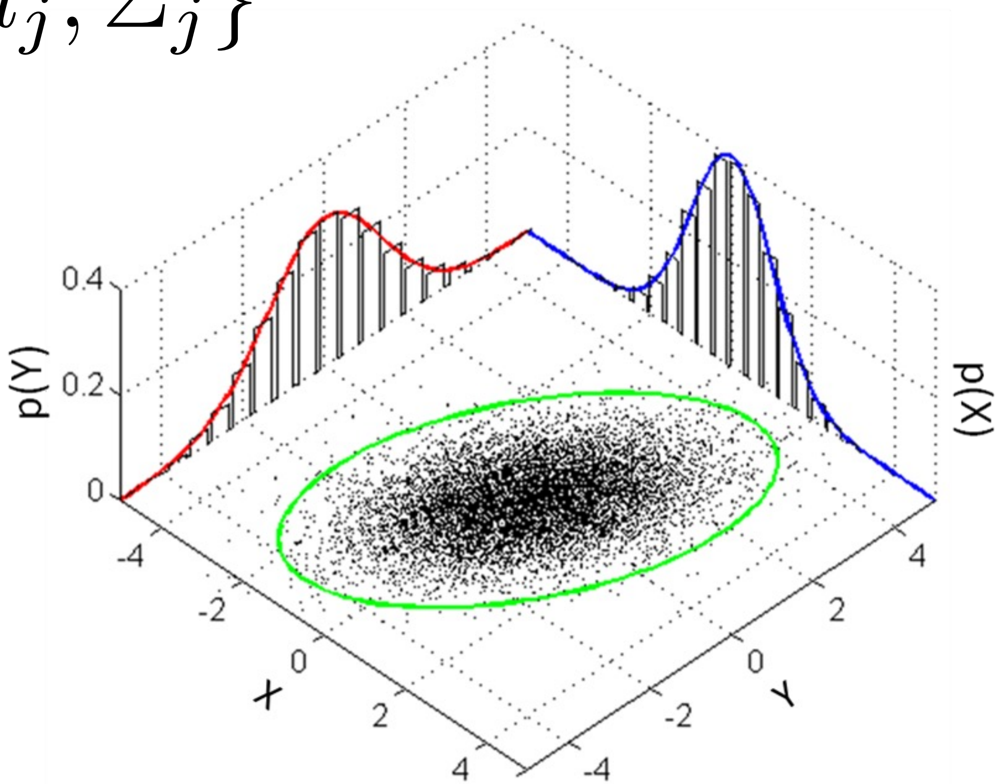
$$\Theta = \{\theta_1 \dots \theta_K\}$$

$$\theta_j = \{\mu_j, \Sigma_j\}$$

$$\mu_j \in \mathbb{R}^D$$

$$\Sigma_j \in \mathbb{R}^{D \times D}$$

```
%2-d Mean and covariance matrix
MeanVec = [0 0];
CovMatrix = [1 0.6; 0.6 2];
```



https://commons.wikimedia.org/wiki/File:MultivariateNormal.png

The Model

- Probability of a point x given

$$\Theta = \{\theta_1 \ldots \theta_K\}$$

$$\theta_j = \{\mu_j, \Sigma_j\}$$

$$p(x|\Theta) = \sum_{j=1}^{K} w_j p_j(x|\theta_j)$$

- w_j is probability *any* x belongs to cluster j
  - Note: does not depend on any of the other points

The Model

- Extend this to all points (likelihood of all data)

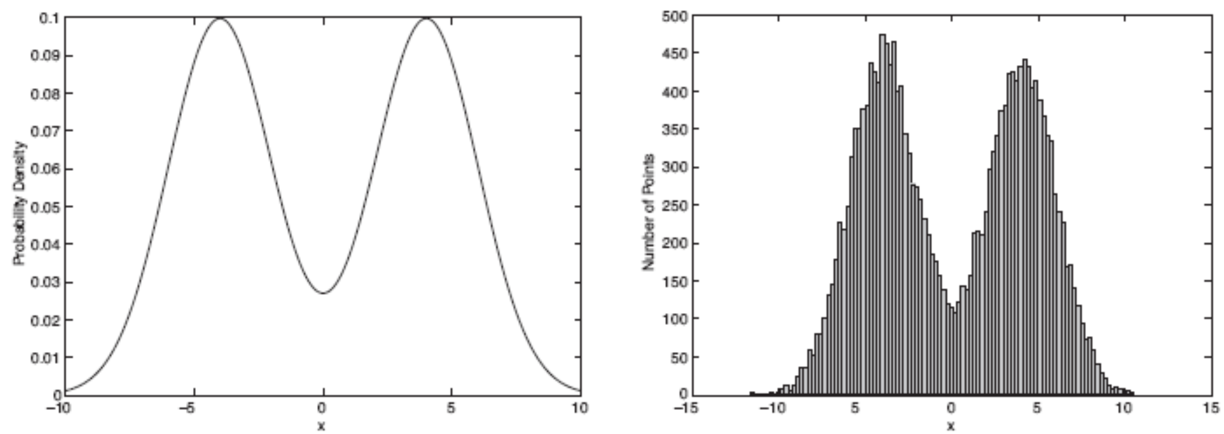
$$
\begin{aligned}
p(X|\Theta) &= \prod_{i=1}^{N} p(x_i|\Theta) \\
&= \prod_{i=1}^{N} \sum_{j=1}^{K} w_j \; p_j(x_i|\theta_j)
\end{aligned}
$$

Univariate Normal Case

$$
\begin{aligned}
p_j(x|\theta) &= p_j(x|\mu, \sigma) \\
&= \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\end{aligned}
$$

Example Mixture

- Two univariate gaussians with
  - μ_1=4, μ_2=-4,
  - σ_1 = 2, σ_1= 2,
  - w_1 = 0.5
  - w_2 = 0.5



(a) Probability density function for the mixture model.

(b) 20,000 points generated from the mixture model.

Figure 9.2. Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

How to Estimate the Params?

- Calculate the MLE!

- Turns out to be the sample mean and sample standard deviation

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2}$$

We're Missing Some Info

We just calculated:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2}$$

Reminder: Here's what we need to calculate:

$$p(x|\Theta) = \sum_{j=1}^{K} w_j p_j(x|\theta_j)$$

- Can't calculate the mean and std without knowing which m points belong to which cluster

We’re Missing Some Info

- Can’t calculate the mean and std without knowing which m points belong to which cluster
- But we can’t assign points to clusters without knowing the mean and std of the clusters
- EM handles this circularity

EM Algorithm

- Select initial set of parameters
  - i.e. Set μ and σ randomly, set all w = 1/K
- Repeat:
  - E-step: for each item x, calculate the probability that it belongs to each distribution p(dist j | x, Θ)
  - M-step: given probs from e-step, calculate new estimates of params that maximize the expected likelihood
- Until the params don’t change too much or likelihood doesn’t change too much

- To the derivation…

E-step (example with K=2 clusters)

- Find probability for belonging to each cluster
  - e.g. with two clusters:

$$p(dist\ j|x_i,\theta) = \frac{w_j\ p(x_i|\theta_j)}{w_1\ p(x_i|\theta_1) + w_2\ p(x_i|\theta_2)}$$

- (by Bayes rule)

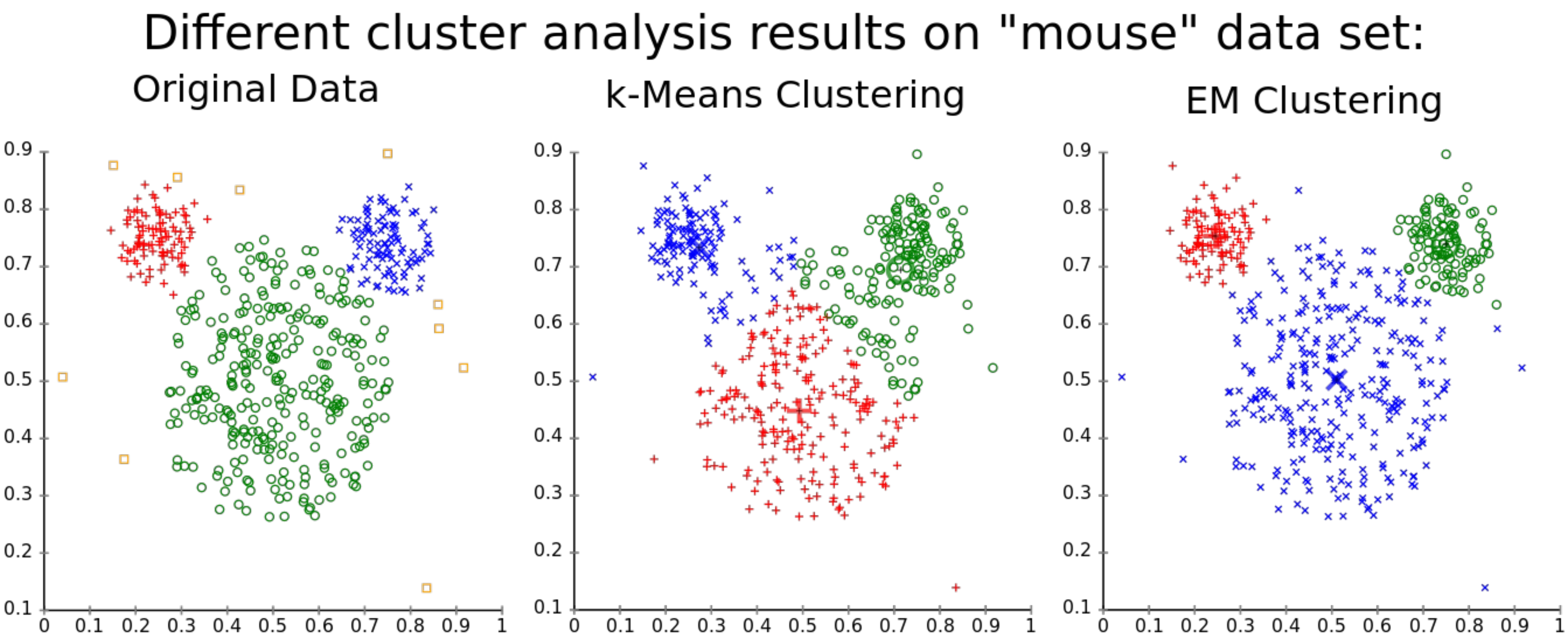
M-step

$$w_j = \frac{1}{N}\sum_{i=1}^{N} p(dist\ j|x_i, \theta)$$

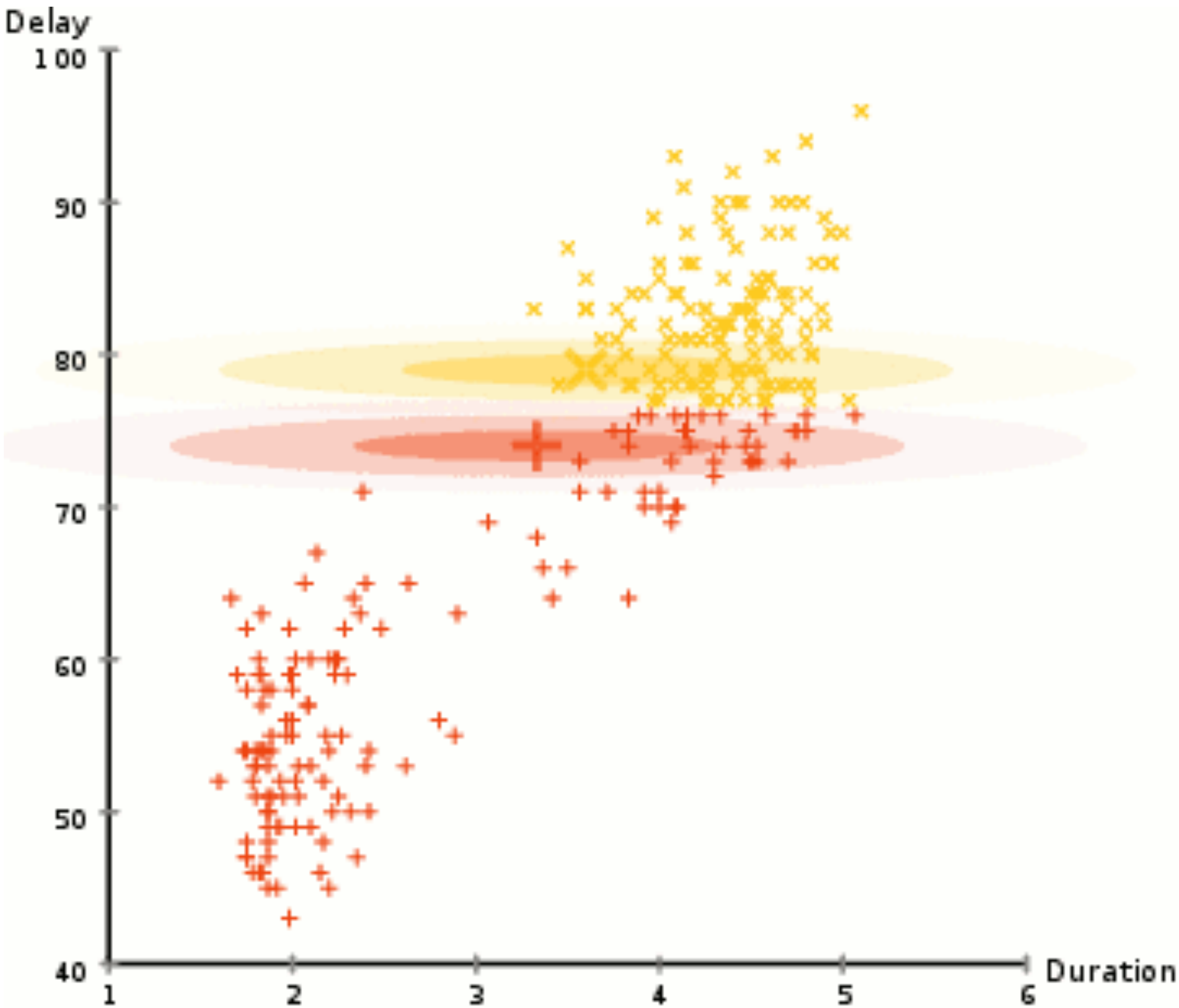
$$\mu_j = \frac{\sum_{i=1}^{N} p(dist\ j|x_i, \theta)x_i}{\sum_{i=1}^{N} p(dist\ j|x_i, \theta)}$$

$$\sigma_j = \frac{\sum_{i=1}^{N} p(dist\ j|x_i, \theta)(x_i - \mu_j)^2}{\sum_{i=1}^{N} p(dist\ j|x_i, \theta)}$$

K Means vs EM



from https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm



from https://en.wikipedia.org/wiki/Expectation-maximization_algorithm

Differences in Density

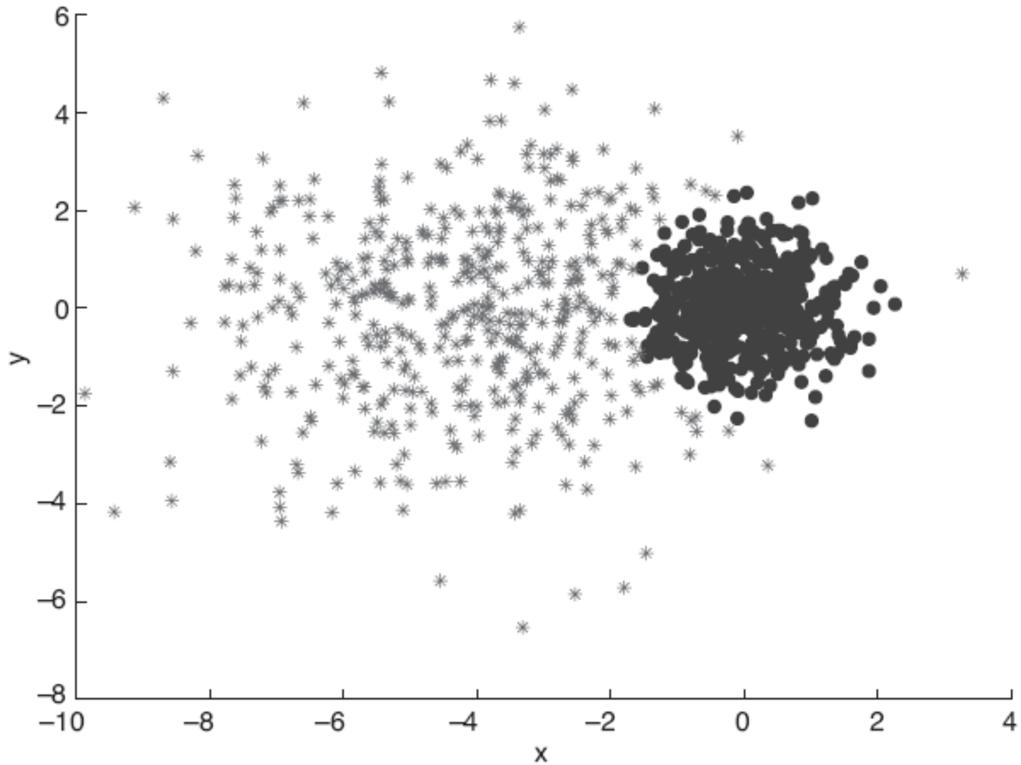
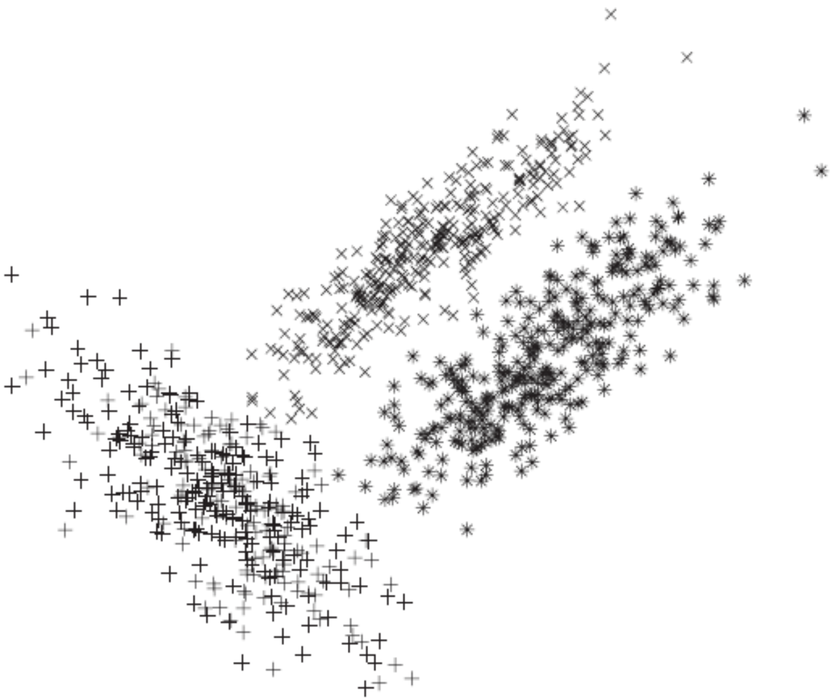


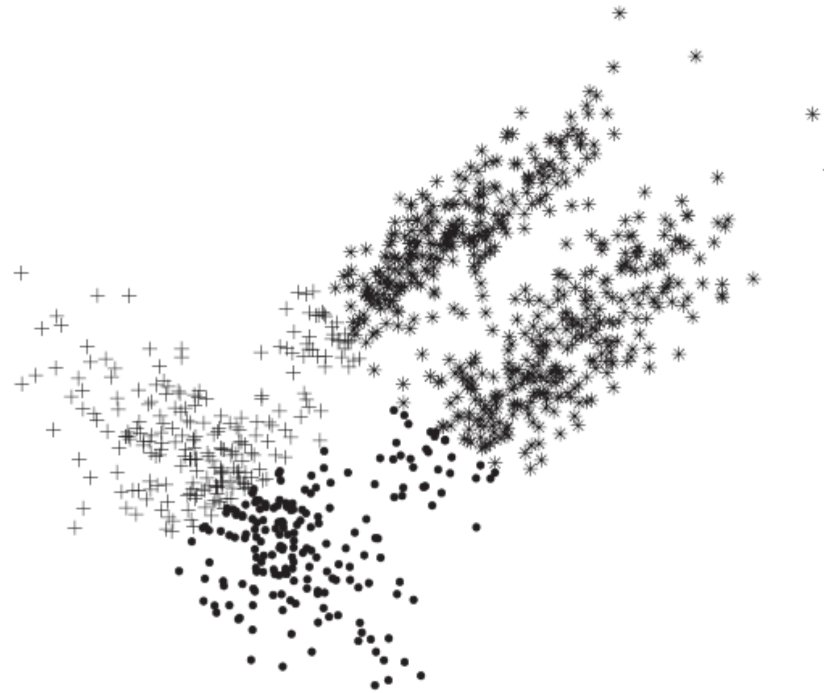
Figure 9.5. EM clustering of a two-dimensional point set with two clusters of differing density.

“Introduction to Data Mining” Tan, Steinbach and Kumar

Non-spherical data



(a) Clusters produced by mixture model clustering.



(b) Clusters produced by K-means clustering.

“Introduction to Data Mining” Tan, Steinbach and Kumar

Moving to higher D

- In higher D:

$$\mu_j \in \mathbb{R}^D$$

$$\Sigma_j \in \mathbb{R}^{D \times D}$$

- When D becomes very large computing the full DxD cov matrix is expensive
- Let’s look at this coding example…

Example code

- https://colab.research.google.com/drive/1dbZ24FZgaMb7YDY4JIQNJaiRVqf7cJUQ?usp=sharing

Other resources

- Nice video with example calculations
  - https://www.youtube.com/watch?v=XLKoTqGao7U