Mixture Models

By Dr. Shahira Azab

Mixture models in 1d

- Hard Clustering : Element either belongs to cluster or it does not
- soft clustering: clusters may overlap
- Mixture models
  - each cluster: a generative model (Gaussian or multinomial)
  - parameters (e.g. mean/covariance are unknown)
- Expectation Maximization (EM) algorithm
  - Algorithm discover all parameters for the K "sources".

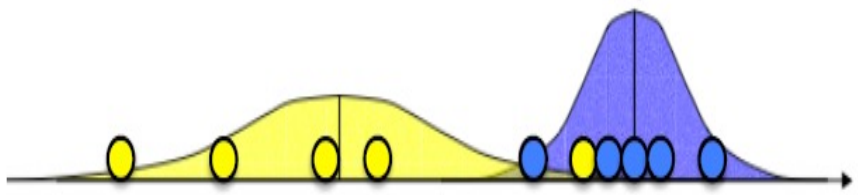
https://www.youtube.com/watch?v=-mYQ-nbykCo&list=PLBv09BD7ez_7beI0_fuE96lSbsr_8K8YD&index=2

Mixture models in 1-d

- Observations $x_1, x_2, \ldots\ldots x_2$
  - K=2 Gaussians with unknown μ,σ^2
  - estimation trivial if we know the source of each observation.
  - What if we don't know the source?
    - If we knew parameters of the Gaussians (μ, σ^2), we can guess whether point is more likely to be in class a or b.

$$\mu_b = \frac{x_1 + x_2 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_b)^2 + \ldots + (x_n - \mu_b)^2}{n_b}$$



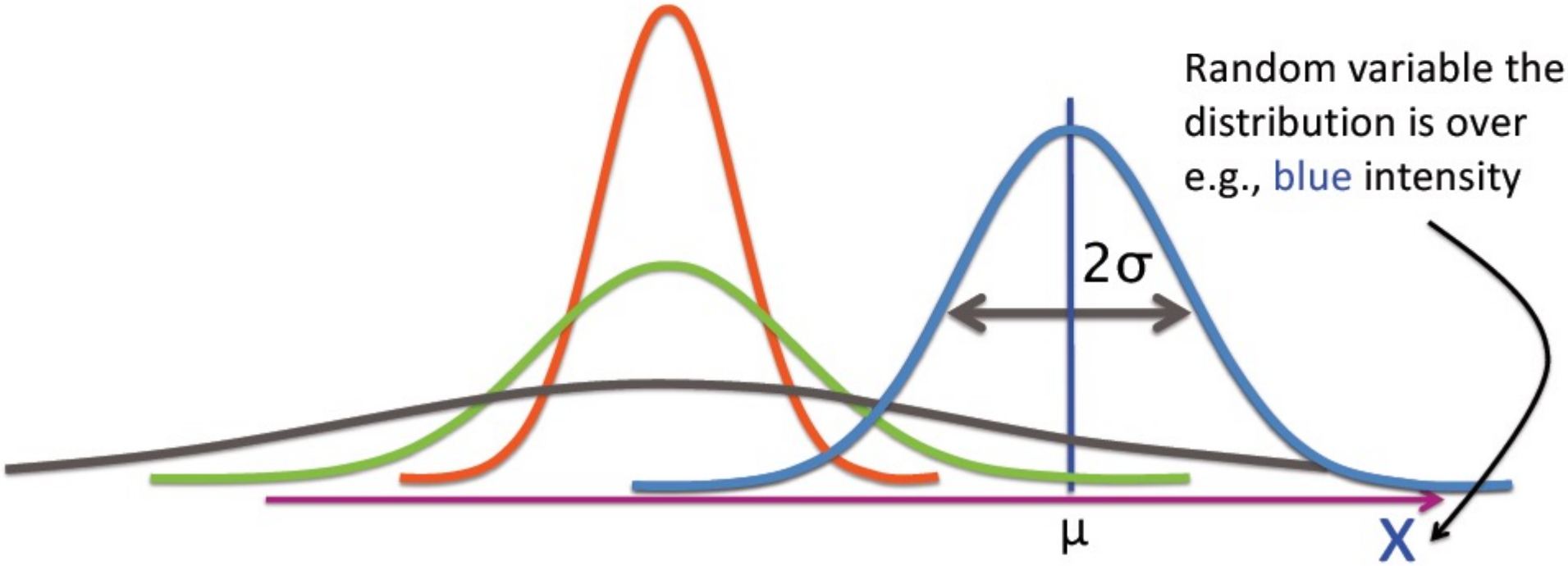
$$P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left\{ -\frac{(x_i - \mu_b)^2}{2\sigma_b^2} \right\}$$

https://www.youtube.com/watch?v=-mYQ-nbykCo&list=PLBv09BD7ez_7beI0_fuE96lSbsr_8K8YD&index=2

1D Gaussians

Fully specified by mean μ and variance σ^2 (or st. dev. σ)



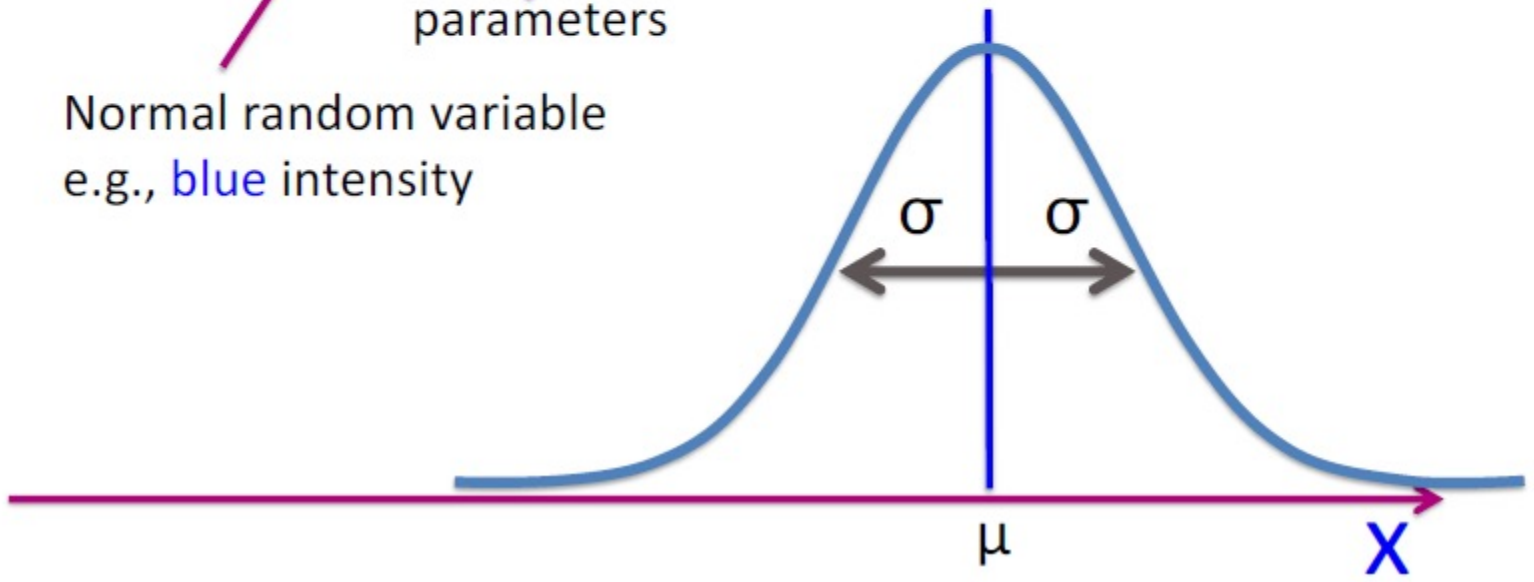
https://www.coursera.org/learn/ml-clustering-and-retrieval/

Notation a 1D Gaussian distribution

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

parameters

Normal random variable
e.g., blue intensity



https://www.coursera.org/learn/ml-clustering-and-retrieval/

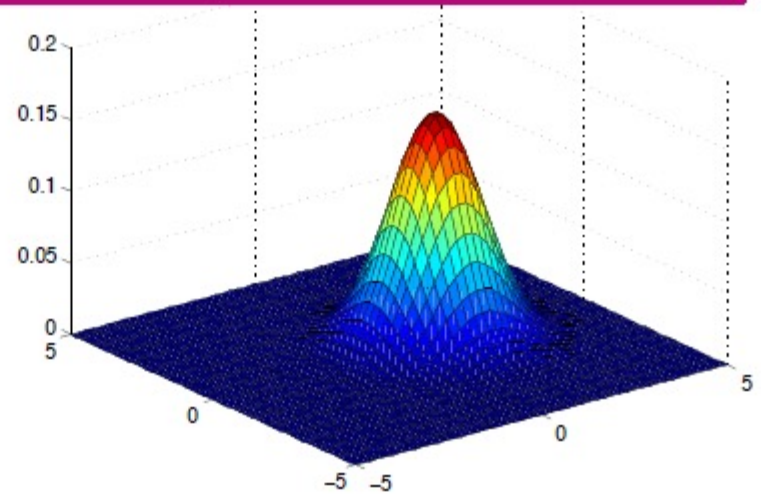
Multivariate Gaussian density

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$

parameters

Random vector
e.g., [R, G, B] intensities



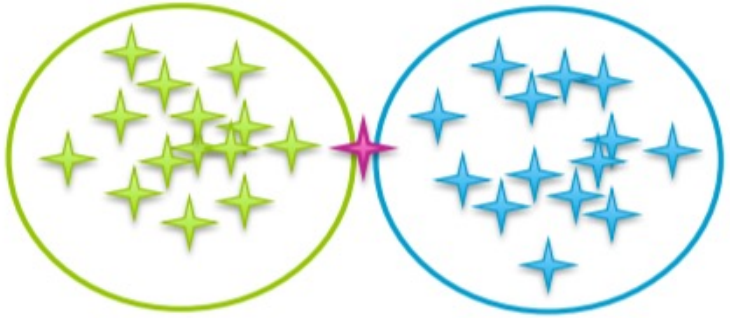
https://www.coursera.org/learn/ml-clustering-and-retrieval/

Part 1:
What if we knew the cluster parameters $\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}$?

Responsibilities in pictures

Green cluster takes more responsibility

Blue cluster takes more responsibility

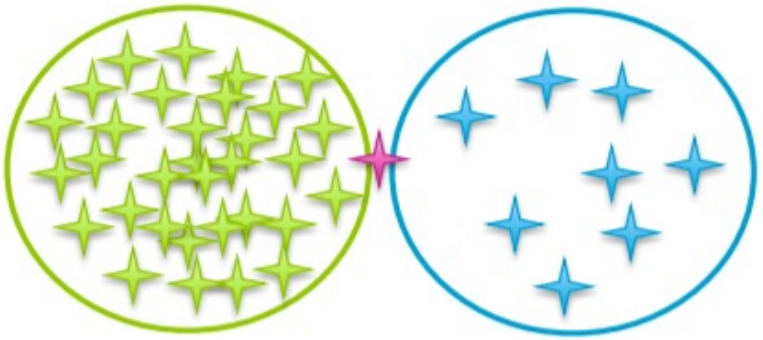


Uncertain... split responsibility

https://www.coursera.org/learn/ml-clustering-and-retrieval/

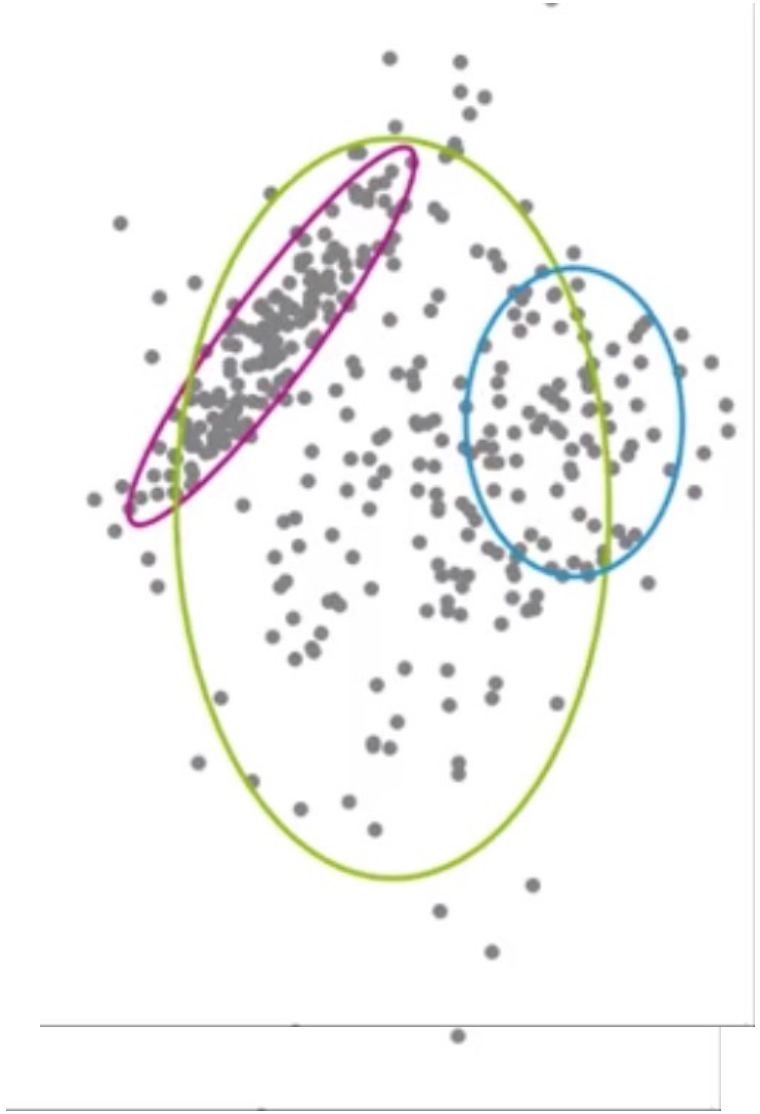
Responsibilities in pictures

Need to weight by cluster probabilities,
not just cluster shapes



Still uncertain,
but green cluster seems
more probable...
takes more responsibility

https://www.coursera.org/learn/ml-clustering-and-retrieval/



Responsibility cluster k takes for observation i

$$r_{ik} = p(z_i = k \mid \{\pi_j, \mu_j, \Sigma_j\}_{j=1}^{K}, x_i)$$

probability of assignment to cluster k

given model parameters and observed value

Responsibility cluster k takes for observation i

$$r_{ik} = \frac{\pi_k \; N(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)}$$

Normalized over all possible cluster assignments

https://www.coursera.org/learn/ml-clustering-and-retrieval/

An application of Bayes' rule

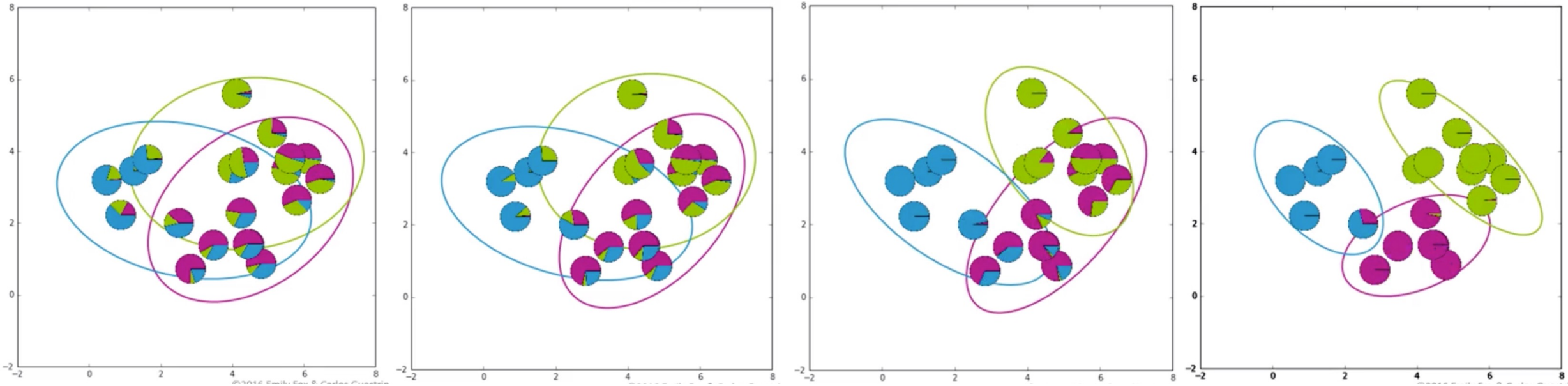
$$r_{ik} = p(A|B,\text{params})$$

$$= \frac{p(A|\text{params})p(B|A,\text{params})}{\sum_C p(C|\text{params})p(B|C,\text{params})}$$

$$= \frac{p(A|\text{params})p(B|A,\text{params})}{p(B|\text{params})}$$

Expectation Maximization (EM)

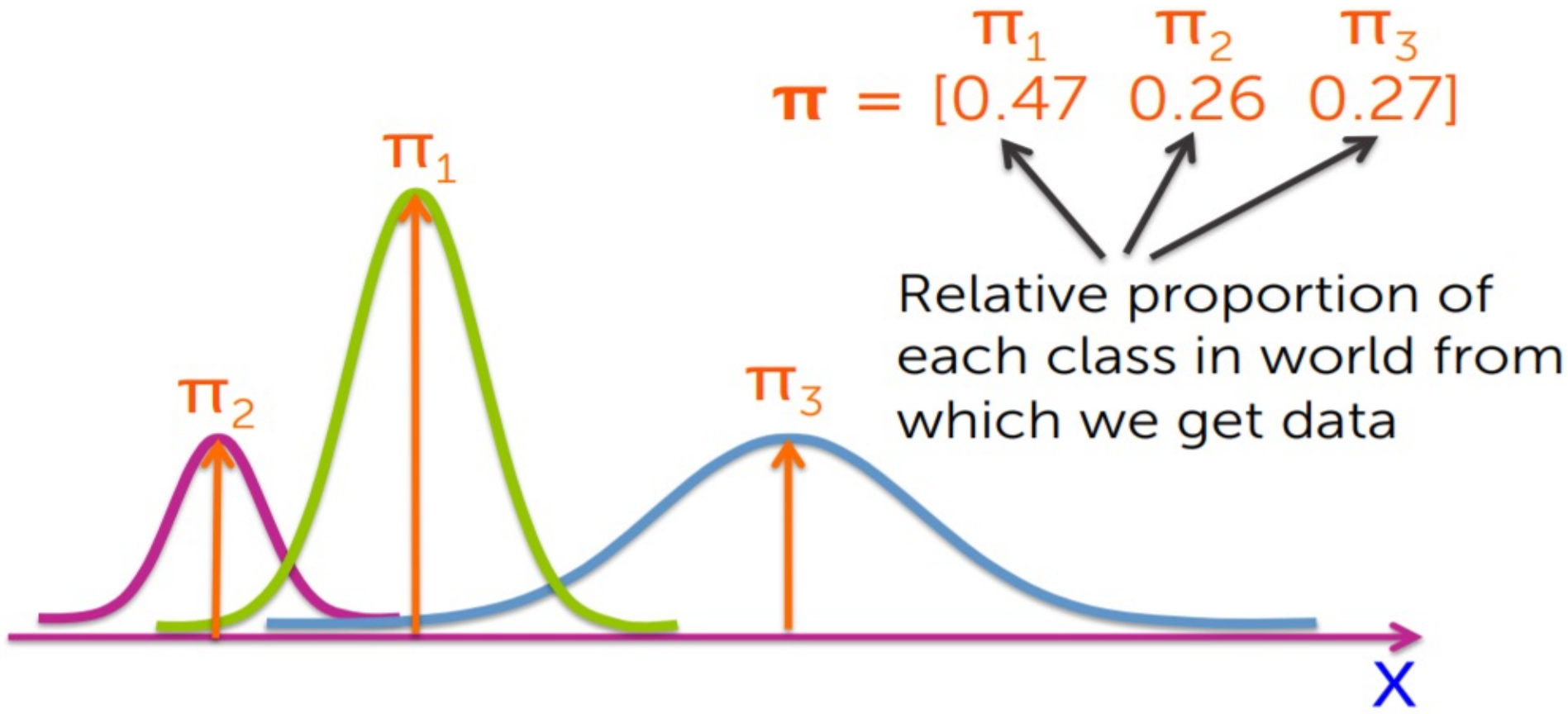
- EM algorithm
  - start with two randomly placed Gaussians $(\mu_a, {\sigma_a}^2), (\mu_b, {\sigma_b}^2)$
  - E-step: — for each point: $P(bl\ x_i)$ = does it look like it came from b?
  - M-step adjust $(\mu_a, {\sigma_a}^2), (\mu_b, {\sigma_b}^2)$ to fit points assigned to them M-step
  - iterate until convergence



https://www.youtube.com/watch?v=-mYQ-nbykCo&list=PLBv09BD7ez_7beI0_fuE96lSbsr_8K8YD&index=2

Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component

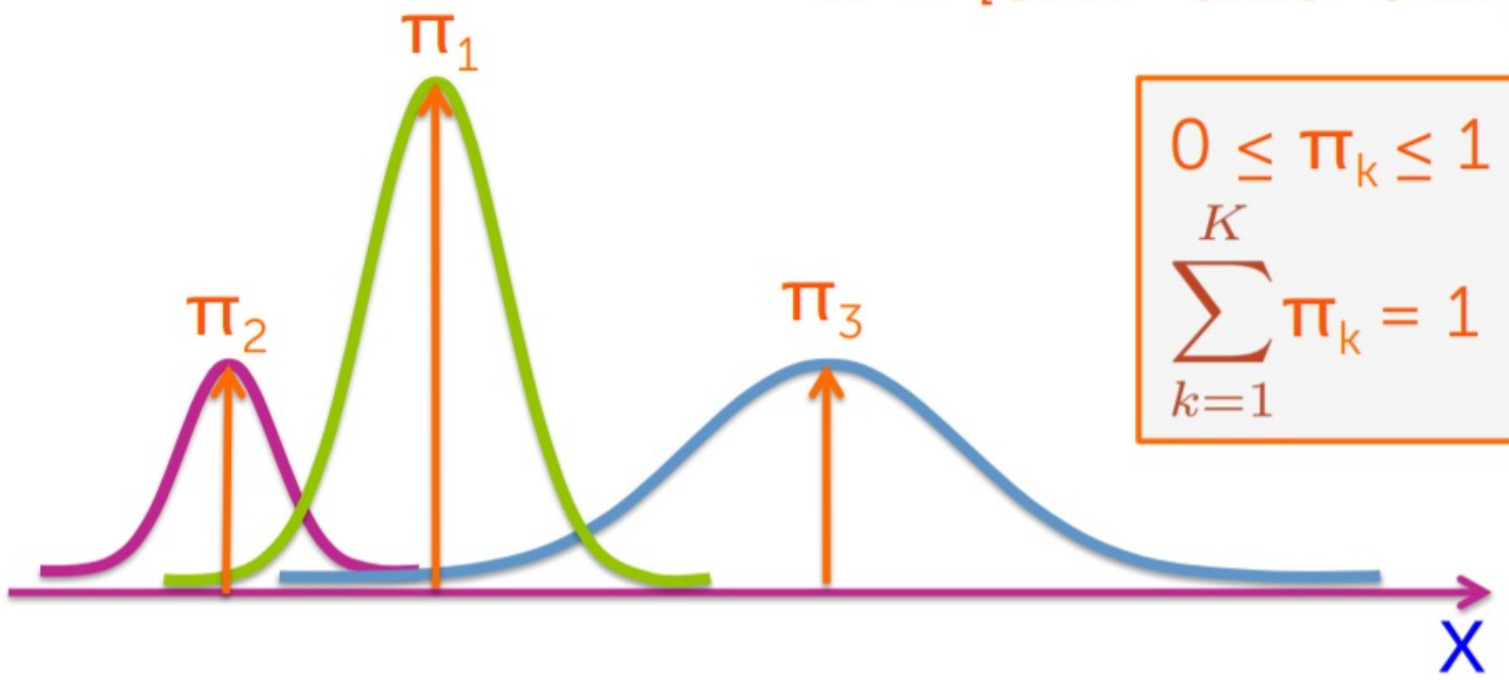


https://www.coursera.org/learn/ml-clustering-and-retrieval/

Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component

$$\boldsymbol{\pi} = [\pi_1 \quad \pi_2 \quad \pi_3] = [0.47 \quad 0.26 \quad 0.27]$$



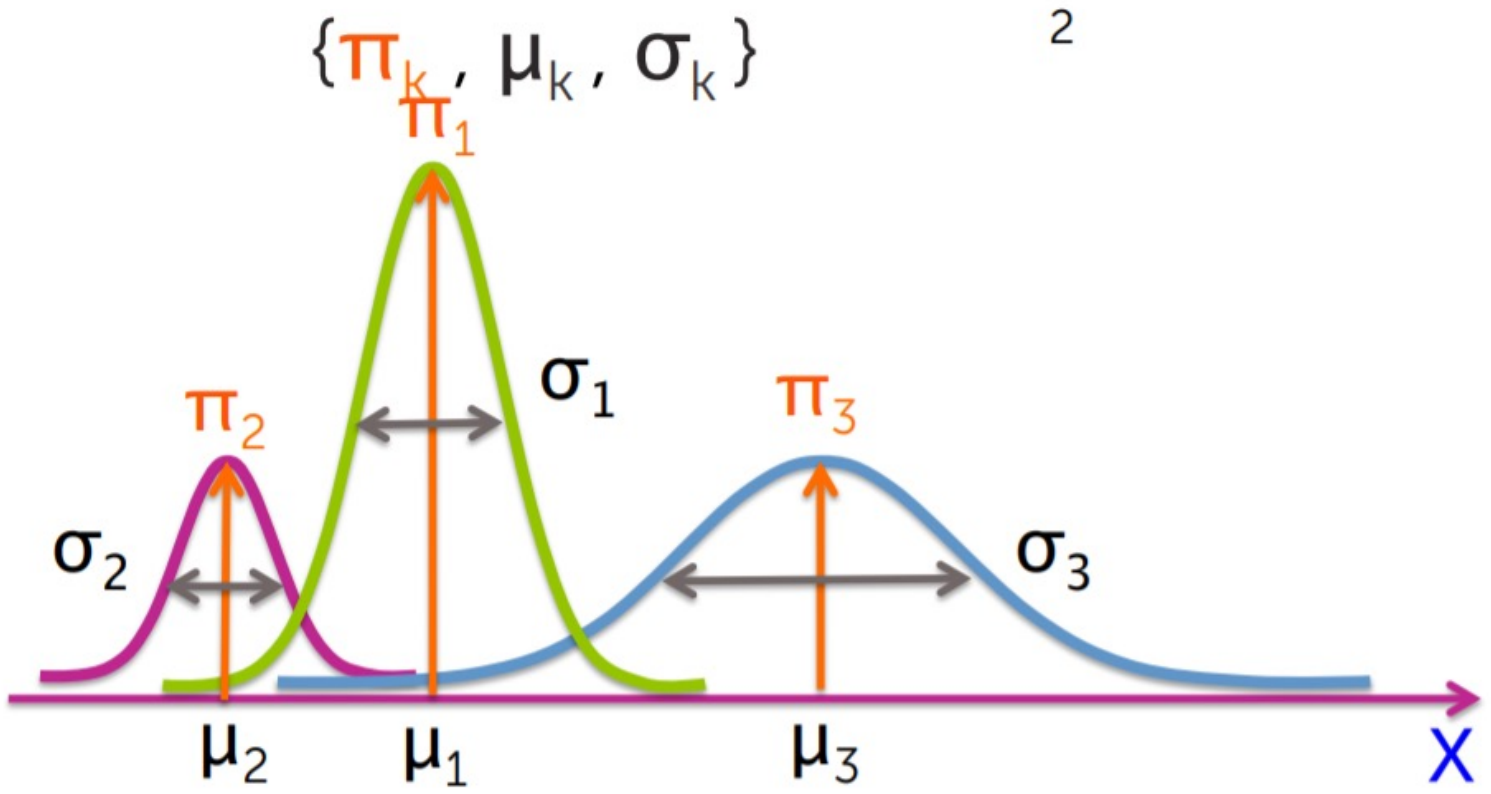
$$0 \leq \pi_k \leq 1$$

$$\sum_{k=1}^{K} \pi_k = 1$$

https://www.coursera.org/learn/ml-clustering-and-retrieval/

Mixture of Gaussians (1D)

Each mixture component represents a unique cluster specified by:

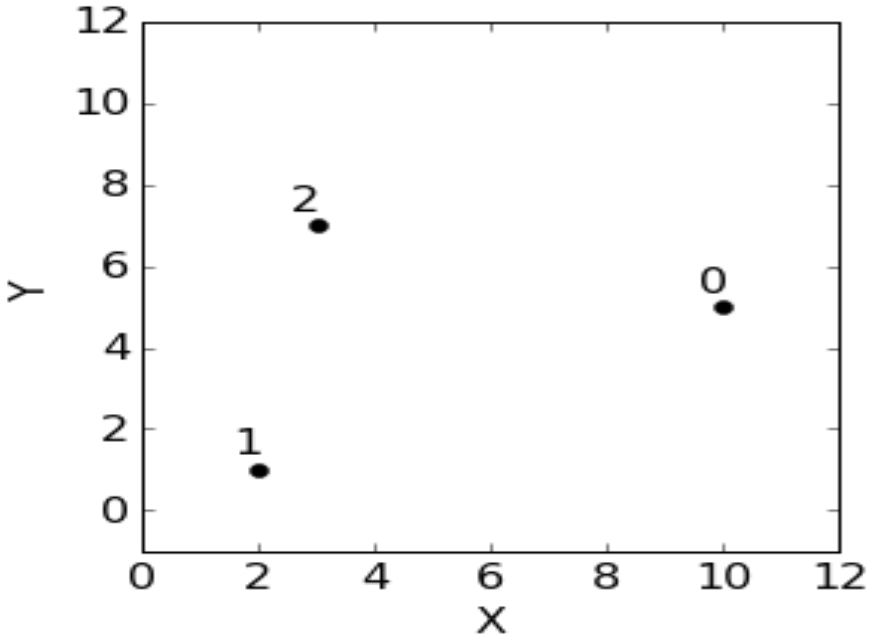


https://www.coursera.org/learn/ml-clustering-and-retrieval/

A worked-out example for K-means

- Reference: https://www.coursera.org/learn/ml-clustering-and-retrieval/
- When it comes to k-means, the notion of clusters is straightforward -- each point belongs to the cluster with the nearest centroid (mean). If someone gave us some data points and centroids, we can readily label each point for cluster.

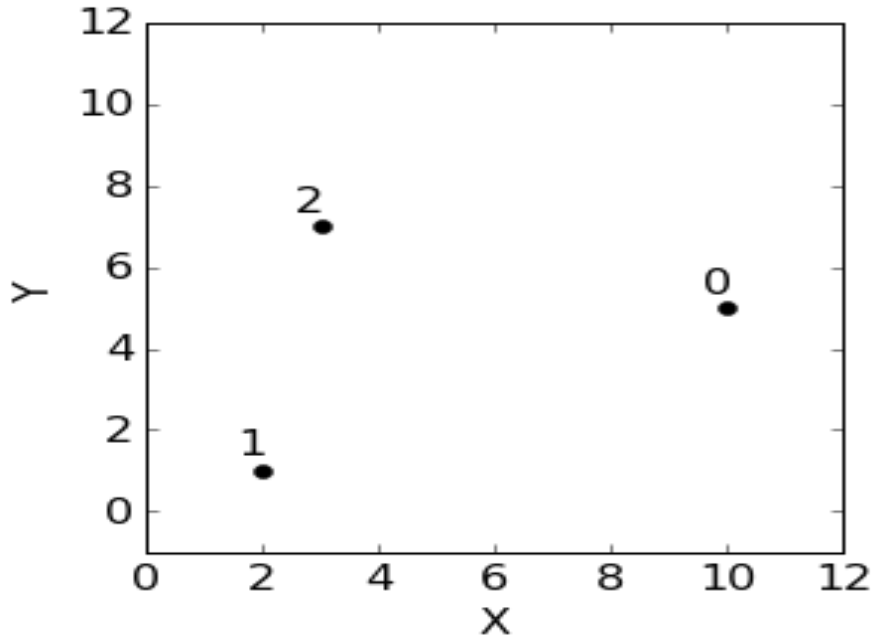
Dataset	X	Y
Data point 0	10	5
Data point 1	2	1
Data point 2	3	7



A worked-out example for K-means

- Reference: https://www.coursera.org/learn/ml-clustering-and-retrieval/
- When it comes to k-means, the notion of clusters is straightforward -- each point belongs to the cluster with the nearest centroid (mean). If someone gave us some data points and centroids, we can readily label each point for cluster.

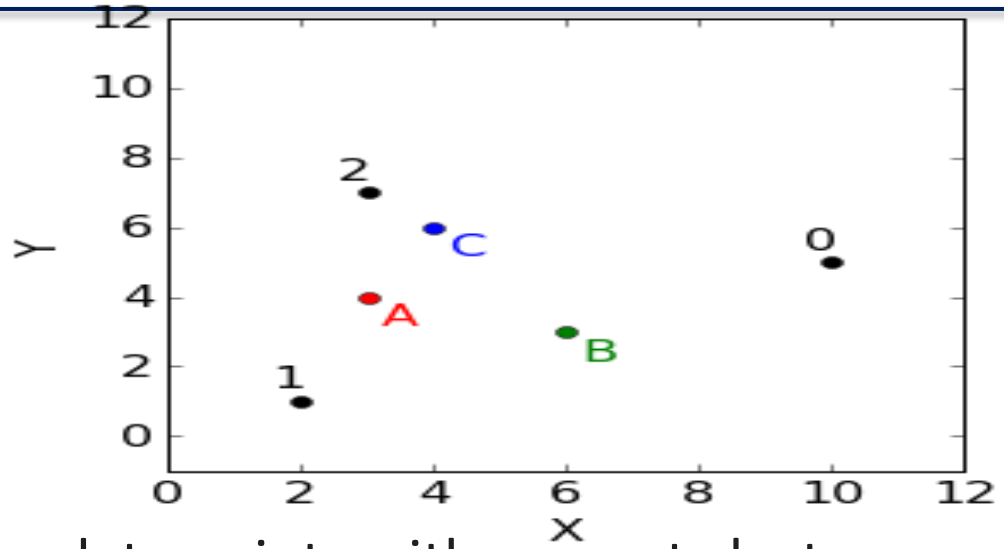
Dataset	X	Y
Data point 0	10	5
Data point 1	2	1
Data point 2	3	7



A worked-out example for K-means

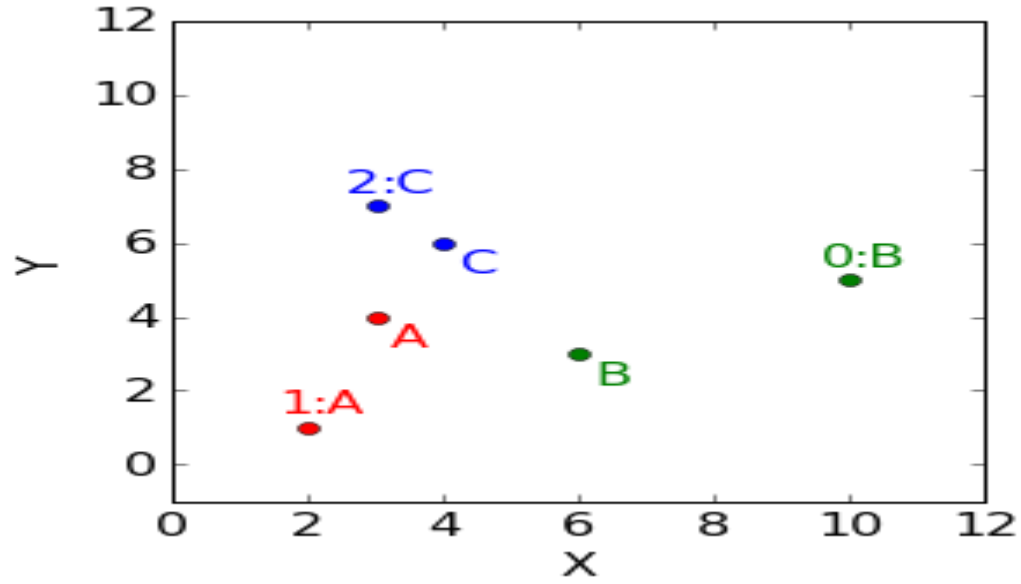
- Initial centroid

Centroids	X	Y
Cluster A	3	4
Cluster B	6	3
Cluster C	4	6



- Using the pairwise Euclidean distances, we label the data points with nearest clusters:

Pairwise distances	Centroid of Cluster A	Centroid of Cluster B	Centroid of Cluster C	**Cluster Assignment**
Data point 0	7.071	**4.472**	6.083	**Cluster B**
Data point 1	**3.162**	4.472	5.385	**Cluster A**
Data point 2	3.000	5.000	**1.414**	**Cluster C**



A worked-out example for K-means

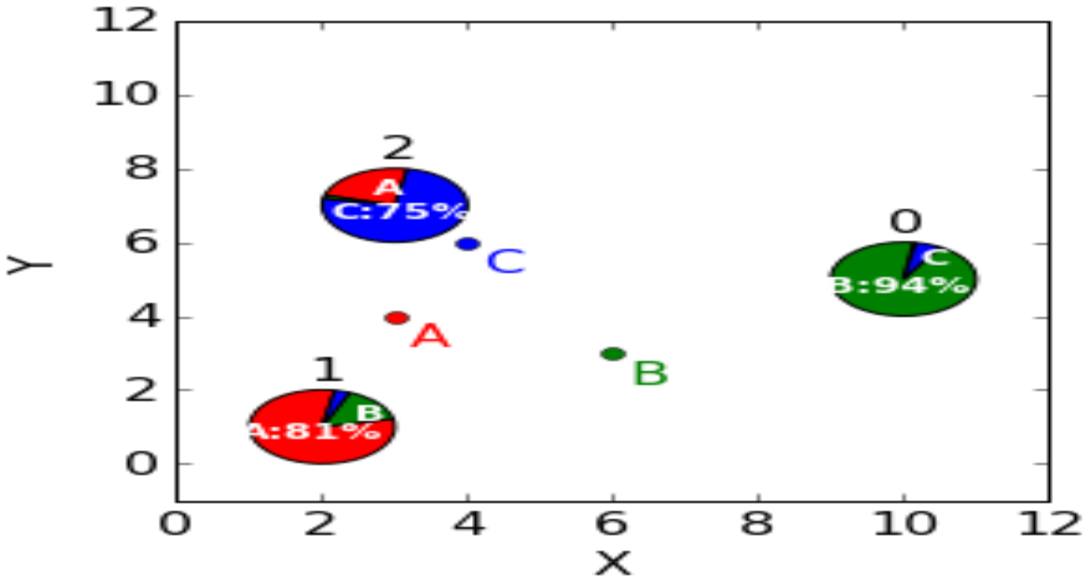
- But what does it mean for a data point to be assigned to a cluster? For instance, data point 1 is "assigned" to cluster A and none other. That is, the entire fraction (100%) of data point 1 is given to cluster A and zero fraction to cluster B and cluster C. Following the same reasoning, we can write out the fractions of all data points going to all clusters:

	Cluster A	Cluster B	Cluster C
Data point 0	0	**1**	0
Data point 1	**1**	0	0
Data point 2	0	0	**1**
Member counts	1	1	1

A worked-out example for EM

1) Cluster responsibilities : The cluster responsibility is the fraction of a data point being represented in a certain cluster

Responsibility matrix	Cluster A	Cluster B	Cluster C
Data point 0	0.007	0.938	0.055
Data point 1	0.812	0.154	0.034
Data point 2	0.234	0.016	0.750
Soft counts	1.053	1.108	0.839



The first row tells us that data point 0 belongs to cluster B with 93.8% probability, an overwhelming certainty. (We shall use percentages instead of fractions here.) The next row expresses some uncertainty regarding the membership of data point 1, with 81.2% for cluster A and 15.4% for cluster B. The last row does so as well: 75.0% for cluster C and 23.4% for cluster A. These fractions are known as cluster responsibilities. For example, the responsibility of cluster A for data point 0 is 0.7%.

Unlike in k-means, where each point was assigned to one single cluster, we now have all points represented in all clusters, to a varying degree. For instance, 93.8% portion of data point 0 was represented in cluster A and the remaining portion in other clusters.

A worked-out example for EM

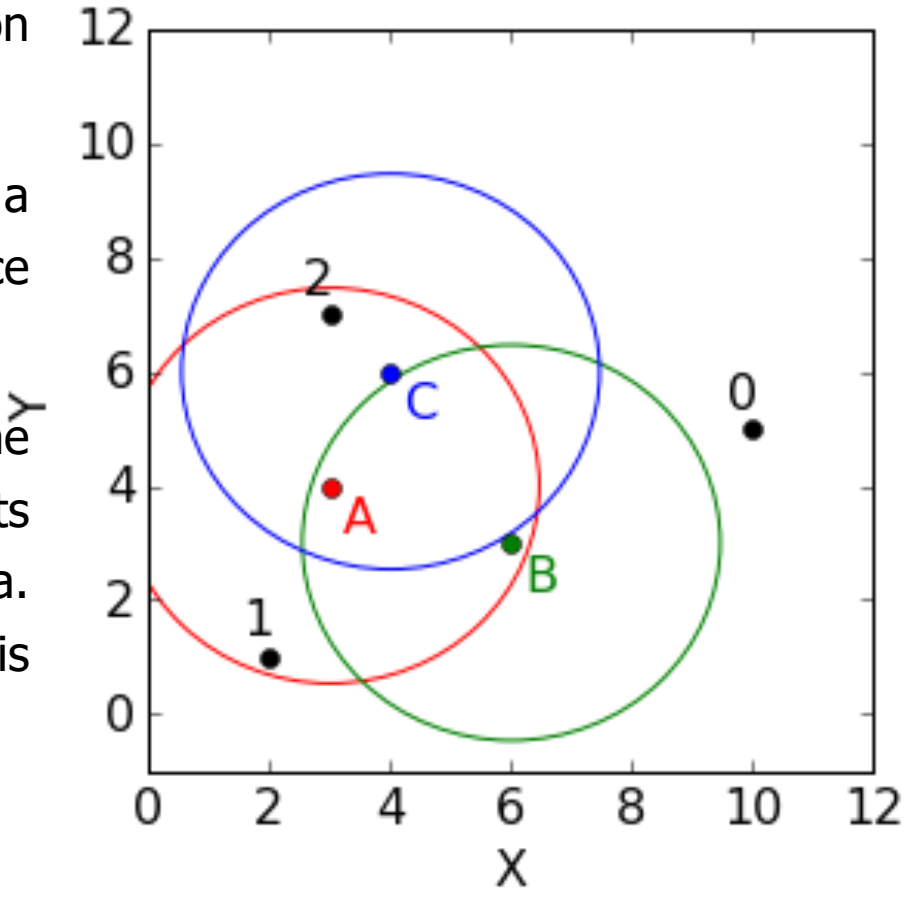
- Finally, as in the assignment matrix, the column and row sums have meanings, too:
  - Summing over a row always yields 1, as all fractions of a point must add up to a whole.
  - Summing over a column yields what is known as the "soft count" of the cluster. This quantifies the total of the fractional representations of all points in that particular cluster. It's a bit like counting the points within a cluster, except we're talking decimals like 1.053.
  - The soft counts for all clusters sums to the number of data points. In this example, 1.053+1.108+0.839=3.000.

A worked-out example for EM

- E-step: Compute cluster responsibilities, given cluster parameters

How do we come up with the cluster responsibilities?

- We look at how likely the data point is given the distribution of the cluster.
- Recall from the lectures that each cluster in EM consists of a cluster weight, a mean vector (centroid) and a covariance matrix.
- The mean defines the center of the cluster, and the covariance defines the spread. The cluster weight represents the relative representation of the cluster in the data. Summing all cluster weights yields 1. Finally, each cluster is modeled by a multivariate Gaussian distribution



A worked-out example for EM

- Each ellipse visualizes the covariance matrix. In this case, all three clusters have the diagonal covariance [[3,0],[0,3]]. Since the off-diagonal entries are zero, the ellipses shown above are in fact circles. Also, since there is no reason to believe that one cluster would be better represented than other, let's assign a cluster weight of 1/3 to all clusters.

- Note. The way we initialize covariances and weights may appear arbitrary. This is actually okay, just as initializing k-means with a random set of centroids is okay. We've just used an initial "estimate" of weights, means, and covariances. After computing the cluster responsibilities, we can update these parameters to arrive at a better estimate.

- Let's take data point 0 and ask ourselves: How likely is the data point in the distribution of cluster A? The measure we'll be using is the probability density function (PDF) of the underlying Gaussian distribution. In SciPy, you may compute this measure using scipy.stats.multivariate_normal.pdf. At coordinates (10,5), i.e. data point 0, the PDF of the Gaussian distribution for cluster A has value 1.275e-5

- `print multivariate_normal.pdf([10,5], mean=[3,4], cov=[[3,0],[0,3]])`

- `>>> 1.2751996780192l9e-05`

A worked-out example for EM

- We need to scale this number by the cluster weight, as the cluster weight denotes the importance of the cluster compared to other clusters.
- print 1/3.*multivariate_normal.pdf([10,5], mean=[3,4], cov=[[3,0],[0,3]])
- >>> 4.2506655934e-06
- The likelihood measure for cluster A at data point 0 is 4.251e-6. By itself, this number has no meaning -- we need to compute the same measure for clusters B and C and compare. Compute the PDF for cluster B and scale by the cluster weight:
- `print 1/3.*multivariate_normal.pdf([10,5], mean=[6,3], cov=[[3,0],[0,3]])`
- `>>> 0.000630854709005`
- print 1/3.*multivariate_normal.pdf([10,5], mean=[6,3], cov=[[3,0],[0,3]])
- >>> 0.000630854709005
- The likelihood measure for cluster B is 6.309e-4. Similarly, the likelihood measure for cluster C is 3.710e-5:
- `print 1/3.*multivariate_normal.pdf([10,5], mean=[4,6], cov=[[3,0],[0,3]])`
- `>>> 3.71046481027e-05`

A worked-out example for EM

- Clearly, cluster B has the highest likelihood measure. This makes sense because data point 0 is closer to the center of cluster B than others.
- The likelihood measure is all good, but it is typically a small number. Since we find it easier to reason with percentages than really tiny numbers, let's divide each value by the sum over all values, as follows:

	Cluster A	Cluster B	Cluster C
PDF of cluster at (10,5), multiplied by cluster weight	4.251e-6	6.309e-4	3.710e-5

Data point 0	Cluster A	Cluster B	Cluster C	Sum
Likelihood measure	4.251e-6	6.309e-4	3.710e-5	6.722e-4
Likelihood measure, divided by the sum	4.251e-6 / 6.722e-4 = **0.007**	6.309e-4 / 6.722e-4 = **0.938**	3.710e-5 / 6.722e-4 = **0.055**	

A worked-out example for EM

- the numbers in the last row are the cluster responsibilities for data point 0! Notice that, by dividing through by the sum, we ensured that cluster responsibilities add up to 1. Let's review what we did so far:

- Compute the value of the PDF of the Gaussian distribution at each data point. Use the mean vector and covariance matrix as parameters of the Gaussian distribution.
- Multiply the PDF value by the cluster weight. This is the likelihood measure.
- Normalize all likelihood measures.

A worked-out example for EM

Data point 1	Cluster A	Cluster B	Cluster C	Sum
Likelihood measure	3.340e-3	6.309e-4	1.408e-4	4.112e-3
Likelihood measure, divided by the sum	3.340e-3 / 4.112e-3 = **0.812**	6.309e-4 / 4.112e-3 = **0.154**	1.408e-4 / 4.112e-3 = **0.034**	

Data point 2	Cluster A	Cluster B	Cluster C	Sum
Likelihood measure	3.946e-3	2.742e-4	1.267e-2	1.689e-2
Likelihood measure, divided by the sum	3.946e-3 / 1.689e-2 = **0.234**	2.742e-4 / 1.689e-2 = **0.016**	1.267e-2 / 1.689e-2 = **0.750**	

A worked-out example for EM

Responsibility matrix	Cluster A	Cluster B	Cluster C
Data point 0	0.007	0.938	0.055
Data point 1	0.812	0.154	0.034
Data point 2	0.234	0.016	0.750
Soft counts	1.053	1.108	0.839

A worked-out example for EM

- M-step: Compute cluster parameters, given cluster responsibilities
- Now that we've computed the cluster responsibilities, we must revise the parameters: cluster weights, means, and covariances. Even though the initial set of parameters was a wild guess, the updated parameters will be a better guess, as they will incorporate how the data points interact with the clusters. (Again, notice a parallel to k-means, where we start with guessing the centroids and fix them later using cluster assignments.)