M-step details

$$\mathcal{L}(\theta, q) = \sum_i \sum_c q(t_i = c) \log \frac{p(x_i, t_i = c \mid \theta)}{q(t_i = c)}$$

$$= \sum_i \sum_c q(t_i = c) \log p(x_i, t_i = c \mid \theta)$$

$$- \sum_i \sum_c q(t_i = c) \log q(t_i = c)$$

Const w.r.t. θ

M-step details

$$\mathcal{L}(\theta, q) = \sum_i \sum_c q(t_i = c) \log \frac{p(x_i, t_i = c \mid \theta)}{q(t_i = c)}$$

$$= \sum_i \sum_c q(t_i = c) \log p(x_i, t_i = c \mid \theta)$$

$$\cancel{- \sum_i \sum_c q(t_i = c) \log q(t_i = c)}$$

^^ This is crossed out because we can just ignore it. When we optimise w.r.t. theta, there isn't any theta here! Its as good as a constant.

Const w.r.t. θ

M-step details

$$
\begin{aligned}
\mathcal{L}(\theta, q) &= \sum_i \sum_c q(t_i = c) \log \frac{p(x_i, t_i = c \mid \theta)}{q(t_i = c)} \\
&= \sum_i \sum_c q(t_i = c) \log p(x_i, t_i = c \mid \theta) \\
&\quad - \sum_i \sum_c q(t_i = c) \log q(t_i = c) \\
&= \mathbb{E}_q \log p(X, T \mid \theta) + \text{const}
\end{aligned}
$$

(Usually) concave function w.r.t. θ, easy to optimize

Concave ?
Why?
cos of log function?

Expectation Maximization algorithm

For k = 1, …

E-step

$$q^{k+1} = \arg\min_q \mathcal{KL}\left[q(T) \parallel p(T \mid X, \theta^k)\right]$$

$$\Leftrightarrow$$

$$q^{k+1}(t_i) = p(t_i \mid x_i, \theta^k)$$

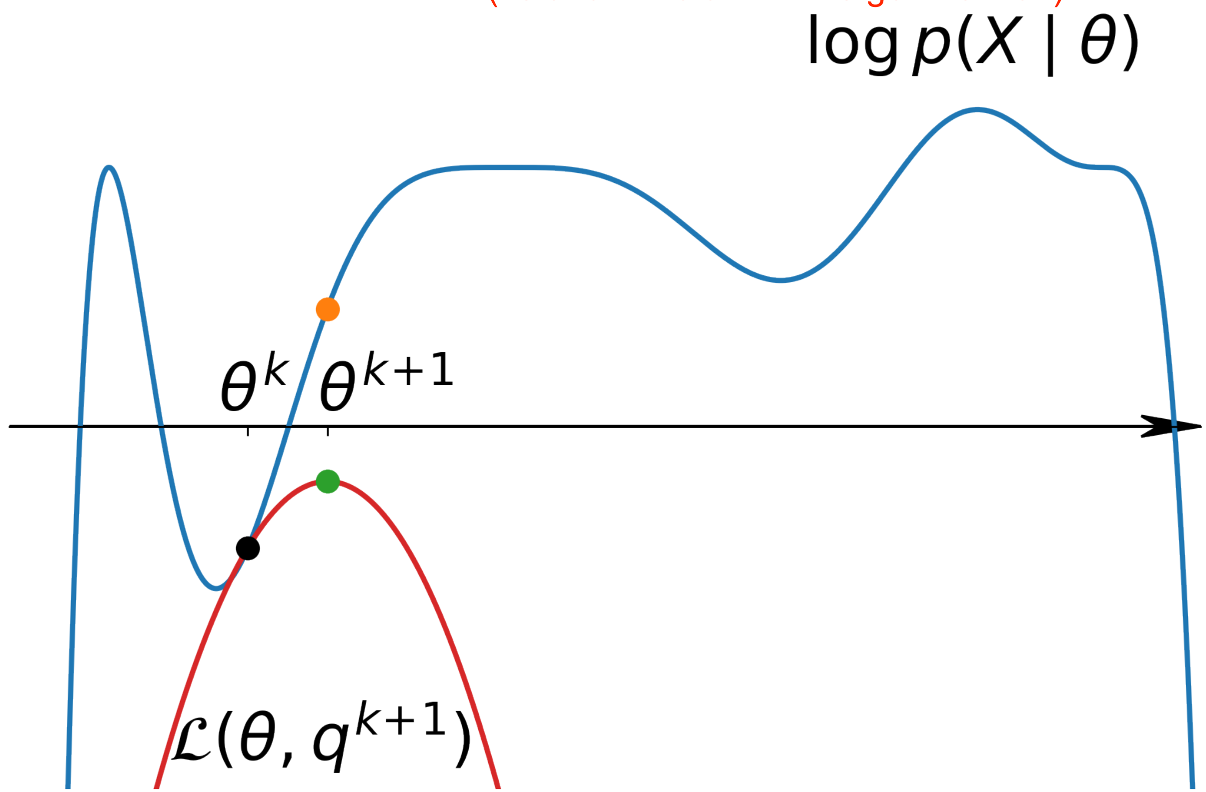
M-step

$$\theta^{k+1} = \arg\max_\theta \mathbb{E}_{q^{k+1}} \log p(X, T \mid \theta)$$

Convergence guaranties

$$\log p(X \mid \theta^{k+1}) \geq \boxed{\mathcal{L}(\theta^{k+1}, q^{k+1}) \geq \mathcal{L}(\theta^{k}, q^{k+1})} = \log p(X \mid \theta^{k})$$

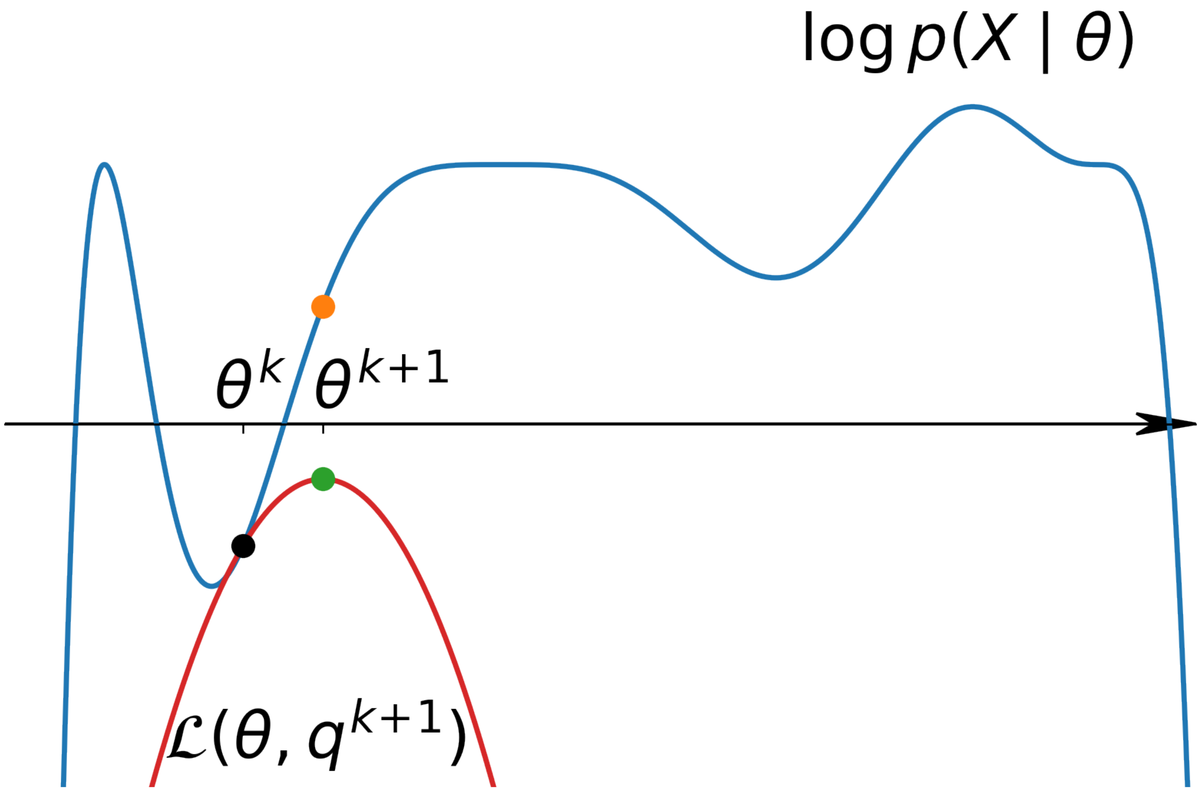
^ this part holds because it is the maximum of the L w.r.t theta.The left is the maximum of the lower bounds before, of which the right L was a candidate. (To attain theta k+1. we get the max).



Convergence guaranties

$$\log p(X \mid \theta^{k+1}) \geq \log p(X \mid \theta^{k})$$

(By transitivity of greater than or equal to operator)
lol



Convergence guaranties

$$\log p(X \mid \theta^{k+1}) \geq \log p(X \mid \theta^{k})$$

- On each iteration EM doesn't decrease the objective (good for debugging!)

i.e. MARGINAL LOG LIKELIHOOD WILL ALWAYS increase.
If it decreases, then we have a bug.

- Guarantied to converge to a local maximum (or saddle point)