Expectation Maximization

Presenter: Xuan Tran

https://www.youtube.com/watch?v=REypj2sy_5U&list=PLBv09BD7ez_4e9LtmK626Evn1ion6ynrt&index=1

https://www.youtube.com/watch?v=rVfZHWTwXSA

https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/slides/lec16-slides.pdf

http://www.inf.ed.ac.uk/teaching/courses/iaml/slides/mix-2x2.pdf

Mixture models

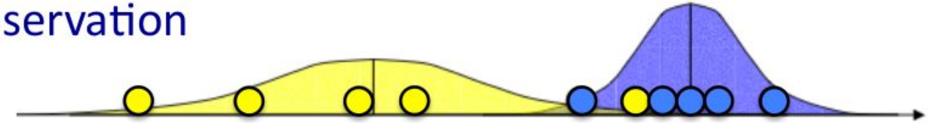
- Recall types of clustering methods
  - hard clustering: clusters do not overlap
    - element either belongs to cluster or it does not
  - soft clustering: clusters may overlap
    - stength of association between clusters and instances
- Mixture models
  - probabilistically-grounded way of doing soft clustering
  - each cluster: a generative model (Gaussian or multinomial)
  - parameters (e.g. mean/covariance are unknown)
- Expectation Maximization (EM) algorithm
  - automatically discover all parameters for the K "sources"

Mixture models in 1-d

- Observations x_1 … x_n
  - K=2 Gaussians with unknown μ, σ^2
  - estimation trivial if we know the source of each observation

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

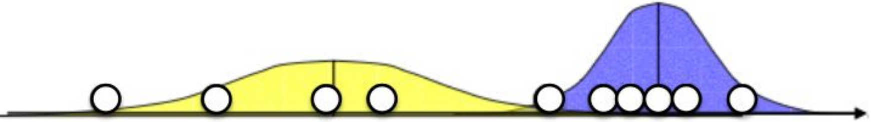
$$\sigma_b^2 = \frac{(x_n - \mu_b)^2 + \ldots + (x_n - \mu_b)^2}{n_b}$$



- If we knew parameters of the Gaussians (μ, σ^2)
  - can guess whether point is more likely to be a or b

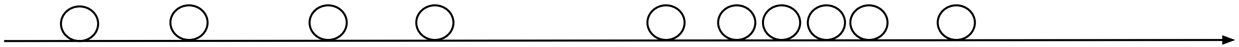
$$P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

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



Expectation Maximization

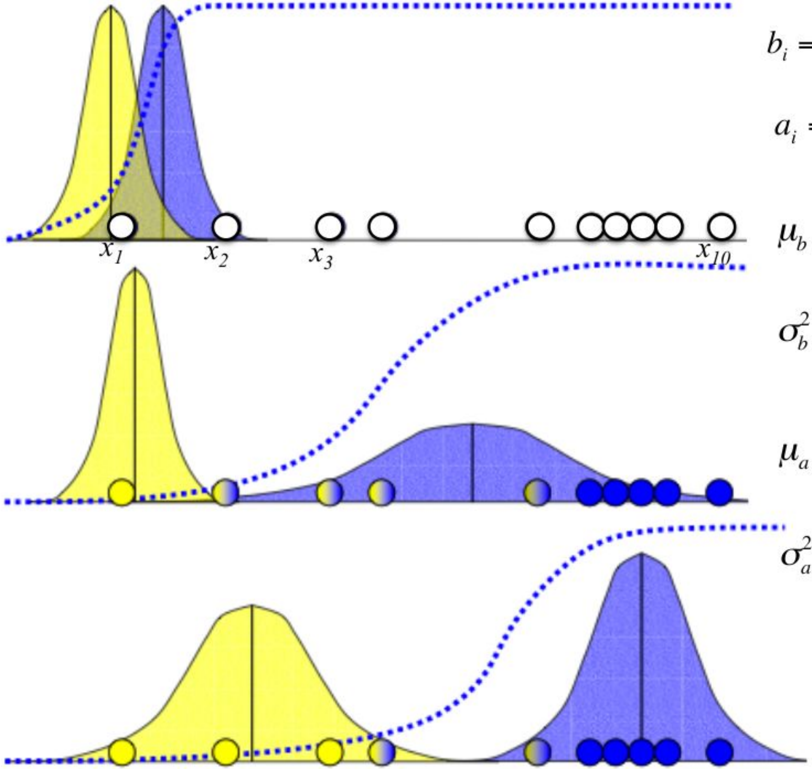
How to deal with the data with no label and no Gaussian parameters???



Expectation Maximization

- Chicken and egg problem
  - need (μ_a, σ_a^2) and (μ_b, σ_b^2) to guess source of points
  - need to know source to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2)
- EM algorithm
  - start with two randomly placed Gaussians (μ_a, σ_a^2), (μ_b, σ_b^2)
  - E-step: for each point: $P(b|x_i)$ = does it look like it came from b?
  - M-step: adjust (μ_a, σ_a^2) and (μ_b, σ_b^2) to fit points assigned to them
  - iterate until convergence

Expectation Maximization



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

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

$$a_i = P(a \mid x_i) = 1 - b_i$$

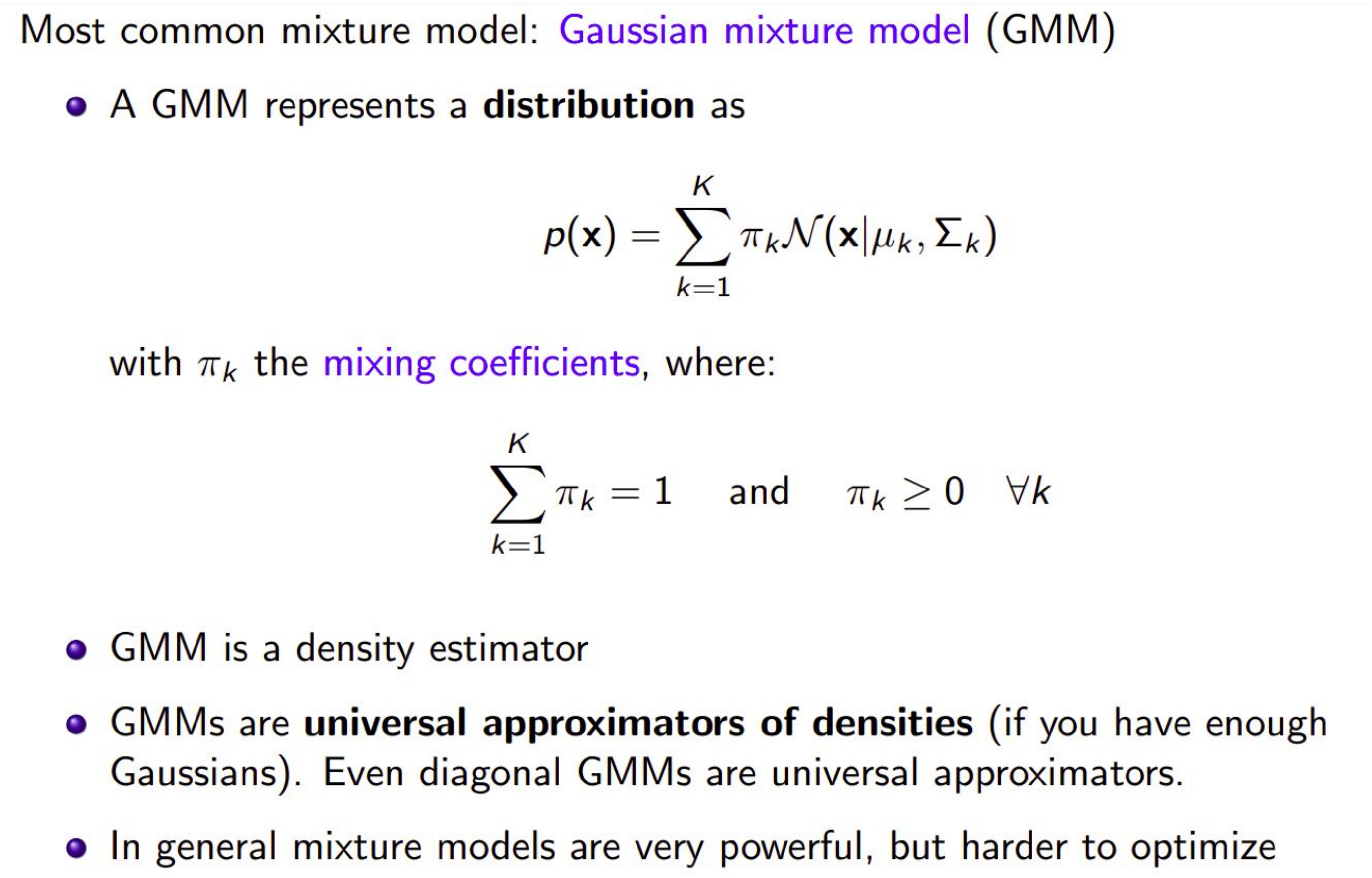
$$\mu_b = \frac{b_1 x_1 + b_2 x_2 + \ldots + b_n x_n}{b_1 + b_2 + \ldots + b_n}$$

$$\sigma_b^2 = \frac{b_1(x_1 - \mu_b)^2 + \ldots + b_n(x_n - \mu_b)^2}{b_1 + b_2 + \ldots + b_n}$$

$$\mu_a = \frac{a_1 x_1 + a_2 x_2 + \ldots + a_n x_n}{a_1 + a_2 + \ldots + a_n}$$

$$\sigma_a^2 = \frac{a_1(x_1 - \mu_a)^2 + \ldots + a_n(x_n - \mu_a)^2}{a_1 + a_2 + \ldots + a_n}$$

Gaussian mixture model (GMM)



Most common mixture model: Gaussian mixture model (GMM)

- A GMM represents a **distribution** as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

  with π_k the mixing coefficients, where:

$$\sum_{k=1}^{K} \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \quad \forall k$$

- GMM is a density estimator
- GMMs are **universal approximators of densities** (if you have enough Gaussians). Even diagonal GMMs are universal approximators.
- In general mixture models are very powerful, but harder to optimize

The Partition Theorem (Law of Total Probability)

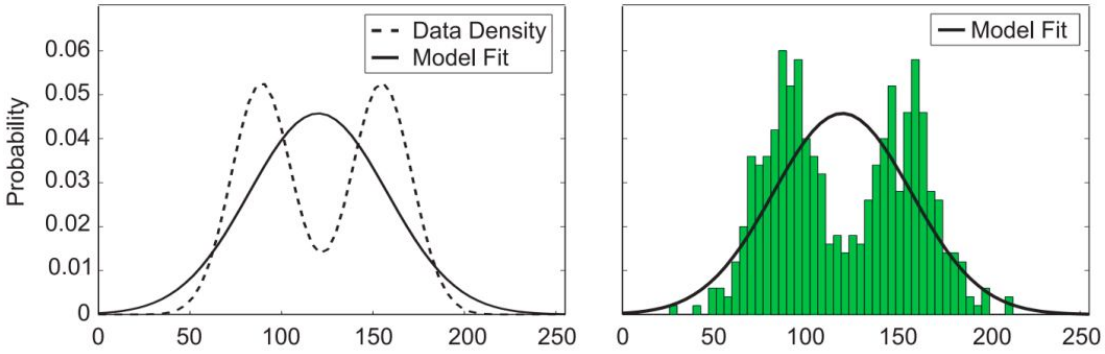
Let $B_1, \ldots, B_m$ form a partition of Ω. Then for any event A,

$$\boxed{\mathbb{P}(A) = \sum_{i=1}^{m} \mathbb{P}(A \cap B_i) = \sum_{i=1}^{m} \mathbb{P}(A \mid B_i)\mathbb{P}(B_i)}$$

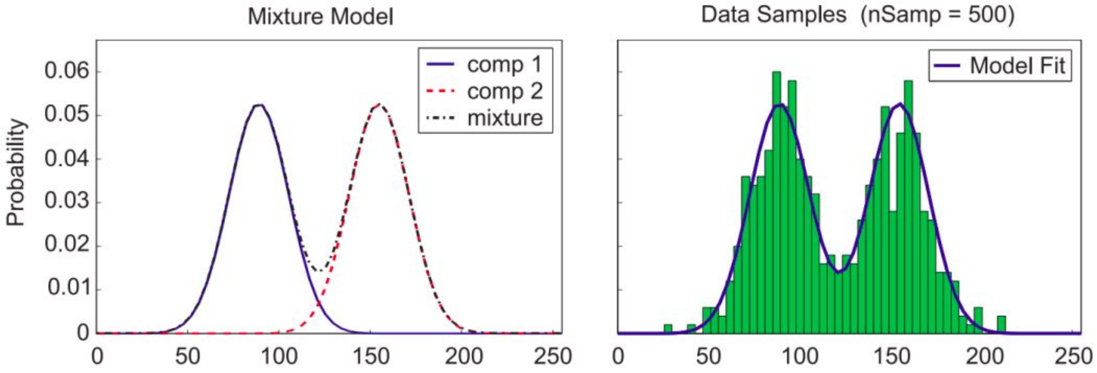
Both formulations of the Partition Theorem are very widely used, but especially the conditional formulation $\sum_{i=1}^{m} \mathbb{P}(A \mid B_i)\mathbb{P}(B_i)$.

Gaussian mixture model (GMM)

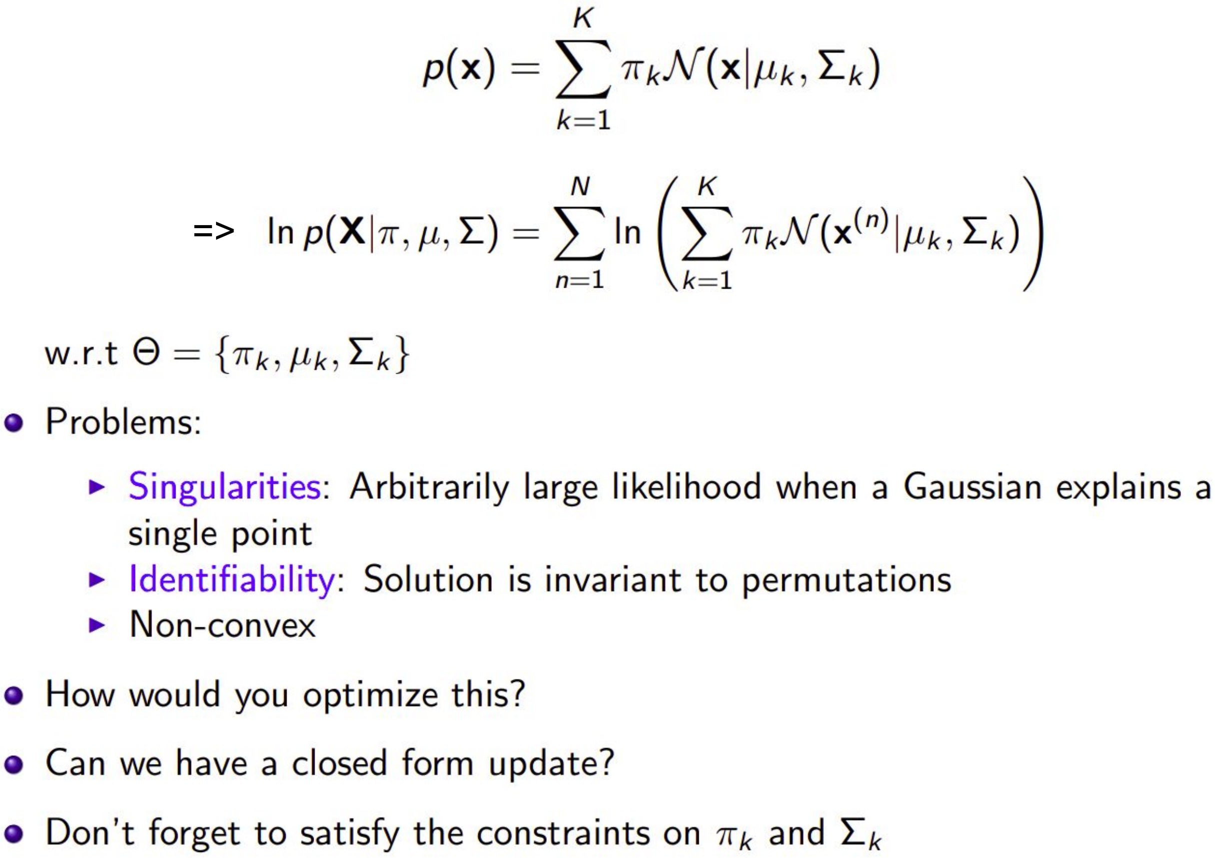
- If you fit a Gaussian to data:



- Now, we are trying to fit a GMM (with $K = 2$ in this example):



GMM: Maximum Likelihood



$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

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

w.r.t $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

- Problems:
  - Singularities: Arbitrarily large likelihood when a Gaussian explains a single point
  - Identifiability: Solution is invariant to permutations
  - Non-convex
- How would you optimize this?
- Can we have a closed form update?
- Don't forget to satisfy the constraints on π_k and Σ_k

Latent Variable

- Our original representation had a hidden (latent) variable z which would represent which Gaussian generated our observation $\mathbf{x}$, with some probability
- Let $z \sim \mathrm{Categorical}(\boldsymbol{\pi})$ $\quad$ (where $\pi_k \geq 0, \quad \sum_k \pi_k = 1$)
- Then:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}, z = k)$$

$$= \sum_{k=1}^{K} \underbrace{p(z = k)}_{\pi_k} \underbrace{p(\mathbf{x}|z = k)}_{\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}$$

- This breaks a complicated distribution into simple components - the price is the hidden variable.

Back to GMM

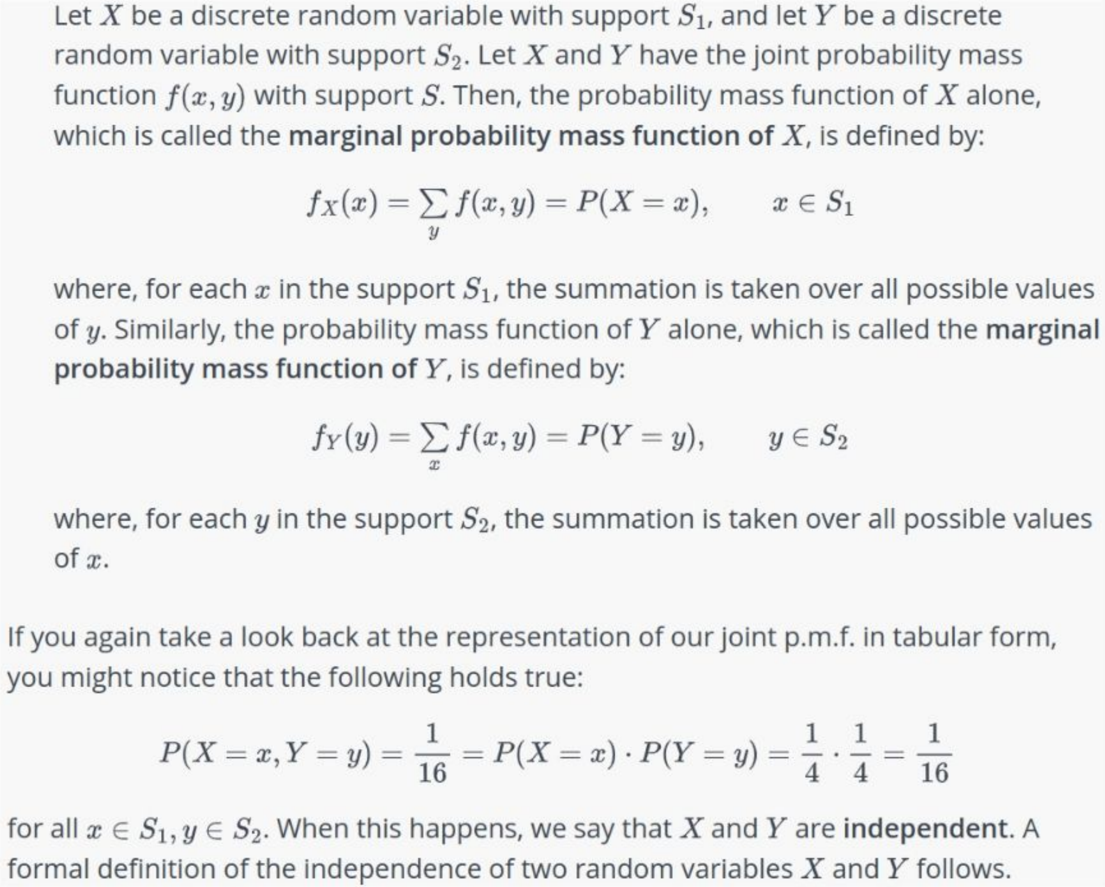
- A Gaussian mixture distribution:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

- We had: $z \sim \mathrm{Categorical}(\boldsymbol{\pi}) \quad (\text{where } \pi_k \geq 0, \quad \sum_k \pi_k = 1)$
- Joint distribution: $\quad p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$
- Log-likelihood:

$$\ell(\mathbf{X}, \Theta) = \sum_i \log(P(\mathbf{x}^{(i)}; \Theta)) = \sum_i \log\left(\sum_j P(\mathbf{x}^{(i)}, z^{(i)} = j; \Theta)\right)$$

Marginal Probability Mass function of X



Let X be a discrete random variable with support S_1, and let Y be a discrete random variable with support S_2. Let X and Y have the joint probability mass function $f(x, y)$ with support S. Then, the probability mass function of X alone, which is called the **marginal probability mass function of X**, is defined by:

$$f_X(x) = \sum_y f(x, y) = P(X = x), \qquad x \in S_1$$

where, for each x in the support S_1, the summation is taken over all possible values of y. Similarly, the probability mass function of Y alone, which is called the **marginal probability mass function of Y**, is defined by:

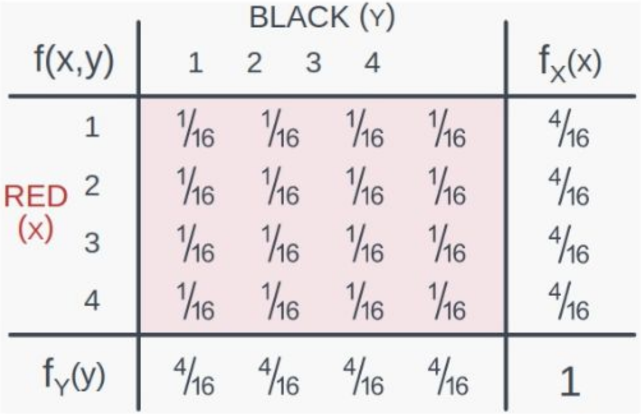
$$f_Y(y) = \sum_x f(x, y) = P(Y = y), \qquad y \in S_2$$

where, for each y in the support S_2, the summation is taken over all possible values of x.

If you again take a look back at the representation of our joint p.m.f. in tabular form, you might notice that the following holds true:

$$P(X = x, Y = y) = \frac{1}{16} = P(X = x) \cdot P(Y = y) = \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$$

for all $x \in S_1, y \in S_2$. When this happens, we say that X and Y are **independent**. A formal definition of the independence of two random variables X and Y follows.



	f(x,y)	BLACK (Y) 1	2	3	4	$f_X(x)$
RED (X)	1	1/16	1/16	1/16	1/16	4/16
	2	1/16	1/16	1/16	1/16	4/16
	3	1/16	1/16	1/16	1/16	4/16
	4	1/16	1/16	1/16	1/16	4/16
	$f_Y(y)$	4/16	4/16	4/16	4/16	1

5P GJOE UIF QSPCBCJMJUZ NBTT GVODUJPO PG9
XF TVN GPS FBDI Y UIF QSPCBCJMJUJFT XIFO
Z BOE 5IBUT GPS FBDI Y XF TVN
GY GY GY BOE GY

E Step

- Remember that optimizing the likelihood is hard because of the sum inside of the log. Using Θ to denote all of our parameters:

$$\ell(\mathbf{X}, \Theta) = \sum_i \log(P(\mathbf{x}^{(i)}; \Theta)) = \sum_i \log\left(\sum_j P(\mathbf{x}^{(i)}, z^{(i)} = j; \Theta)\right)$$

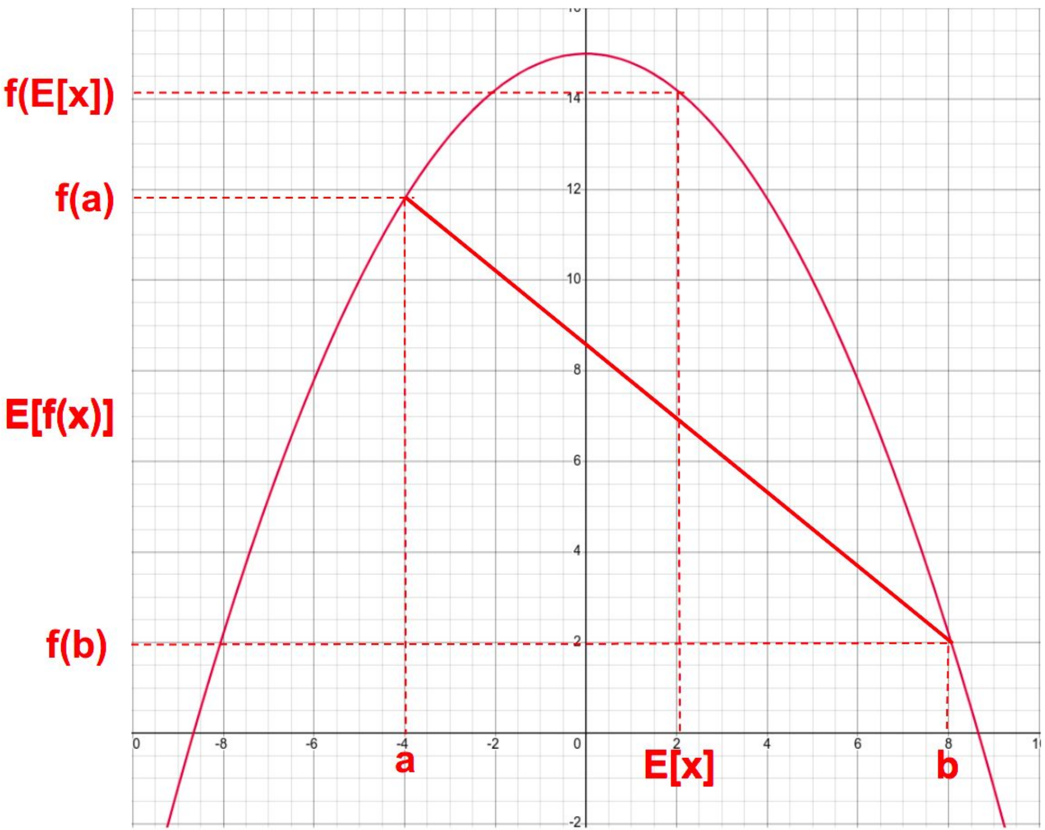
- We can use a common trick in machine learning, introduce a new distribution, q:

$$\ell(\mathbf{X}, \Theta) = \sum_i \log\left(\sum_j q_j \frac{P(\mathbf{x}^{(i)}, z^{(i)} = j; \Theta)}{q_j}\right)$$

- Now we can swap them! Jensen's inequality - for concave function (like log)

$$f(\mathbb{E}[x]) = f\left(\sum_i p_i x_i\right) \geq \sum_i p_i f(x_i) = \mathbb{E}[f(x)]$$

Jensen’s Inequality



$$f(\alpha) = \log(\alpha)$$

$$\alpha(z_i) = \frac{P(x_i, z_i; \theta)}{q_j}$$

$$P(z_i = j) = q_j$$

$$f(E[\alpha]) = ?\,;\ E[f(\alpha)] = ?$$

$$\mathbb{E}(X) = \sum_x \mathbb{P}(X = x) \times x.$$

$$\mathbb{E}\{g(X)\} = \sum_x g(x)\mathbb{P}(X = x).$$

E Step

- Applying Jensen's,

$$\sum_i \log\left(\sum_j q_j \frac{P(\mathbf{x}^{(i)}, z^{(i)} = j; \Theta)}{q_j}\right) \geq \sum_i \sum_j q_j \log\left(\frac{P(\mathbf{x}^{(i)}, z^{(i)} = j; \Theta)}{q_j}\right)$$

- Maximizing this lower bound will force our likelihood to increase.
- But how do we pick a q_i that gives a good bound?

E Step

- We got the sum outside but we have an inequality.

$$\ell(\mathbf{X}, \Theta) \geq \sum_i \sum_j q_j \log \left(\frac{P(\mathbf{x}^{(i)}, z^{(i)} = j; \Theta)}{q_j} \right)$$

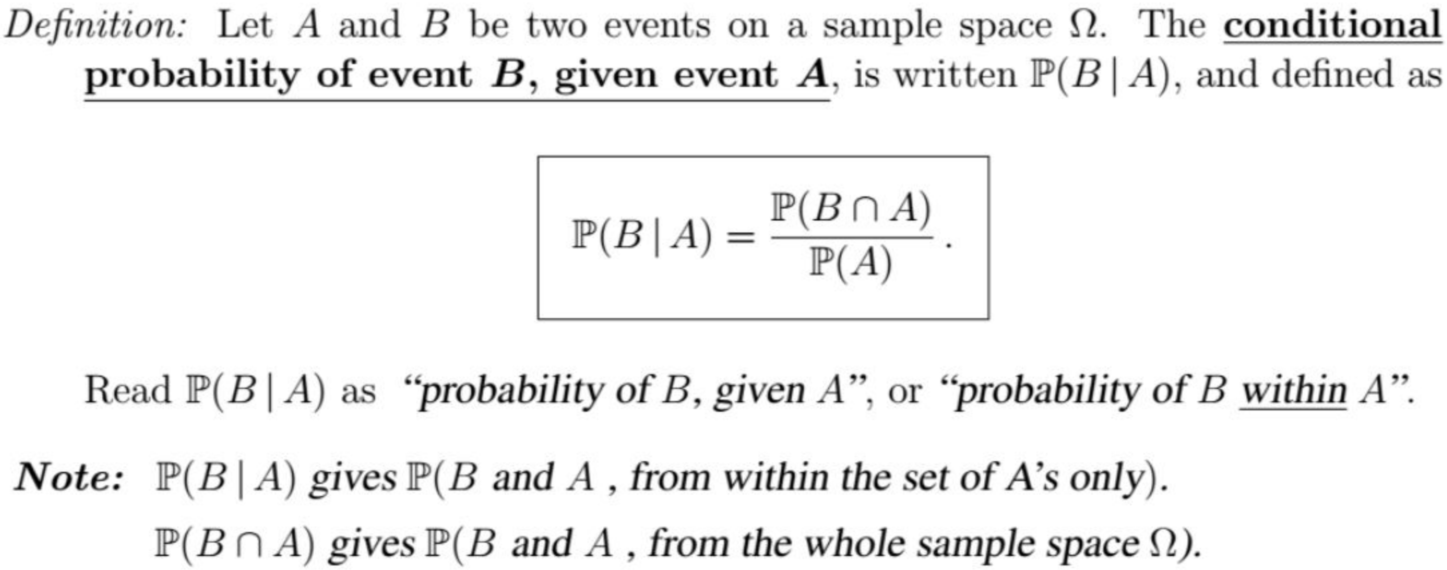
- Lets fix the current parameters to Θ^{old} and try to find a good q_i
- What happens if we pick $\boxed{q_j = p(z^{(i)} = j | x^{(i)}, \Theta^{old})}$?
  - $\frac{P(\mathbf{x}^{(i)}, z^{(i)}; \Theta)}{p(z^{(i)} = j | x^{(i)}, \Theta^{old})} = P(\mathbf{x}^{(i)}; \Theta^{old})$ and the inequality becomes an equality!
- We can now define and optimize

$$Q(\Theta) = \sum_i \sum_j p(z^{(i)} = j | x^{(i)}, \Theta^{old}) \log \left(P(\mathbf{x}^{(i)}, z^{(i)} = j; \Theta) \right)$$

$$= \mathbb{E}_{P(z^{(i)} | \mathbf{x}^{(i)}, \Theta^{old})} [\log \left(P(\mathbf{x}^{(i)}, z^{(i)}; \Theta) \right)]$$

- We ignored the part that doesn't depend on Θ

Formula for conditional probability



Definition: Let A and B be two events on a sample space Ω. The **conditional probability of event B, given event A**, is written $\mathbb{P}(B \mid A)$, and defined as

$$\boxed{\mathbb{P}(B \mid A) = \frac{\mathbb{P}(B \cap A)}{\mathbb{P}(A)}\,.}$$

Read $\mathbb{P}(B \mid A)$ as *"probability of B, given A"*, or *"probability of B within A"*.

Note: $\mathbb{P}(B \mid A)$ *gives* $\mathbb{P}(B$ *and* A *, from within the set of A's only*).

$\mathbb{P}(B \cap A)$ *gives* $\mathbb{P}(B$ *and* A *, from the whole sample space* Ω*)*.

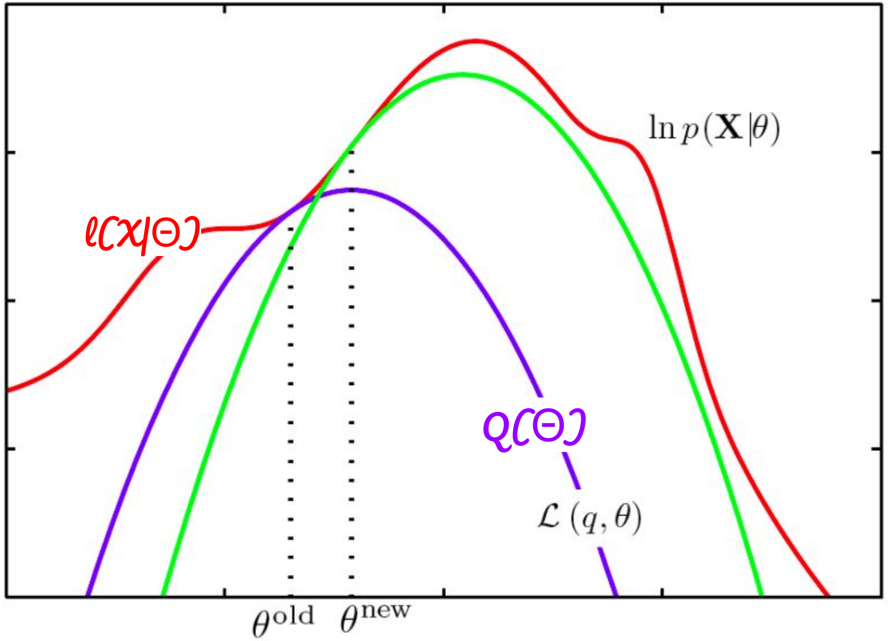
M Step

- So, what just happened?
- Conceptually: We don't know $z^{(i)}$ so we average them given the current model.
- Practically: We define a function $Q(\Theta) = \mathbb{E}_{P(z^{(i)}|\mathbf{x}^{(i)},\Theta^{old})}[\log\left(P(\mathbf{x}^{(i)}, z^{(i)}; \Theta)\right)]$ that lower bounds the desired function and is equal at our current guess.
- If we now optimize Θ we will get a better lower bound!

$$\log(P(\mathbf{X}|\Theta^{old})) = Q(\Theta^{old}) \leq Q(\Theta^{new}) \leq \log(P(\mathbf{X}|\Theta^{new}))$$

- We can iterate between expectation step and maximization step and the lower bound will always improve (or we are done)

EM Derivation



- The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values.

EM Algorithm

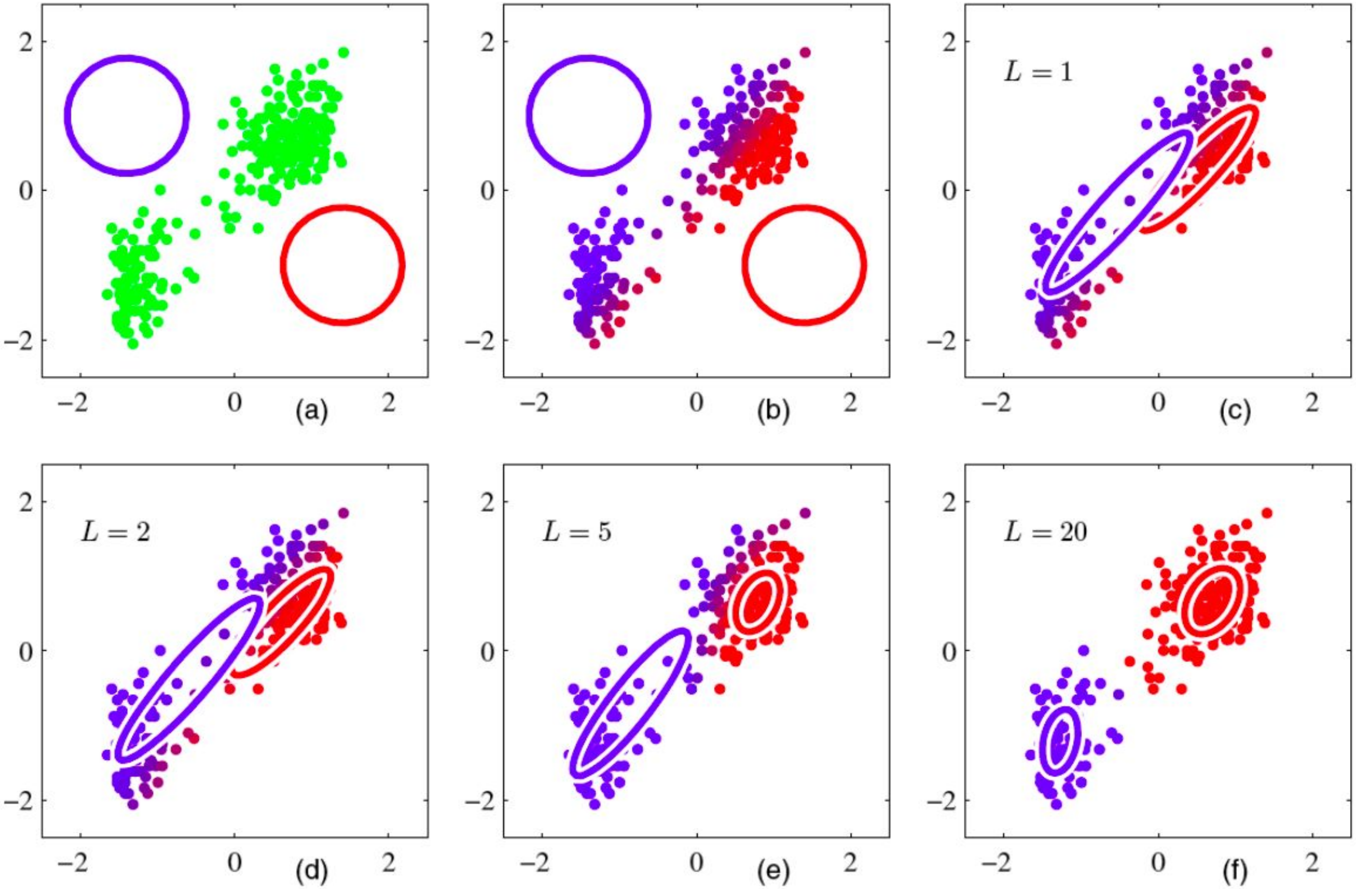
E-step:

Set $q_j = P(z_i = j \mid x_i \; ; \; \theta)$

M-step:

$$argmax_{\theta} \sum_{i} \sum_{j} q_j \log \left(\frac{\mathrm{P}(\mathrm{x_i}, \mathrm{z_i}; \theta)}{q_j} \right) = argmax_{\theta} \sum_{i} \sum_{j} \log \left(\mathrm{P}(\mathrm{z_i} = \mathrm{j}) \times \mathrm{P}(\mathrm{x_i} \mid \mathrm{z_i} = \mathrm{j}) \right)$$

EM Algorithm



EM vs. K-means

- EM for mixtures of Gaussians is just like a soft version of K-means, with fixed priors and covariance
- Instead of hard assignments in the E-step, we do soft assignments based on the softmax of the squared Mahalanobis distance from each point to each cluster.
- Each center moved by weighted means of the data, with weights given by soft assignments
- In K-means, weights are 0 or 1

THE END