CS 229 Lecture Twelve
Unsupervised Learning:
k-Means and Gaussian Mixture Models

Chris Ré

May 6, 2023

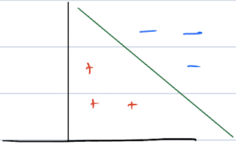
Unsupervised Learning: Our Plan

We begin our tour of unsupervised (and weakly) learning:

- In the next four lectures, we'll learn general techniques for latent variable models including **Expectation Maximization** (EM) and **method of moments** and we'll study many settings.
- We'll see a fun application that is near to my heart and is also in systems that you probably used today **weak supervision**.
- Recent trend *incredibly* weak forms of supervision.
- **Today** We start with k-means, Gaussian Mixture Models (GMMs).

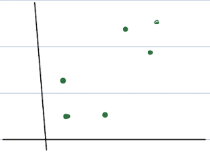
> These techniques and ideas are useful, but this section forces us to grapple with modeling questions in machine learning.

Unsupervised Learning In Pictures



Supervised Setting

Unsupervised is harder than supervised

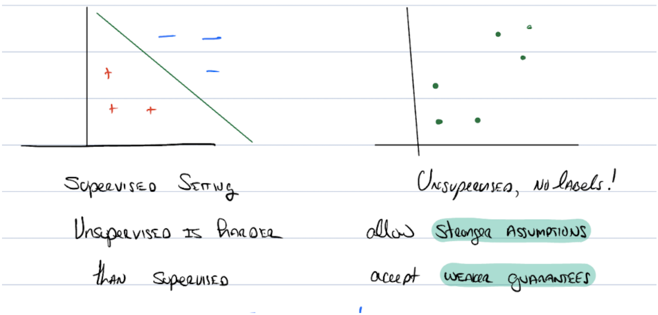


Unsupervised, no labels!

allow stronger assumptions

accept weaker guarantees

Unsupervised Learning In Pictures



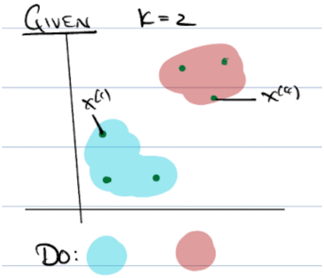
Unsupervised learning is "harder" than supervised, so we'll make *stronger* assumptions and accept *weaker guarantees*.

Our Plan for Lecture

- Start with k-Means clustering a (hopefully!) intuitive method
- A probabilistic method, Gaussian Mixture Model (GMMs)
- **Detour** Convexity and Jensen's inequality (in pictures)
- A first taste of EM (for GMMs) via *maximum likelihood*

k-Means (Picture)

Given $k = 2$ and the following data find clusters.

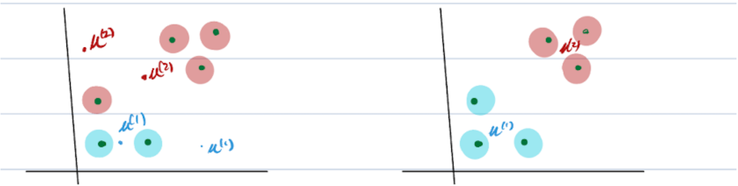


- **Given** an integer k (the number of clusters) and $x^{(1)}, \dots, x^{(n)} \in \mathbb{R}^d$.
- **Do** find an assignment of $x^{(i)}$ to one of the k clusters.

$$C^{(i)} = j \text{ means point } i \text{ in cluster } j$$

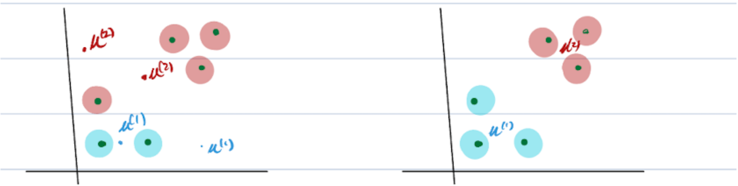
e.g., $C^{(2)} = 2$ and $C^{(4)} = 1$

How do we find these clusters? (Iterative Approach)



- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.

How do we find these clusters? (Iterative Approach)



- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.
- Assign each point, $x^{(i)}$, to closest cluster

$$C^{(i)} = \underset{j=1,\ldots,k}{\operatorname{argmin}} \|\mu^{(j)} - x^{(i)}\|^2 \text{ for } i = 1, \ldots, n$$

How do we find these clusters? (Iterative Approach)

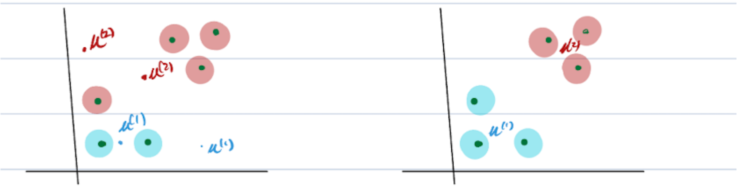
- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.
- Assign each point, $x^{(i)}$, to closest cluster

$$C^{(i)} = \operatorname*{argmin}_{j=1,\ldots,k} \|\mu^{(j)} - x^{(i)}\|^2 \text{ for } i = 1, \ldots, n$$

- Compute new center of each cluster:

$$\mu^{(j)} = \frac{1}{|\Omega_j|} \sum_{i \in \Omega_j} x^{(i)} \text{ where } \Omega_j = \{i : C^{(i)} = j\}$$

How do we find these clusters? (Iterative Approach)



- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.
- Assign each point, $x^{(i)}$, to closest cluster

$$C^{(i)} = \operatorname*{argmin}_{j=1,\dots,k} \|\mu^{(j)} - x^{(i)}\|^2 \text{ for } i = 1, \dots, n$$

- Compute new center of each cluster:

$$\mu^{(j)} = \frac{1}{|\Omega_j|} \sum_{i \in \Omega_j} x^{(i)} \text{ where } \Omega_j = \{i : C^{(i)} = j\}$$

Repeat until clusters stay the same!

Comments about k-means

- Does it terminate?

[1] https://en.wikipedia.org/wiki/K-means%2B%2B

Comments about k-means

- Does it terminate? Yes, see notes! It minimizes

$$J(C, \mu) = \sum_{i=1}^{n} \|x^{(i)} - \mu^{C^{(i)}}\|^2 \text{ decreases momonotonically.}$$

[1] `https://en.wikipedia.org/wiki/K-means%2B%2B`

Comments about k-means

- Does it terminate? Yes, see notes! It minimizes

$$J(C, \mu) = \sum_{i=1}^{n} \|x^{(i)} - \mu^{C^{(i)}}\|^2 \text{ decreases momonotonically.}$$

- Does it find a *global minimum*?

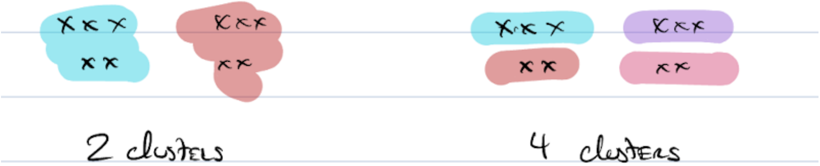
[1] https://en.wikipedia.org/wiki/K-means%2B%2B

Comments about k-means

- Does it terminate? Yes, see notes! It minimizes

$$J(C, \mu) = \sum_{i=1}^{n} \|x^{(i)} - \mu^{C^{(i)}}\|^2 \text{ decreases momonotonically.}$$

- Does it find a *global minimum*? No, it's an NP-Hard problem!
- Side Note: k-means $++$ from great Stanford folks[1]
  - Improved Approximation Ratio and default in SKLearn!
- How do you choose k? *It's a modeling question!*



[1] https://en.wikipedia.org/wiki/K-means%2B%2B

Mixture of Gaussians

Mixture of Gaussians

Toy Astronomy example based on a real UW paper.

- Both quasars and stars are source of light (photons).
- We observe photons–but source is unclear.

Mixture of Gaussians

Toy Astronomy example based on a real UW paper.

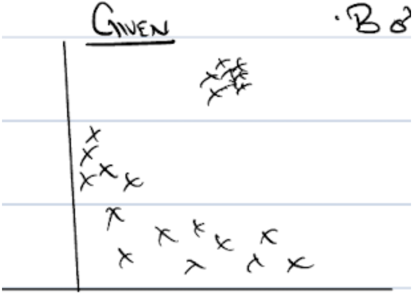
- Both quasars and stars are source of light (photons).
- We observe photons–but source is unclear.

Do Assign each photon to a light source:

$P(z^{(i)} = j)$ is the probability point $z^{(i)}$ belongs to source j

Compare with k-means, a **soft** (probabilistic) assignment

Challenges and Assumptions



Modeling Challenges

- Many sources: Assume we know the number of sources k.
- Sources can have different intensities and shapes!

Challenges and Assumptions

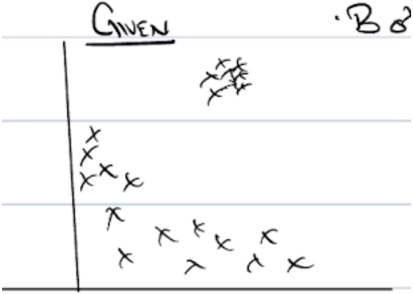
Modeling Challenges

- Many sources: Assume we know the number of sources k.
- Sources can have different intensities and shapes!

Assumptions

- **Unknown Shape** Sources are modeled by Guassian (μ_j, σ_j^2)
- **Unknown Mixture** We do *not* assume equal number of points from each source.

Challenges and Assumptions



Modeling Challenges

- Many sources: Assume we know the number of sources k.
- Sources can have different intensities and shapes!

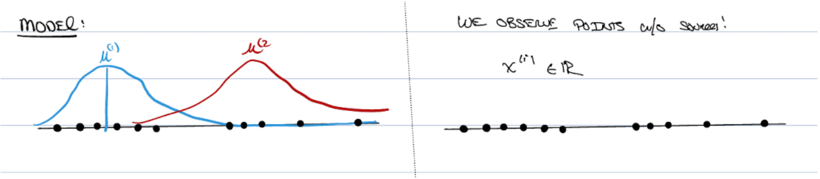
Assumptions

- **Unknown Shape** Sources are modeled by Guassian (μ_j, σ_j^2)
- **Unknown Mixture** We do *not* assume equal number of points from each source.

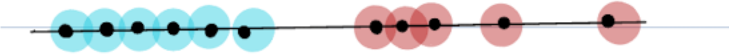
NB: Scientists can check if these values make sense!

The Different Shapes of Guassians

Mixture of Guassians – Model and Setup (1d)



Observation If we know the "cluster labels", we could find "cluster shapes" with GDA!



A key challenge is that we *do not* have these labels–need to estimate them!

Mixture of Guassians: Formal Version

- **Given** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}$ and positive integer k (sources)
- **Do** Find P s.t. for each data point ($i = 1, \ldots, n$) and each source ($j = 1, \ldots, k$) we find a *soft assignment*

$$P(z^{(i)} = j)$$

Mixture of Guassians: Formal Version

- **Given** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}$ and positive integer k (sources)
- **Do** Find P s.t. for each data point ($i = 1, \ldots, n$) and each source ($j = 1, \ldots, k$) we find a *soft assignment*

$$P(z^{(i)} = j)$$

The probability P is modeled via the Guassian Mixture Model,

$$\begin{aligned} P(x^{(i)}, z^{(i)}) &= P(x^{(i)} | z^{(i)}) P(z^{(i)}) && \text{Bayes' Rule} \\ z^{(i)} &\sim \text{Multinomial}(\phi) && \text{Mixture of sources} \\ x^{(i)} \mid z^{(i)} = j &\sim \mathcal{N}(\mu_j, \sigma_j^2) && \text{Guassian in each source} \end{aligned}$$

Mixture of Guassians: Formal Version

- **Given** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}$ and positive integer k (sources)
- **Do** Find P s.t. for each data point ($i = 1, \ldots, n$) and each source ($j = 1, \ldots, k$) we find a *soft assignment*

$$P(z^{(i)} = j)$$

The probability P is modeled via the Guassian Mixture Model,

$$\begin{aligned} P(x^{(i)}, z^{(i)}) &= P(x^{(i)}|z^{(i)})P(z^{(i)}) && \text{Bayes' Rule} \\ z^{(i)} &\sim \text{Multinomial}(\phi) && \text{Mixture of sources} \\ x^{(i)} \mid z^{(i)} = j &\sim \mathcal{N}(\mu_j, \sigma_j^2) && \text{Guassian in each source} \end{aligned}$$

We call $z^{(i)}$ a **hidden** or **latent variable**, as the value of $z^{(i)}$ is *not* directly observed. The parameters of the model $\phi, \mu_1, \sigma_1, \ldots, \mu_k, \sigma_k$, are in the color blue

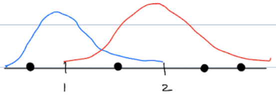
Mixture of Guassians: Unpack Model by Sampling

P s.t. for each data point ($i = 1, \ldots, n$) and each source ($j = 1, \ldots, k$) we find a *soft assignment* $P(z^{(i)} = j)$

$P(x^{(i)}, z^{(i)}) = P(x^{(i)} \mid z^{(i)}) P(z^{(i)})$	Bayes' Rule
$z^{(i)} \sim \text{Multinomial}(\phi)$	Mixture of sources
$x^{(i)} \mid z^{(i)} = j \sim \mathcal{N}(\mu_j, \sigma_j^2)$	Guassian in each source

Suppose we did know parameters $\phi, \mu_1, \sigma_1^2, \ldots, \mu_k, \sigma_k^2$, imagine data $x^{(i)}$ generated by a sampling procedure:

Example "think Sampling"



$\phi_1 = 0.7$ $\phi_2 = 0.3$

$\mu_1 = 1$ $\mu_2 = 2$ $\sigma_1^2 = \sigma_2^2 = \frac{1}{3}$ (roughly)

For each data point i,

- Pick cluster 1 prob. $\phi_1 = 0.7$ or 2 $\phi_2 = 0.3$, call that $z^{(i)}$
- Suppose point i assigned to cluster $z^{(i)}$, sample from Guassian with mean $\mu_{z(i)}$, that's your sample $x^{(i)}$

Recap: The Key Idea of the Latent Model

- Given a set of parameters, we can assess how likely the observed data $x^{(1)}, \ldots, x^{(n)}$ is according to the GMM model.
- As usual, we turn this observation on its head: The likelihood model of the GMM is enough for us to estimate those parameters from the observed data.
- **The twist** is that $z^{(i)}$ is latent model, that is we do not observe the value of $z^{(i)}$. However, we do know something about its structure (e.g., there are k clusters)

Let's see an Algorithm, which will look like k-means *and* in later lectures we'll relate to our old friend MLE.

GMM Algorithm (Mirrors k-Means)

Sketch of the Expectation Maximization Algorithm (EM):

- **E-Step** *"Guess the latent values of $z^{(i)}$"* for each point $i = 1, \ldots, n$.
- **M-Step** Update the parameters.

GMM Algorithm (Mirrors k-Means)

Sketch of the Expectation Maximization Algorithm (EM):

- **E-Step** *"Guess the latent values of $z^{(i)}$"* for each point $i = 1, \ldots, n$.
- **M-Step** Update the parameters.

E-Step in more detail

- **Given**: Data, $x^{(1)}, \ldots, x^{(n)}$, and current estimate of parameters $\phi, \mu_1, \sigma_1^2, \ldots, \mu_k, \sigma_k^2$.
- **Do:** For each $i = 1, \ldots, n$ and $j = 1, \ldots, k$, estimate the probability of

$$w_j^{(i)} = P(z^{(i)} = j | x^{(i)}; \phi, \mu, \sigma)$$

That is, write $w_j^{(i)}$ in terms of $\phi, \mu_1, \sigma_1^2, \ldots, \mu_k, \sigma_k^2$.

Deriviation of the E-step

$$w_j^{(i)} = P(z^{(i)} = j \mid x^{(i)}; \phi, \mu, \sigma) \qquad \text{our goal}$$

Deriviation of the E-step

$$w_j^{(i)} = P(z^{(i)} = j \mid x^{(i)}; \phi, \mu, \sigma) \qquad \text{our goal}$$

$$= \frac{P(z^{(i)} = j, x^{(i)}; \phi, \mu, \sigma)}{P(x^{(i)}; \phi, \mu, \sigma)} \qquad \text{Bayes' Rule}$$

Deriviation of the E-step

$$
\begin{aligned}
w_j^{(i)} &= P(z^{(i)} = j \mid x^{(i)}; \phi, \mu, \sigma) && \text{our goal} \\
&= \frac{P(z^{(i)} = j, x^{(i)}; \phi, \mu, \sigma)}{P(x^{(i)}; \phi, \mu, \sigma)} && \text{Bayes' Rule} \\
&= \frac{P(x^{(i)} \mid z^{(i)} = j; \phi, \mu, \sigma) P(z^{(i)} = j)}{P(x^{(i)}; \phi, \mu, \sigma)} && \text{Bayes' Rule}
\end{aligned}
$$

Deriviation of the E-step

$$
\begin{aligned}
w_j^{(i)} =& P(z^{(i)} = j \mid x^{(i)}; \phi, \mu, \sigma) && \text{our goal} \\
=& \frac{P(z^{(i)} = j, x^{(i)}; \phi, \mu, \sigma)}{P(x^{(i)}; \phi, \mu, \sigma)} && \text{Bayes' Rule} \\
=& \frac{P(x^{(i)} \mid z^{(i)} = j; \phi, \mu, \sigma) P(z^{(i)} = j)}{P(x^{(i)}; \phi, \mu, \sigma)} && \text{Bayes' Rule} \\
=& \frac{P(x^{(i)} \mid z^{(i)} = j; \phi, \mu, \sigma) P(z^{(i)} = j; \phi, \mu, \sigma)}{\sum_{l=1}^{k} P(x^{(i)} | z^{(i)} = l; \phi, \mu, \sigma) P(z^{(i)} = l; \phi, \mu, \sigma)} && \text{Bayes' Rule}
\end{aligned}
$$

Deriviation of the E-step

$$\begin{aligned}
w_j^{(i)} =& P(z^{(i)} = j \mid x^{(i)}; \phi, \mu, \sigma) && \text{our goal} \\
=& \frac{P(z^{(i)} = j, x^{(i)}; \phi, \mu, \sigma)}{P(x^{(i)}; \phi, \mu, \sigma)} && \text{Bayes' Rule} \\
=& \frac{P(x^{(i)} \mid z^{(i)} = j; \phi, \mu, \sigma) P(z^{(i)} = j)}{P(x^{(i)}; \phi, \mu, \sigma)} && \text{Bayes' Rule} \\
=& \frac{P(x^{(i)} \mid z^{(i)} = j; \phi, \mu, \sigma) P(z^{(i)} = j; \phi, \mu, \sigma)}{\sum_{l=1}^{k} P(x^{(i)} | z^{(i)} = l; \phi, \mu, \sigma) P(z^{(i)} = l; \phi, \mu, \sigma)} && \text{Bayes' Rule}
\end{aligned}$$

Key point: We can compute all terms from the parameters!

$$P(x^{(i)} \mid z^{(i)} = j; \phi, \mu, \sigma) \text{ is } \mathcal{N}(\mu_j, \sigma_j^2)$$
$$P(z^{(i)} = j; \phi, \mu, \sigma) = \phi_j$$

Recall: Now for the M-Step

Sketch of the Expectation Maximization Algorithm (EM):

- **E-Step** *"Guess the latent values of $z^{(i)}$"* for each point $i = 1, \dots, n$.
- **M-Step** Update the parameters.

M-Step in more detail:

- **Given** $w_j^{(i)}$ our current estimate of $P(z^{(i)} = j)$ for $i = 1, \dots, n$ and $j = 1, \dots, k$.
- **Do** Estimate the parameters $\phi, \mu_1, \sigma_1^2, \dots, \mu_n, \sigma_n^2$.

This is just MLE (we'll show this soon!) but:

$$\phi_j = \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)} \qquad \text{fractional elements from source } j$$

$$\mu_j = \frac{\sum_{i=1}^{n} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{n} w_j^{(i)}} \qquad \text{points fractionally weighted.}$$

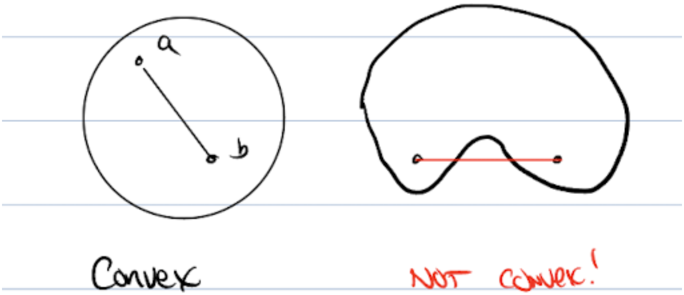
Detour! Convexity and Jensen's Inequality.

Key source of confusion, we'll go slow.

Detour: Convexity & Jensen's Inequality

A set Ω is convex if for any $a, b \in \Omega$, the line joining a, b is in Ω as well. In symbols, Ω is convex if:

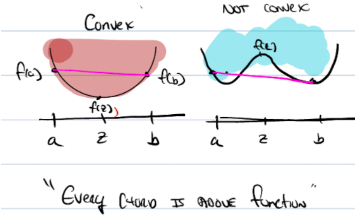
$$\forall a, b \in \Omega . \forall \lambda \in [0,1] \; \lambda a + (1-\lambda) b \in \Omega .$$



Given a function f the graph of f, G_f is a set defined as

$$G_f = \{(x, y) : y \geq f(x)\}$$

A function f is convex if G_f is convex (as a set).



In symbols, the set definition:

$$\lambda(a, f(a)) + (1 - \lambda)(b, f(b)) \in G_f$$

or let $z = \lambda a + (1 - \lambda)b$ then $(z, \lambda f(a) + (1 - \lambda)f(b)) \in G_f$ if

$$\lambda f(a) + (1 - \lambda)f(b) \geq f(z)$$

Convex for Differentiable Functions

If f is twice differentiable, then $\forall x\ f''(x) \geq 0$ then f is convex.
Use Taylor's theorem with remainder:

$$
\begin{aligned}
f(a) &= f(z) + f'(z)(a-z) + f''(\eta_a)(a-z)^2 \qquad \text{for } \eta_a \in [a, z] \\
f(b) &= f(z) + f'(z)(b-z) + f''(\eta_b)(b-z)^2 \qquad \text{for } \eta_b \in [z, b]
\end{aligned}
$$

Convex for Differentiable Functions

If f is twice differentiable, then $\forall x \; f''(x) \geq 0$ then f is convex.
Use Taylor's theorem with remainder:

$$\begin{aligned} f(a) &= f(z) + f'(z)(a-z) + f''(\eta_a)(a-z)^2 \quad &\text{for } \eta_a \in [a, z] \\ f(b) &= f(z) + f'(z)(b-z) + f''(\eta_b)(b-z)^2 \quad &\text{for } \eta_b \in [z, b] \end{aligned}$$

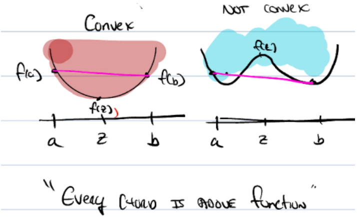
Observe that $f'(z)(\lambda a + (1-\lambda)b - z) = 0$ and since $f''(x) \geq 0$

$$\lambda f(a) + (1-\lambda)f(b) = f(z) + c \text{ for } c \geq 0$$

That is, $\lambda f(a) + (1-\lambda)f(b) \geq f(z)$, i.e., f is convex.

Strongly Convex

We say f is strongly convex if $f''(x) > 0$ for all x in domain of f.



$$f(x) = x^2 \implies f''(x) = 2 > 0 \text{ so strongly convex}$$

$$f(x) = x^2(x-1)^2 \implies f''(x) = 12x^2 - 12x + 1$$

$$f''(0.5) = -2 \text{ so not strongly convex}$$

Jensen's Inequality

For convex f, Jensen's inequality is:

$$\mathbb{E}[f(x)] \geq f(\mathbb{E}[x])$$

A simple example:

$$x \text{ takes value } a \text{ with probability } \lambda$$
$$x \text{ takes value } b \text{ with probability } 1 - \lambda$$

Jensen's Inequality

For convex f, Jensen's inequality is:

$$\mathbb{E}[f(x)] \geq f(\mathbb{E}[x])$$

A simple example:

$$x \text{ takes value } a \text{ with probability } \lambda$$
$$x \text{ takes value } b \text{ with probability } 1 - \lambda$$

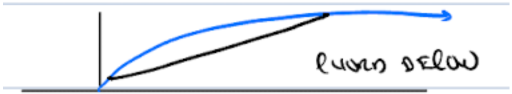
then,

$$\begin{aligned} \mathbb{E}[f(x)] =& \lambda f(a) + (1-\lambda) f(b) \\ f(\mathbb{E}[x]) =& f(\lambda a + (1-\lambda) b) = f(z) \end{aligned}$$

Jensen's inequality holds from the definition of convexity.

Concave and Convex

We say that a function g is **concave** if $-g$ is convex.



- $g(x) = \log(x)$ is concave since $g''(x) = -x^{-2} < 0$ on $(0, \infty)$.
- Jensen's inequality has the inequality reversed:

$$\mathbb{E}[g(x)] \leq g(\mathbb{E}[x]).$$

- What about $h(x) = ax + b$? it's convex and concave since $h''(x) = 0$.

End of Detour through Jensen's, Convexity, and Concavity.

Start of EM as Maximum Likelihood.

EM Algorithm as Maximum Likelihood

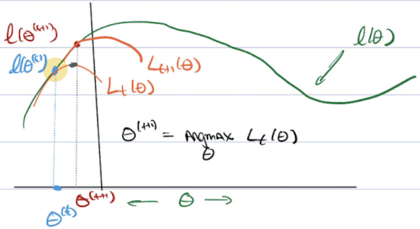
$$\ell(\theta) = \sum_{i=1}^{n} \log P(x^{(i)}; \theta).$$

we assume that

$$P(x; \theta) = \sum_{z} P(x, z, \theta) \text{ of GMM Latent Variable}$$

Here θ bundles all the paramters for convenience, and we are going to give a generic algorithm to maximize the likelihood for latent variable models.

Picture of EM Algorithm

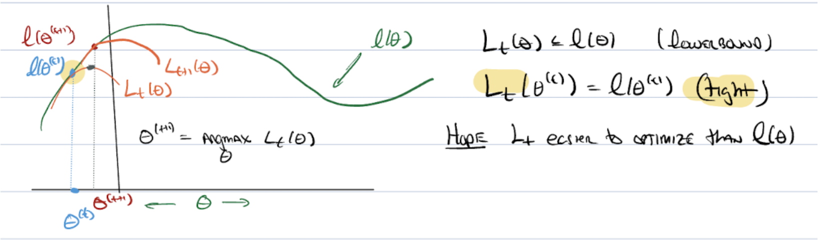


$L_t(\theta) \le \ell(\theta)$ (lowerbound)

$L_t(\theta^{(t)}) = \ell(\theta^{(t)})$ (tight)

<u>Hope</u> L_t easier to optimize than $\ell(\theta)$

Picture of EM Algorithm



Rough Algorithm.

- **E-Step** Given $\theta^{(t)}$ find a curve L_t
- **M-Step** Given L_t, set $\theta^{(t+1)} = \text{argmax}_\theta\, L_t(\theta)$.

How do we construct L_t?

We examine a single data point (and drop scripts). First a trick,

$$\log \sum_z P(x, z; \theta) = \log \sum_z \frac{Q(z) P(x, z; \theta)}{Q(z)}. \text{ for any } Q(z)$$

We pick $Q(z)$ s.t. $\sum_z Q(z) = 1$ and $Q(z) \geq 0$ then,

$$= \log \mathbb{E}_{z \sim Q(z)} \left[\frac{P(x, z; \theta)}{Q(z)} \right] \qquad \text{Def of } \mathbb{E}$$

How do we construct L_t?

We examine a single data point (and drop scripts). First a trick,

$$\log \sum_z P(x, z; \theta) = \log \sum_z \frac{Q(z)P(x, z; \theta)}{Q(z)}. \text{ for any } Q(z)$$

We pick $Q(z)$ s.t. $\sum_z Q(z) = 1$ and $Q(z) \geq 0$ then,

$$= \log \mathbb{E}_{z \sim Q(z)} \left[\frac{P(x, z; \theta)}{Q(z)} \right] \qquad \text{Def of } \mathbb{E}$$

$$\geq \mathbb{E}_{z \sim Q(z)} \left[\log \frac{P(x, z; \theta)}{Q(z)} \right] \qquad \text{Jensen, since log is concave.}$$

How do we construct L_t?

We examine a single data point (and drop scripts). First a trick,

$$\log \sum_z P(x, z; \theta) = \log \sum_z \frac{Q(z)P(x, z; \theta)}{Q(z)}. \text{ for any } Q(z)$$

We pick $Q(z)$ s.t. $\sum_z Q(z) = 1$ and $Q(z) \geq 0$ then,

$$
\begin{aligned}
&= \log \mathbb{E}_{z \sim Q(z)} \left[\frac{P(x, z; \theta)}{Q(z)} \right] && \text{Def of } \mathbb{E} \\
&\geq \mathbb{E}_{z \sim Q(z)} \left[\log \frac{P(x, z; \theta)}{Q(z)} \right] && \text{Jensen, since log is concave.} \\
&= \sum_z Q(z) \log \frac{P(x, z; \theta)}{Q(z)} && \text{Def of } \mathbb{E}
\end{aligned}
$$

How do we construct L_t?

We examine a single data point (and drop scripts). First a trick,

$$\log \sum_z P(x, z; \theta) = \log \sum_z \frac{Q(z) P(x, z; \theta)}{Q(z)}. \text{ for any } Q(z)$$

We pick $Q(z)$ s.t. $\sum_z Q(z) = 1$ and $Q(z) \geq 0$ then,

$$\begin{aligned}
&= \log \mathbb{E}_{z \sim Q(z)} \left[\frac{P(x, z; \theta)}{Q(z)} \right] && \text{Def of } \mathbb{E} \\
&\geq \mathbb{E}_{z \sim Q(z)} \left[\log \frac{P(x, z; \theta)}{Q(z)} \right] && \text{Jensen, since log is concave.} \\
&= \sum_z Q(z) \log \frac{P(x, z; \theta)}{Q(z)} && \text{Def of } \mathbb{E}
\end{aligned}$$

This lowerbound holds for *any* such choice of Q–a family of lower bounds. We can select Q *per point*.

How do we make it tight?

Select each Q to make tight for its term...

How do we make it tight?

Select each Q to make tight for its term...

$\log \frac{P(x, z; \theta)}{Q(z)} = c$ is constant wrt z, then Jensen is trivially an equality.

How do we make it tight?

Select each Q to make tight for its term...

$\log \frac{P(x, z; \theta)}{Q(z)} = c$ is constant wrt z, then Jensen is trivially an equality.

So what if $Q(z) = P(z \mid x; \theta)$ then

$$\log \frac{P(x, z; \theta)}{P(z \mid x; \theta)} = \log P(x; \theta)$$

If we examine the argument above, the only inequality is now equality so with this choice of Q we are tight!

> Note: $Q(z)$ *depends* on θ and x, so we will select a $Q^{(i)}(z)$ for each point $x^{(i)}$ for $i = 1, \ldots, n$.

ELBO!

We define the Evidence Lower Bound (ELBO) as:

$$\mathrm{ELBO}(x, Q, \theta) = \sum_{z} Q(z) \log \frac{P(x, z; \theta)}{Q(z)}.$$

So now, we've shown:

$$\ell(\theta) \geq \sum_{i=1}^{n} \mathrm{ELBO}(x^{(i)}, Q^{(i)}, \theta) \qquad \text{for any } Q^{(i)}$$

ELBO!

We define the Evidence Lower Bound (ELBO) as:

$$\mathrm{ELBO}(x, Q, \theta) = \sum_z Q(z) \log \frac{P(x, z; \theta)}{Q(z)}.$$

So now, we've shown:

$$\ell(\theta) \geq \sum_{i=1}^{n} \mathrm{ELBO}(x^{(i)}, Q^{(i)}, \theta) \qquad \text{for any } Q^{(i)}$$

$$\ell(\theta^{(t)}) = \sum_{i=1}^{n} \mathrm{ELBO}(x^{(i)}, Q^{(i)}, \theta^{(t)}) \qquad \text{for the choice of } Q^{(i)} \text{ above.}$$

ELBO!

We define the Evidence Lower Bound (ELBO) as:

$$\mathrm{ELBO}(x, Q, \theta) = \sum_{z} Q(z) \log \frac{P(x, z; \theta)}{Q(z)}.$$

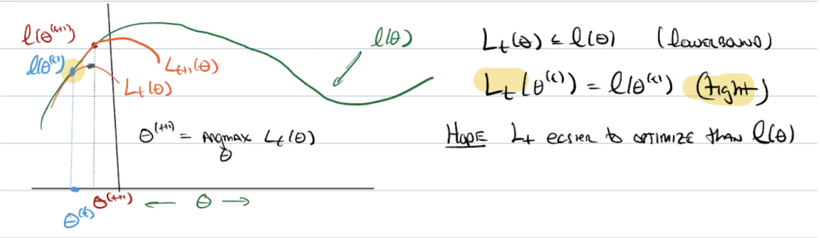
So now, we've shown:

$$\ell(\theta) \geq \sum_{i=1}^{n} \mathrm{ELBO}(x^{(i)}, Q^{(i)}, \theta) \qquad \text{for any } Q^{(i)}$$

$$\ell(\theta^{(t)}) = \sum_{i=1}^{n} \mathrm{ELBO}(x^{(i)}, Q^{(i)}, \theta^{(t)}) \qquad \text{for the choice of } Q^{(i)} \text{ above.}$$

We've shown lowerbound and tight, deriving the picture!

Wrap-up of EM!



- **E-Step** $Q^{(i)}(z) = P(z^{(i)} \mid x^{(i)}; \theta)$ for $i = 1, \dots, n$.
- **M-Step** $\theta^{(t+1)} = \text{argmax}_\theta L_t(\theta)$ in which

$$L_t(\theta) = \sum_{i=1}^{n} \text{ELBO}(x^{(i)}, Q^{(i)}, \theta).$$

Some comments:

- Why does this terminate? $\ell(\theta^{(t+1)}) \ge \ell(\theta^{(t)})$
- Is it globally optimal? Nope! See the picture.

Summary:

- We started with a "hard" clustering method in k-means, and solved with an alternating method.
- We generalized this to GMM and other "Latent" models with soft-clustering.
- We derived the EM algorithm in terms of MLE.