Generative ML

CS771: Introduction to Machine Learning

Purushottam Kar

Recap of Last Lecture

Multivariate Gaussian/Laplacian distributions, their use as priors

MAP estimation turns out to be regularized optimization problems

Bayesian Learning: learning not one model but an entire distribution over models (the posterior probability distribution $\mathbb{P}\left[\mathbf{w} \mid \left\{\mathbf{x}^i, y^i\right\}\right]$)

Predictive Posterior: $\mathbb{P}\left[y \mid \mathbf{x}^t, \left\{\mathbf{x}^i, y^i\right\}\right] = \int_{\mathbb{R}^d} \mathbb{P}[y \mid \mathbf{w}, \mathbf{x}^t] \cdot \mathbb{P}\left[\mathbf{w} \mid \left\{\mathbf{x}^i, y^i\right\}\right] d\mathbf{w}$

Mostly inaccessible in closed form – need approximate methods

Conjugacy: nicely behaved likelihood-prior pairs where the posterior is available in closed form and is of the same family as the prior

Warning: *predictive posterior may still not be available in nice closed form*

Probabilistic Clustering: soft k-means

Generative Algorithms

ML algos that can learn dist. of the form $\mathbb{P}[\mathbf{x} \mid y]$ or $\mathbb{P}[\mathbf{x}, y]$ or $\mathbb{P}[\mathbf{x}]$

A slightly funny bit of terminology used in machine learning

- ***Discriminative Algorithms**: that only use $\mathbb{P}[y \mid \mathbf{x}]$ to do their stuff*
- ***Generative Algorithms**: that use $\mathbb{P}[\mathbf{x} \mid y]$, $\mathbb{P}[\mathbf{x}, y]$, or $\mathbb{P}[\mathbf{x}]$ etc to do their stuff*

Generative Algorithms have their advantages and disadvantages

- *More expensive: slower train times, slower test times, larger models*
- *An overkill: often, need only $\mathbb{P}[y \mid \mathbf{x}]$ to make predictions – disc. algos enough!*
- *More frugal: can work even if we have very less training data (e.g. RecSys)*
- *More robust: can work even if features corrupted e.g. some features missing*

A recent application of generative techniques (GANs etc) allows us to

- *Generate novel examples of a certain class of data points*
- *Generate more training examples for those classes as well!*

A very simple generative model

Given a few feature vectors (never mind labels for now) $\mathbf{x}^1, \ldots \mathbf{x}^n \in \mathbb{R}^d$

We wish to learn a probability distribution $\mathbb{P}[\cdot]$ with support over $\mathbb{R}^d$

This distribution should capture interesting properties about the data in a way that allows us to do things like generate similar-looking feature vectors etc

Let us try to learn a standard Gaussian as this distribution i.e. wish to learn $\boldsymbol{\mu} \in \mathbb{R}^d$ so that the distribution $\mathcal{N}(\boldsymbol{\mu}, I_d)$ explains this data well

One way is to look for a $\boldsymbol{\mu}$ *that achieves maximum likelihood i.e. MLE!!*

As before, assume that our feature vectors were independently generated

$\arg\max_{\boldsymbol{\mu} \in \mathbb{R}^d} \mathbb{P}[\mathbf{x}^1 \ldots \mathbf{x}^n \mid \boldsymbol{\mu}, I_d] = \arg\min_{\boldsymbol{\mu} \in \mathbb{R}^d} \sum_{i=1}^n \left\|\mathbf{x}^i - \boldsymbol{\mu}\right\|_2^2$ which, upon applying first order optimality, gives us $\widehat{\boldsymbol{\mu}}_{\text{MLE}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}^i$

We just learnt $\mathcal{N}(\widehat{\boldsymbol{\mu}}_{\text{MLE}}, I_d)$ as our generating dist. for data features!

A more powerful generative model

Suppose we are not satisfied with the above simple model

Suppose we wish to instead learn $\mathbf{\mu} \in \mathbb{R}^d$ as well as a $\sigma \geq \mathbf{0}$ so that the distribution $\mathcal{N}(\mathbf{\mu}, \sigma^2 \cdot I_d)$ explains the data well

Log likelihood function (be careful – cannot ignore any σ terms now)

$\arg \max_{\mathbf{\mu} \in \mathbb{R}^d, \sigma \geq 0} \ln \mathbb{P}[\mathbf{x}^1 \ldots \mathbf{x}^n \mid \mathbf{\mu}, \sigma^2] = \arg \min_{\mathbf{\mu} \in \mathbb{R}^d, \sigma \geq 0} f(\mathbf{\mu}, \sigma)$ where

$$f(\mathbf{\mu}, \sigma) = dn \ln \sigma + \frac{1}{2\sigma^2} \sum_{i=1}^n \left\| \mathbf{x}^i - \mathbf{\mu} \right\|_2^2$$

F.O. optimality w.r.t. $\mathbf{\mu}$ *i.e.* $\frac{\partial f}{\partial \mathbf{\mu}} = \mathbf{0}$ *gives us* $\hat{\mathbf{\mu}}_{\text{MLE}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}^i$

F.O. optimality w.r.t σ *i.e.* $\frac{\partial f}{\partial \sigma} = 0$ *gives us* $\hat{\sigma}_{\text{MLE}}^2 = \frac{1}{dn} \sum_{i=1}^n \left\| \mathbf{x}^i - \hat{\mathbf{\mu}}_{\text{MLE}} \right\|_2^2$

Since $\hat{\sigma}_{\text{MLE}}^2 \geq 0$ *this must be global opt. too!*

A still more powerful generative model

Suppose we wish to instead learn $\boldsymbol{\mu} \in \mathbb{R}^d$ as well as a $\Sigma \succcurlyeq \mathbf{0}$ so that the distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ explains the data well ($A \succcurlyeq \mathbf{0}$ notation for PSD)

$\arg\max_{\boldsymbol{\mu}\in\mathbb{R}^d, \Sigma\succcurlyeq 0} \ln \mathbb{P}[\mathbf{x}^1 \ldots \mathbf{x}^n \mid \boldsymbol{\mu}, \Sigma] = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^d, \Sigma\succcurlyeq 0} f(\boldsymbol{\mu}, \Sigma)$where

$$f(\boldsymbol{\mu}, \Sigma) = \frac{n}{2}\ln|\Sigma| + \frac{1}{2}\sum_{i=1}^n \left(\mathbf{x}^i - \boldsymbol{\mu}\right)^\top \Sigma^{-1}\left(\mathbf{x}^i - \boldsymbol{\mu}\right)$$

F.O.O. w.r.t. $\boldsymbol{\mu}$ *i.e.* $\frac{\partial f}{\partial \boldsymbol{\mu}} = \mathbf{0}$ *gives* $\left(\Sigma^{-1} + \left(\Sigma^{-1}\right)^\top\right)\sum_{i=1}^n \left(\mathbf{x}^i - \boldsymbol{\mu}\right) = \mathbf{0}$

Definitely $\frac{\partial f}{\partial \boldsymbol{\mu}} = \mathbf{0}$ when $\sum_{i=1}^n \left(\mathbf{x}^i - \boldsymbol{\mu}\right) = \mathbf{0}$ i.e. when $\hat{\boldsymbol{\mu}}_{\text{MLE}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}^i$

We may have $\frac{\partial f}{\partial \boldsymbol{\mu}} = \mathbf{0}$ *in some other funny cases even when* $\sum_{i=1}^n \left(\mathbf{x}^i - \boldsymbol{\mu}\right) \neq \mathbf{0}$ *which basically means there may be multiple optima for this problem*

F.O. optimality w.r.t Σ i.e. $\frac{\partial f}{\partial \Sigma} = \mathbf{0}\mathbf{0}^\top$ requires more work

A still more powerful generative model

For a square matrix $X \in \mathbb{R}^{d \times d}$, its trace $\mathrm{tr}(X) \triangleq \sum_{j=1}^{d} X_{ii}$ is defined as the sum of its diagonal elements

Easy result: if $\mathbf{a} \in \mathbb{R}^d$, *then* $\mathbf{a}^\top X \mathbf{a} = \mathrm{tr}(A^\top X)$ *where* $A = \mathbf{a}\mathbf{a}^\top \in \mathbb{R}^{d \times d}$

Not so easy result: if A is a constant matrix, then $\frac{\partial \mathrm{tr}(A^\top X)}{\partial X} = A$

Recall: dims of derivs always equal those of quantity w.r.t which deriv is taken

Let us denote $\Lambda \triangleq \Sigma^{-1}$ for convenience

New expression: $f(\boldsymbol{\mu}, \Sigma) = \frac{n}{2} \ln|\Sigma| + \frac{1}{2} \sum_{i=1}^{n} \mathrm{tr}(\Lambda^\top S^i)$ where $S^i = (\mathbf{x}^i - \boldsymbol{\mu})(\mathbf{x}^i - \boldsymbol{\mu})^\top$

A still more powerful generative model

For any $A, B, C \in \mathbb{R}^{d \times d}$ we have the following

Symmetry: $\text{tr}(A^\top B) = \text{tr}(B^\top A)$

Linearity: $\text{tr}(A^\top B) + \text{tr}(A^\top C) = \text{tr}\left(A^\top(B + C)\right)$

New expression: $f(\boldsymbol{\mu}, \Sigma) = \frac{n}{2}\ln|\Sigma| + \frac{1}{2}\text{tr}(S^\top \Lambda)$ where $S = \sum_{i=1}^n S^i$

$$\frac{\partial \ln|\Sigma|}{\partial \Sigma} = \frac{1}{|\Sigma|} \cdot \frac{\partial |\Sigma|}{\partial \Sigma} = \frac{1}{|\Sigma|} \cdot \left(|\Sigma| \cdot (\Sigma^{-1})^\top\right) = (\Sigma^{-1})^\top = \Sigma^{-1} \text{ (assume symm)}$$

$$\frac{\partial \text{tr}\left(S^\top \Lambda\right)}{\partial \Sigma} = \frac{\partial \Lambda}{\partial \Sigma} \cdot \frac{\partial \text{tr}\left(S^\top \Lambda\right)}{\partial \Lambda} = \frac{\partial \Sigma^{-1}}{\partial \Sigma} \cdot \frac{\partial \text{tr}\left(S^\top \Lambda\right)}{\partial \Lambda} = -\Sigma^{-2} S$$

F.O.O. w.r.t. Σ i.e. $\frac{\partial f}{\partial \Sigma} = \mathbf{0}\mathbf{0}^\top$ gives $\frac{n}{2} \cdot \Sigma^{-1} - \frac{1}{2}\Sigma^{-2} S = \mathbf{0}\mathbf{0}^\top$ which gives

$$\hat{\Sigma}_{\text{MLE}} = \frac{1}{n} \cdot S = \frac{1}{n}\sum_{i=1}^n \left(\mathbf{x}^i - \boldsymbol{\mu}\right)\left(\mathbf{x}^i - \boldsymbol{\mu}\right)^\top$$

Since $\hat{\Sigma}_{\text{MLE}} \succcurlyeq 0$ *as well as symmetric, this must be the global optimum!*

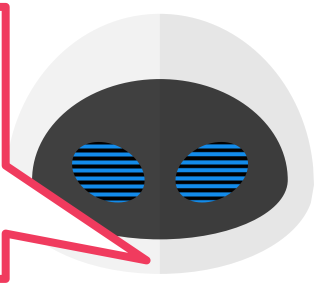
A still more powerful generative model

For any $A, B, C \in \mathbb{R}^{d \times d}$ we have the foll

Symmetry: $\mathrm{tr}(A^\top B) = \mathrm{tr}(B^\top A)$

Linearity: $\mathrm{tr}(A^\top B) + \mathrm{tr}(A^\top C) = \mathrm{tr}\left(A^\top (B + C)\right)$

See “The Matrix Cookbook” (reference section on course webpage) for these results



New expression: $f(\boldsymbol{\mu}, \Sigma) = \frac{n}{2} \ln|\Sigma| + \frac{1}{2} \mathrm{tr}(S^\top \Lambda)$ where $S = \sum_{i=1}^n S^i$

$\frac{\partial \ln|\Sigma|}{\partial \Sigma} = \frac{1}{|\Sigma|} \cdot \frac{\partial |\Sigma|}{\partial \Sigma} = \frac{1}{|\Sigma|} \cdot \left(|\Sigma| \cdot (\Sigma^{-1})^\top\right) = (\Sigma^{-1})^\top = \Sigma^{-1}$ (assume symm)

$$\frac{\partial \mathrm{tr}\left(S^\top \Lambda\right)}{\partial \Sigma} = \frac{\partial \Lambda}{\partial \Sigma} \cdot \frac{\partial \mathrm{tr}\left(S^\top \Lambda\right)}{\partial \Lambda} = \frac{\partial \Sigma^{-1}}{\partial \Sigma} \cdot \frac{\partial \mathrm{tr}\left(S^\top \Lambda\right)}{\partial \Lambda} = -\Sigma^{-2} S$$

F.O.O. w.r.t. Σ i.e. $\frac{\partial f}{\partial \Sigma} = \mathbf{0}\mathbf{0}^\top$ gives $\frac{n}{2} \cdot \Sigma^{-1} - \frac{1}{2} \Sigma^{-2} S = \mathbf{0}\mathbf{0}^\top$ which gives

$$\hat{\Sigma}_{\text{MLE}} = \frac{1}{n} \cdot S = \frac{1}{n} \sum_{i=1}^n \left(\mathbf{x}^i - \boldsymbol{\mu}\right)\left(\mathbf{x}^i - \boldsymbol{\mu}\right)^\top$$

Since $\hat{\Sigma}_{\text{MLE}} \succcurlyeq 0$ as well as symmetric, this must be the global optimum!

MAP, Bayesian Generative Models?

The previous techniques allow us to learn the parameters of a Gaussian distribution (either $\mathbf{\mu}$ or $\mathbf{\mu}, \sigma^2$ or $\mathbf{\mu}, \Sigma$) that offer the highest likelihood of observed data features by computing the MLE

We can incorporate priors over $\mathbf{\mu}$ (e.g. Gaussian, Laplacian), priors over σ^2 (e.g. inverse Gamma dist. which has support only over non-negative numbers) and Σ (e.g. inverse Wishart dist. which has support only over PSD matrices) and computer the MAP

We can also perform full-blown Bayesian inference by computing posterior distributions over quantities such as $\mathbf{\mu}, \sigma^2, \Sigma$ – calculations involving predictive posterior get messy – beyond scope of CS771

However, can make generative models more powerful in other ways too that are much less expensive

Still more powerful generative model?

Suppose we are concerned that a single Gaussian cannot capture all the variations in our data

Just as in LwP when we realized sometimes, a single prototype not enough

Can we learn 2 (or more) Gaussians to represent our data instead?

Such a generative model is often called a mixture *of Gaussians*

The Expectation Maximization (EM) algorithm is a very powerful technique for performing this and several other tasks

Soft clustering, learning Gaussian mixture models (GMM)

Robust learning, Mixed Regression

Also underlies more powerful variational *algorithms such as VAE*

Learning a Mixture of Two Gaussians

We suspect that instead of one Gaussian, two Gaussians are involved in generating our feature vectors

For sake of simplicity, let them be $\mathcal{N}(\boldsymbol{\mu}^1, I_d)$ *and* $\mathcal{N}(\boldsymbol{\mu}^2, I_d)$

Each of these is called a component *of this GMM*

Covariance matrices, more than two components can also be incorporated

Since we are unsure which data point came from which component, we introduce a *latent variable* $z_i \in \{1,2\}$ per data point to denote this

The English word "latent" means hidden or dormant or concealed

Nice name since this variable describes something that was hidden from us

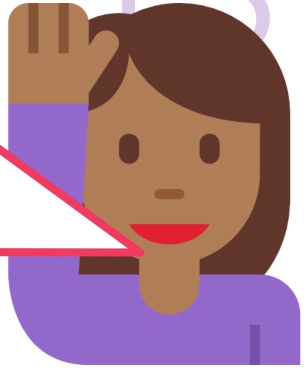
These latent variables may seem similar to the one we used in (soft) k-means

Not an accident – the connections will be clear soon!

Latent variables can be discrete or continuous

Lea

This means that if someone tells us that $z_i = 1$ this means that the first Gaussian is responsible for that data point and consequently, the likelihood expression is $\mathbb{P}\left[\mathbf{x}^i \mid z_i = 1, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \mathcal{N}\left(\mathbf{x}^i; \boldsymbol{\mu}^1\right)$. Similarly, if someone tells us that $z_j = 2$ this means that the second Gaussian is responsible for that data point and the likelihood expression is $\mathbb{P}\left[\mathbf{x}^j \mid z_j = 2, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \mathcal{N}\left(\mathbf{x}^j; \boldsymbol{\mu}^2\right)$.



We

gen

- *For sake of simplicity, let them be $\mathcal{N}(\boldsymbol{\mu}^1, I_d)$ and $\mathcal{N}(\boldsymbol{\mu}^2, I_d)$*
- *Each of these is called a* component *of this GMM*
- *Covariance matrices, more than two components can also be incorporated*

Since we are unsure which data point came from which component, we introduce a *latent variable* $z_i \in \{1,2\}$ per data point to denote this

- *The English word "latent" means hidden or dormant or concealed*
- *Nice name since this variable describes something that was hidden from us*
- *These latent variables may seem similar to the one we used in (soft) k-means*
- *Not an accident – the connections will be clear soon!*
- *Latent variables can be discrete or continuous*

MLE with Latent Variables

We wish to obtain the maximum (log) likelihood models i.e.

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln \mathbb{P}\left[\mathbf{x}^i \mid \boldsymbol{\mu}^1,\boldsymbol{\mu}^2\right]$$

Since we do not know the values of latent variables, use brute force way to introduce them using law of total probability

Recall we did the same thing while deriving predictive posterior expression

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\mu}^1,\boldsymbol{\mu}^2\right]\right)$$

Very difficult optimization problem – NP-hard in general

However, two heuristics exist which work reasonably well in practice

Also theoretically sound if data is "nice" (details in a learning theory course)

Heuristic 1: Alternating Optimization

Convert the original optimization problem

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]\right)$$

to a double opt. (assume $\mathbb{P}[z_i \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2] =$ const. for sake of simplicity

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d, z_i\in\{1,2\}} \sum_{i=1}^n \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

In several ML problems with latent vars, although the above optimization problem (still) difficult, following two problems are easy

Step 1*: Fix $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2 \in \mathbb{R}^d$ and update latent variables z_i to their optimal values*

$$\arg\max_{z_i\in\{1,2\}} \sum_{i=1}^n \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \arg\max_{z_i\in\{1,2\}} \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

Step 2*: Fix latent variables z_i and update $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2 \in \mathbb{R}^d$ to their optimal values*

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

Heuristic 1: Alternating Optimization

Convert the original optimi

Keep alternating between step 1 and step 2 till you are tired or till the process has converged!

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]\right)$$

to a double opt. (assume $\mathbb{P}[z_i \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2] =$ const. for sake of simplicity

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d, z_i\in\{1,2\}} \sum_{i=1}^n \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

In seve

optimi a

Step

arg $_{z_i\in\{1,2\}}$ $_{z_i\in\{1,2\}}$ $]$

The most important difference between the original and the new problem is that original has a **sum of log of sum** which is very difficult to optimize whereas the new problem gets rid of this and looks simply like a **MLE** problem. We know how to solve MLE problems very easily!

Step 2*: Fix latent variables z_i and update $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2 \in \mathbb{R}^d$ to their optimal values*

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

Heuristic 1 at Work

As discussed before, we assume a mixture of two Gaussians

$$\mathbb{P}\left[\mathbf{x}^i \mid z_i = 1, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \mathcal{N}\left(\mathbf{x}^i; \boldsymbol{\mu}^1\right) \textit{ and } \mathbb{P}\left[\mathbf{x}^j \mid z_j = 2, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \mathcal{N}\left(\mathbf{x}^j; \boldsymbol{\mu}^2\right)$$

Step 1 becomes

$$\arg \max_{z_i \in \{1,2\}} \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \arg \min_{z_i \in \{1,2\}} \left\|\mathbf{x}^i - \boldsymbol{\mu}^{z_i}\right\|_2^2$$

Step 2 becomes

$$\arg \max_{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2 \in \mathbb{R}^d} \sum_{i=1}^n \ln \mathbb{P}\left[\mathbf{x}^i \mid z_i, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

$$= \arg \min_{\boldsymbol{\mu}^1 \in \mathbb{R}^d} \sum_{i: z_i = 1} \left\|\mathbf{x}^i - \boldsymbol{\mu}^1\right\|_2^2 + \arg \min_{\boldsymbol{\mu}^2 \in \mathbb{R}^d} \sum_{i: z_i = 2} \left\|\mathbf{x}^i - \boldsymbol{\mu}^2\right\|_2^2$$

Thus, $\boldsymbol{\mu}^1 = \frac{1}{n_1} \sum_{i: z_i = 1} \mathbf{x}^i$ *and* $\boldsymbol{\mu}^2 = \frac{1}{n_2} \sum_{i: z_i = 2} \mathbf{x}^i$ *where* n_c *is the number of data points for which we have* $z_i = c$

Repeat!

Heuristic 1 at Work

Isn't this like the k-means algorithm?

As discussed before, we assume a mixture of two Gaussians

$$\mathbb{P}\left[\mathbf{x}^i \mid z_i = 1, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \mathcal{N}\left(\mathbf{x}^i; \boldsymbol{\mu}^1\right) \textit{ and } \mathbb{P}\left[\mathbf{x}^i \mid z_i = 2, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right] = \mathcal{N}\left(\mathbf{x}^i; \boldsymbol{\mu}^2\right)$$

Step 1 becomes

$$\arg\max_{z_i \in \{1,2\}} \ln \mathbb{P}\left[\mathbf{x}\right. \quad z_i \in \{1,2\}$$

Not just "like" – this **is** the k-means algorithm! This means that the k-means algorithm is one heuristic way to compute an MLE which is difficult to compute directly!

Step 2 becomes

$$\arg\max_{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2 \in \mathbb{R}^d} \sum_{i=1}^{n}$$

Indeed! Notice that even here, instead of choosing just one value of the latent variables z_i at each time step, we can instead use a distribution over their support $\{1,2\}$

$$= \arg\min_{\boldsymbol{\mu}^1 \in \mathbb{R}^d} \sum_{i:z_i=1} \left\|\mathbf{x}^i - \boldsymbol{\mu}^1\right\|_2^2 + \arg\min \sum \left\|\mathbf{x}^i - \boldsymbol{\mu}^2\right\|^2$$

Thus, $\boldsymbol{\mu}^1 = \frac{1}{n_1} \sum_{i:z_i=1} \mathbf{x}^i$ *and* $\boldsymbol{\mu}^2 =$

data points for which we have $z_i = c$

I have a feeling that the second heuristic will also give us something we have already studied!

Repeat!

Heuristic 2: Expectation Maximization

Original Prob: $\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]\right)$

Step 1 (E Step) Consists of two sub-steps

Step 1.1 *Assume our current model estimates are* $\boldsymbol{\mu}^1 = \mathbf{p}, \boldsymbol{\mu}^2 = \mathbf{q}$

Use the current models to ascertain how likely are different values of z_i for the i-th data point i.e. compute $q_c^i = \mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}\right]$ for both $c \in \{1,2\}$

Step 1.2 *Use weights* q_c^i *to set up a new objective function*

As before, assume $\mathbb{P}[z_i \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2] =$ constant for sake of simplicity

$\sum_{i=1}^n \sum_{c\in\{1,2\}} q_c^i \cdot \ln \mathbb{P}\left[\mathbf{x}^i, \mid z_i = c, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$

Step 2 (M Step) Maximize the new obj. fn. to get new models

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \sum_{c\in\{1,2\}} q_c^i \cdot \ln \mathbb{P}\left[\mathbf{x}^i, \mid z_i = c, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

Repeat!

Heuristic 2: Expectation Maximization

Original Prob: $\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]\right)$

Step 1 (E Step) Consists of two sub-steps

Step 1.1 *Assume our current model estimates are* $\boldsymbol{\mu}^1 = \mathbf{p}, \boldsymbol{\mu}^2 = \mathbf{q}$

Use the c he i ta

point i.e.

Step 1.2 *U*

As before

$\sum_{i=1}^n \sum_{c\in\{1,2\}} q_c^i \cdot \ln \mathbb{P}\left[\mathbf{x}^i, \mid z_i = c, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$

Yet again, the new problem gets rid of the treacherous "sum of log of sum" terms which are difficult to optimize. The new problem instead looks simply like a ***weighted*** **MLE** problem with weights q_c^i and we know how to solve MLE problems very easily!

Step 2 (M Step) Maximize the new obj. fn. to get new models

$$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum_{i=1}^n \sum_{c\in\{1,2\}} q_c^i \cdot \ln \mathbb{P}\left[\mathbf{x}^i, \mid z_i = c, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]$$

Repeat!

Derivation of the E Step

Let $\boldsymbol{\theta}$ denote the models $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2$ to avoid clutter. Also let $\boldsymbol{\theta}^0$ denote our current estimate of the model

Just need to see derivation for a single point, say the i-th point

$$\ln \mathbb{P}\left[\mathbf{x}^i \mid \boldsymbol{\theta}\right] = \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\theta}\right]\right)$$

$$= \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \boldsymbol{\theta}^0\right] \cdot \frac{\mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\theta}\right]}{\mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \boldsymbol{\theta}^0\right]}\right)$$

$$\geq \sum_{c\in\{1,2\}} \mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \boldsymbol{\theta}^0\right] \cdot \ln\left(\frac{\mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\theta}\right]}{\mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \boldsymbol{\theta}^0\right]}\right)$$

$$= \sum_{c\in\{1,2\}} q_c^i \cdot \ln\left(\frac{\mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\theta}\right]}{q_c^i}\right)$$

$$= \sum_{c\in\{1,2\}} q_c^i \cdot \ln \mathbb{P}\left[\mathbf{x}^i,\ \ z_i = c \mid \boldsymbol{\theta}\right] + e_i$$

Jensen's inequality tells us that $f(\mathbb{E}X) \leq \mathbb{E}[f(X)]$ for any convex function. We used the fact that $\ln(\cdot)$ is a concave function and so the inequality reverses since every concave function is the negative of a convex function

er

our current estimate of the model

Just need to see derivation for a single point, s

Law of total probability

$$\ln \mathbb{P}\left[\mathbf{x}^i \mid \boldsymbol{\theta}\right] = \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\theta}\right]\right)$$

$$= \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \boldsymbol{\theta}^0\right] \cdot \frac{\mathbb{P}\left[\mathbf{x}^i, z_i{=}c \mid \boldsymbol{\theta}\right]}{\mathbb{P}\left[z_i{=}c \mid \mathbf{x}^i{,}\boldsymbol{\theta}^0\right]}\right)$$

Simply multiply and divide by the same term

$$\geq \sum_{c\in\{1,2\}} \mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \boldsymbol{\theta}^0\right] \cdot \ln\left(\frac{\mathbb{P}\left[\mathbf{x}^i, z_i{=}c \mid \boldsymbol{\theta}\right]}{\mathbb{P}\left[z_i{=}c \mid \mathbf{x}^i{,}\boldsymbol{\theta}^0\right]}\right)$$

Jensen's inequality

$$= \sum_{c\in\{1,2\}} q_c^i \cdot \ln\left(\frac{\mathbb{P}\left[\mathbf{x}^i, z_i{=}c \mid \boldsymbol{\theta}\right]}{q_c^i}\right)$$

Just renaming $q_c^i = \mathbb{P}\left[z_i = c \mid \mathbf{x}^i, \boldsymbol{\theta}^0\right]$

$$= \sum_{c\in\{1,2\}} q_c^i \cdot \ln \mathbb{P}\left[\mathbf{x}^i, \ \ z_i = c \mid \boldsymbol{\theta}\right] + e_i$$

e_i is a constant that does not depend on $\boldsymbol{\theta}$

The EM Algorithm

If we instantiate the EM algorithm with the GMM likelihoods, we will recover the soft k-means algorithm

Thus, the soft k-means algorithm is yet another heuristic way (the k-means algo is the other) to compute an MLE which is difficult to compute directly!

The EM algorithm has pros and cons over alternating optimization

Con*: EM is usually more expensive to execute than alternating optimization*

Pro*: EM will ensures that objective value of the original problem i.e.*

$$\underset{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d}{\arg\max} \sum_{i=1}^n \ln\left(\sum_{c\in\{1,2\}} \mathbb{P}\left[\mathbf{x}^i, z_i = c \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2\right]\right)$$

… always keeps going up at every iteration – monotonic progress!!

May include details in the course notes (not very difficult)

However, no guarantee that we will ever reach the global maximum

May converge to, and get stuck at, a local maximum instead

Note: assumptions such as $\mathbb{P}[z_i \mid \boldsymbol{\mu}^1, \boldsymbol{\mu}^2] = \text{const}$ are made for sake of simplicity only. Can execute EM perfectly well without making these assumptions a well. However, updates get more involved – be careful not to make mistakes

If we instantiate the EM algorithm with the GMM likelihoods, we will recover the sof

Thus, the soft ... *ay (the k-means algo is the oth* ... *ompute directly!*

The EM algorit ... ptimization

Con: EM is usu ... *ing optimization*

Pro: EM will e ... *blem i.e.*

$\arg\max_{\boldsymbol{\mu}^1,\boldsymbol{\mu}^2\in\mathbb{R}^d} \sum$

... always kee ... *ogress!!*

May include d

However, no g ... *naximum*

May converge ... *d*

EM for GMM

1. Initialize means $\{\boldsymbol{\mu}^c\}_{c=1\ldots C}$
2. For $i \in [n]$, update π^i_c using $\{\boldsymbol{\mu}^c\}$
   1. Let $p^i_c = \exp\left(-\frac{\left\|\mathbf{x}^i - \boldsymbol{\mu}^c\right\|_2^2}{2}\right)$
   2. Let $q^i_c = \frac{p^i_c}{\sum_{c=1}^C p^i_c}$ (normalize)
3. Let $n_c = \sum_{i=1}^n q^i_c$
4. Update $\boldsymbol{\mu}^c = \frac{1}{n_c}\sum_{i=1}^n q^i_c \cdot \mathbf{x}^i$
5. Repeat until convergence

The Q Function

Let $Q_t(\boldsymbol{\theta})$ = … new

objective fu…

The EM alg… …ring

the E-step a…

$\boldsymbol{\theta}^{t+1} = \arg$…

We have al…

Can also s…

Some indi…

Alt. Opt. ins… …form

$Q_t(\boldsymbol{\theta}) = \sum_{i=}^{n}$ … $z_i \in \{1,2\}$

The Generic EM Algorithm

1. Initialize model $\boldsymbol{\theta}^0$
2. For every latent variable z_i and every possible value $z \in \mathcal{Z}$ it could take, compute
$$q_z^{i,t} = \mathbb{P}\left[z_i = z \mid \mathbf{x}^i, \boldsymbol{\theta}^t\right]$$
3. Compute the Q-function
$$Q_t(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{z \in \mathcal{Z}} q_z^{i,t} \ln \mathbb{P}\left[x^i, z^i = z \mid \boldsymbol{\theta}\right]$$
4. Update $\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q_t(\boldsymbol{\theta})$
5. Repeat until convergence

The Q Function

Let $Q_t(\boldsymbol{\theta}) =$ new

objective fu

The EM alg ring

the E-step a

$\boldsymbol{\theta}^{t+1} = \arg$

We have al

Can also s

Some indi

Alt. Opt. ins orm

$Q_t(\boldsymbol{\theta}) = \sum_{i=}^{n}$ $z_i \in \{1,2\}$

The Generic EM Algorithm

1. Initialize model $\boldsymbol{\theta}^0$
2. For every latent variable z_i and every possible value $z \in \mathcal{Z}$ it could take, compute
$$q_z^{i,t} = \mathbb{P}\left[z_i = z \mid \mathbf{x}^i, \boldsymbol{\theta}^t\right]$$
3. Compute the Q-function
$$Q_t(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{z \in \mathcal{Z}} q_z^{i,t} \ln \mathbb{P}\left[x^i, z^i = z \mid \boldsymbol{\theta}\right]$$
4. Update $\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q_t(\boldsymbol{\theta})$
5. Repeat until convergence

A pictorial depiction of the EM

The Q_{t}-curves always lie below the red curve

$$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM

The Q_{t}-curves always lie below the red curve

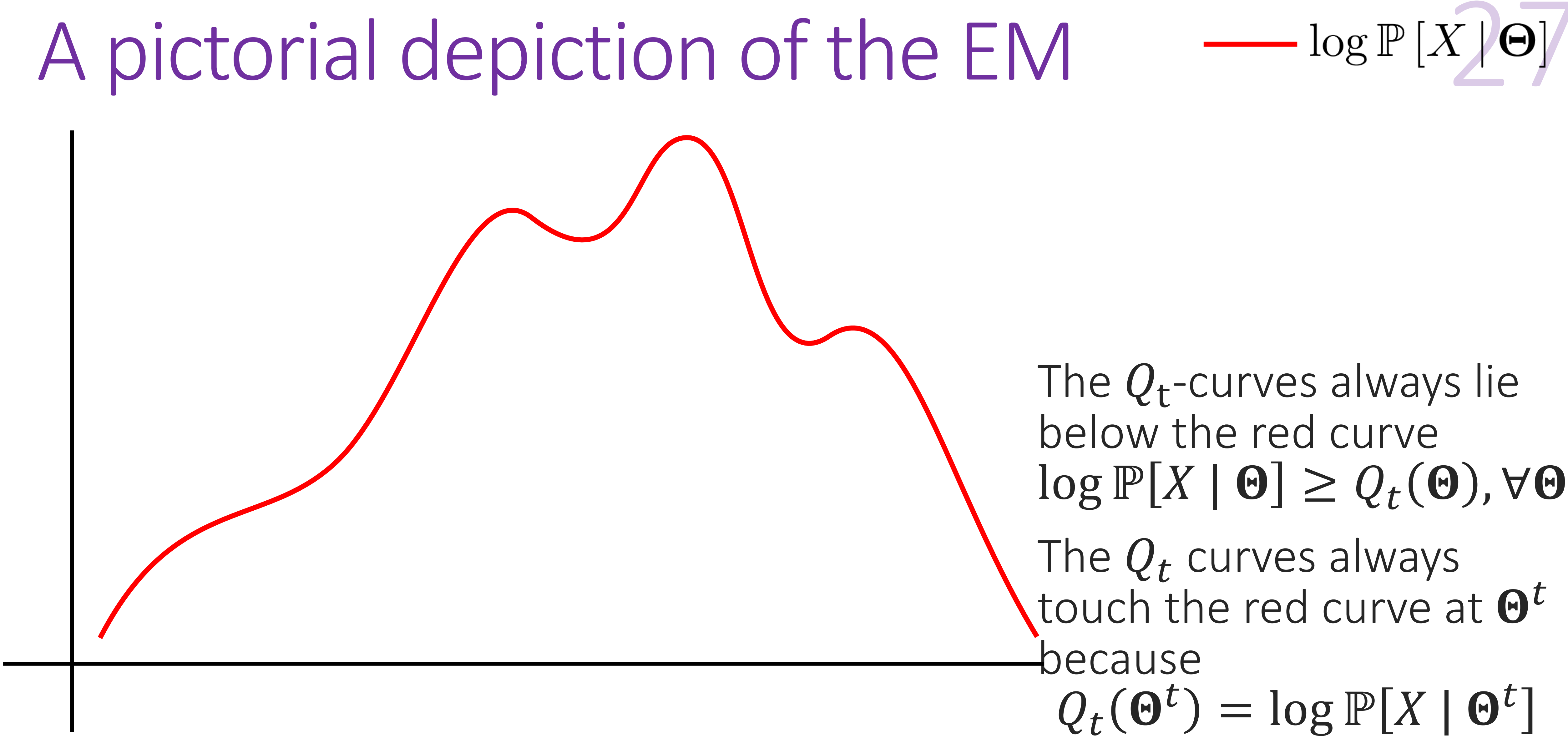
$$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

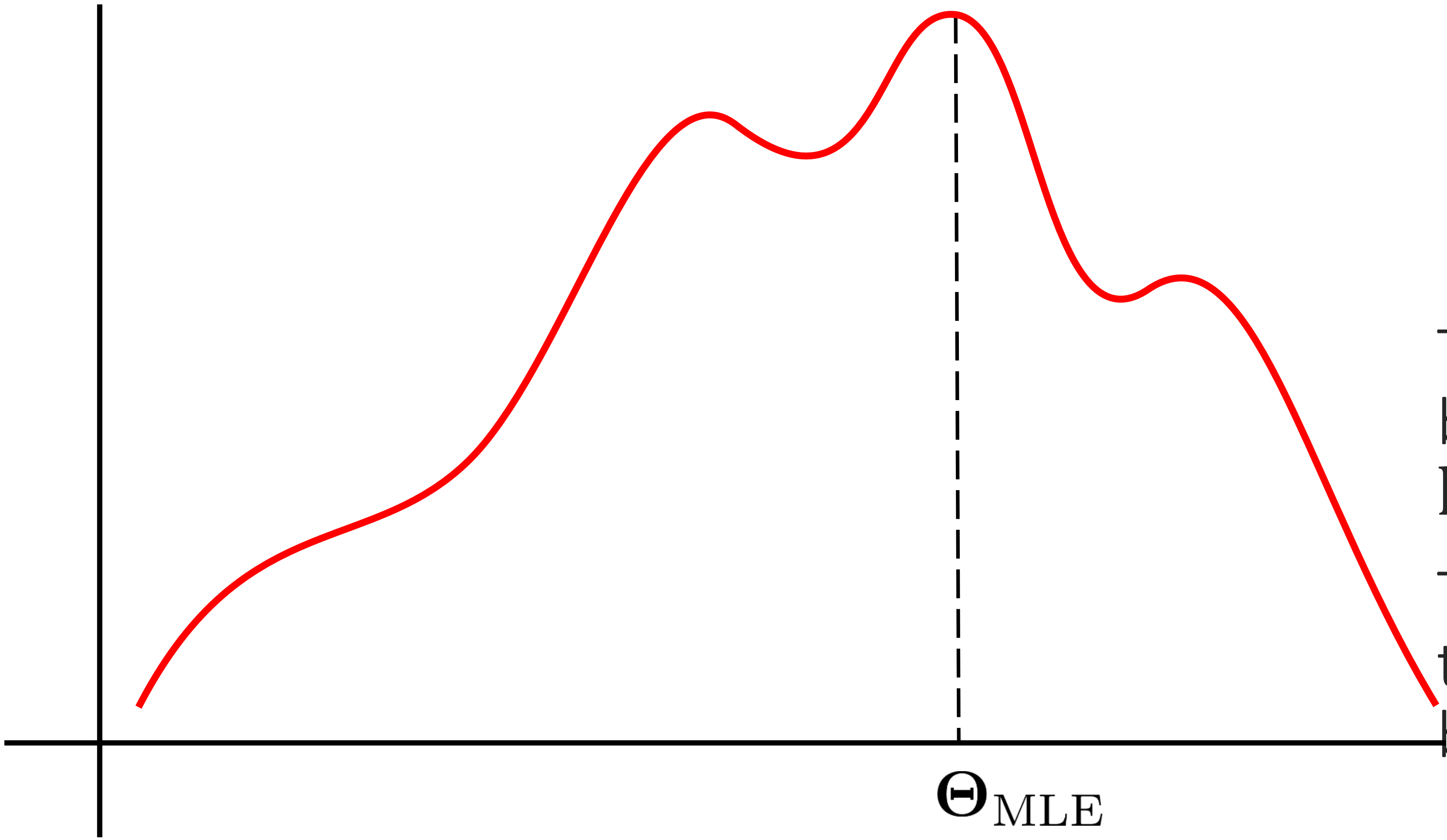
$$\log \mathbb{P}[X \mid \boldsymbol{\Theta}] \geq Q_t(\boldsymbol{\Theta}), \forall \boldsymbol{\Theta}$$

The Q_t curves always touch the red curve at $\boldsymbol{\Theta}^t$ because

$$Q_t(\boldsymbol{\Theta}^t) = \log \mathbb{P}[X \mid \boldsymbol{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

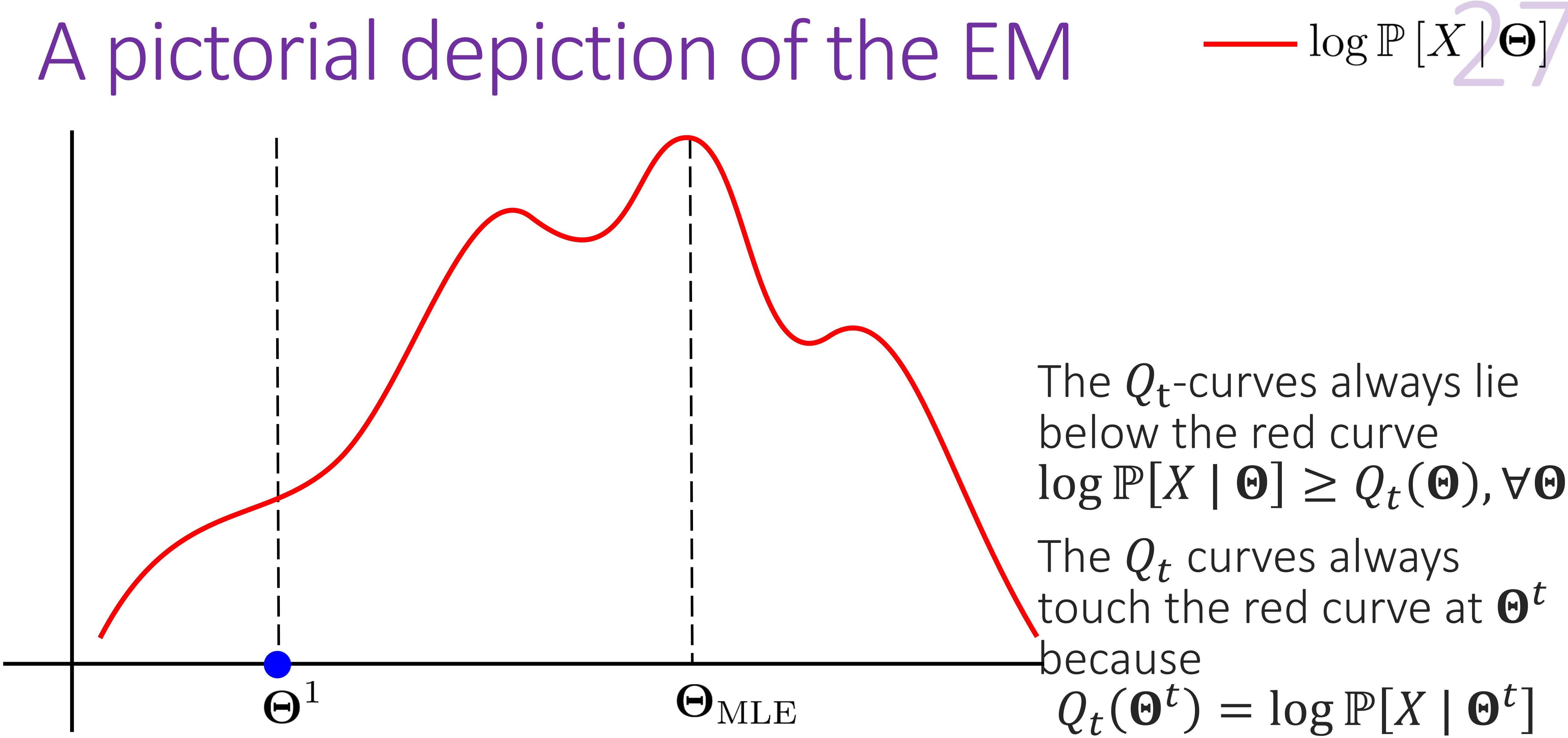
$$\log \mathbb{P}[X \mid \boldsymbol{\Theta}] \geq Q_t(\boldsymbol{\Theta}), \forall \boldsymbol{\Theta}$$

The Q_t curves always touch the red curve at $\boldsymbol{\Theta}^t$ because

$$Q_t(\boldsymbol{\Theta}^t) = \log \mathbb{P}[X \mid \boldsymbol{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

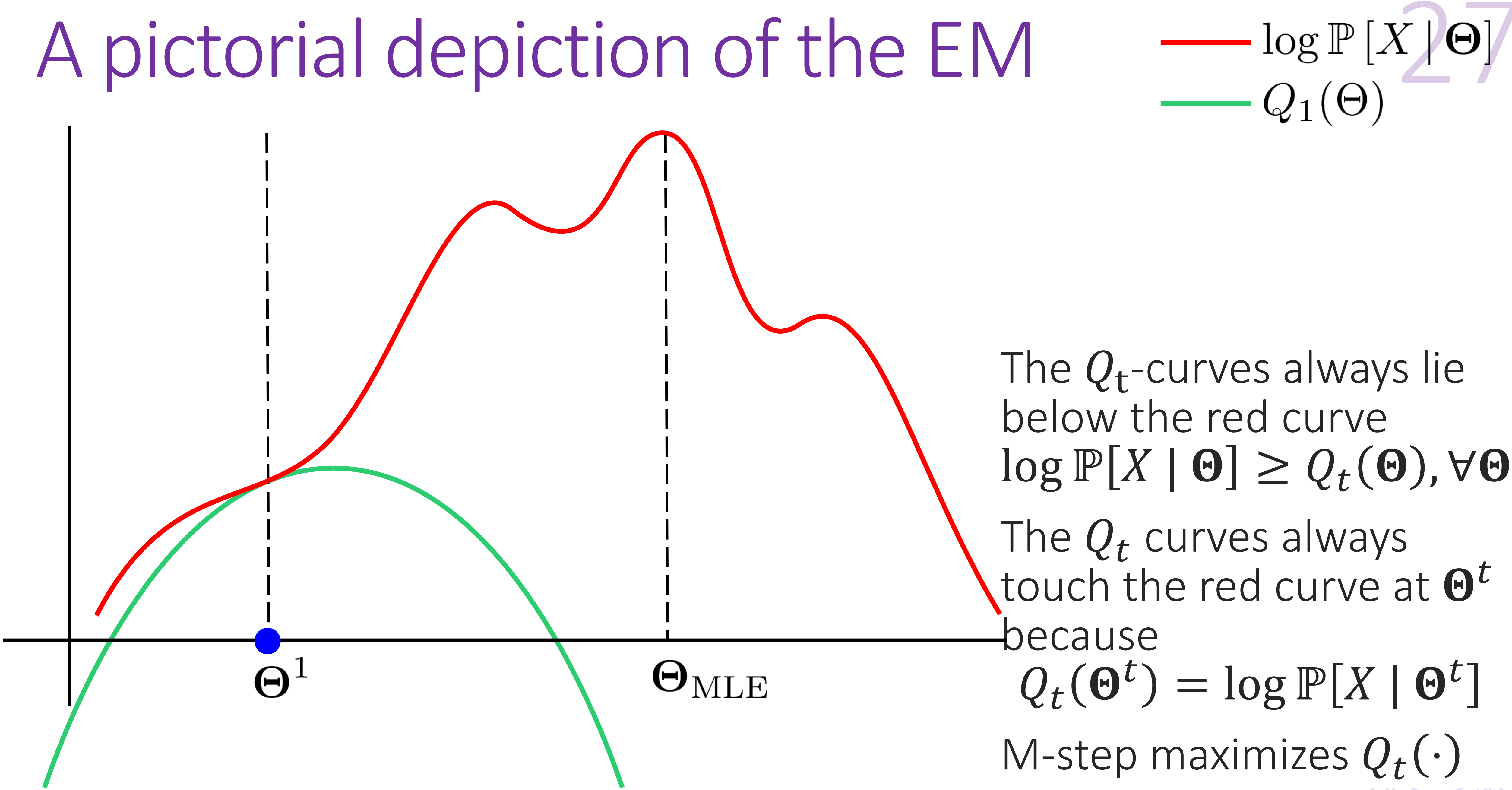
$$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

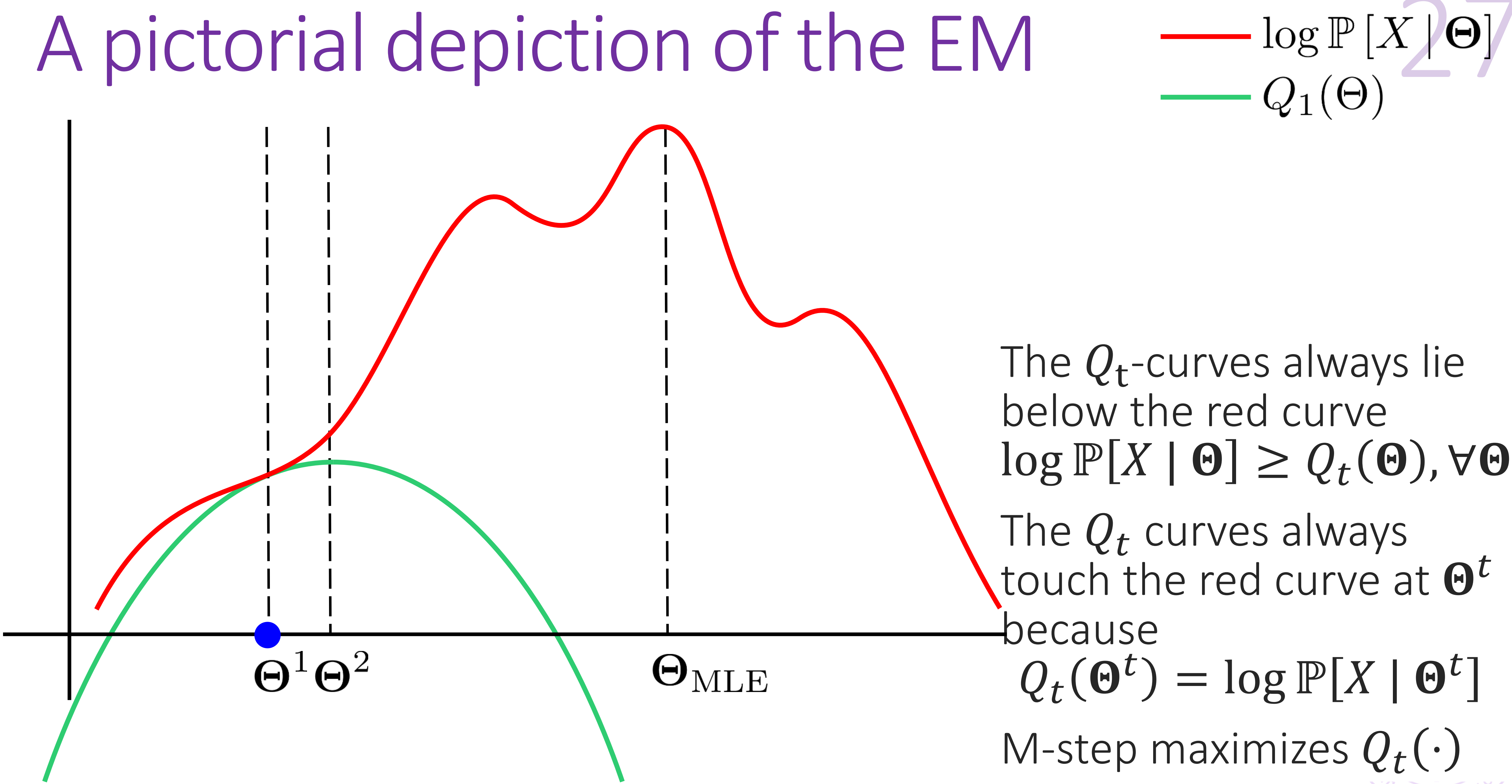
$$\log \mathbb{P}[X \mid \boldsymbol{\Theta}] \geq Q_t(\boldsymbol{\Theta}), \forall \boldsymbol{\Theta}$$

The Q_t curves always touch the red curve at $\boldsymbol{\Theta}^t$ because

$$Q_t(\boldsymbol{\Theta}^t) = \log \mathbb{P}[X \mid \boldsymbol{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

$$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

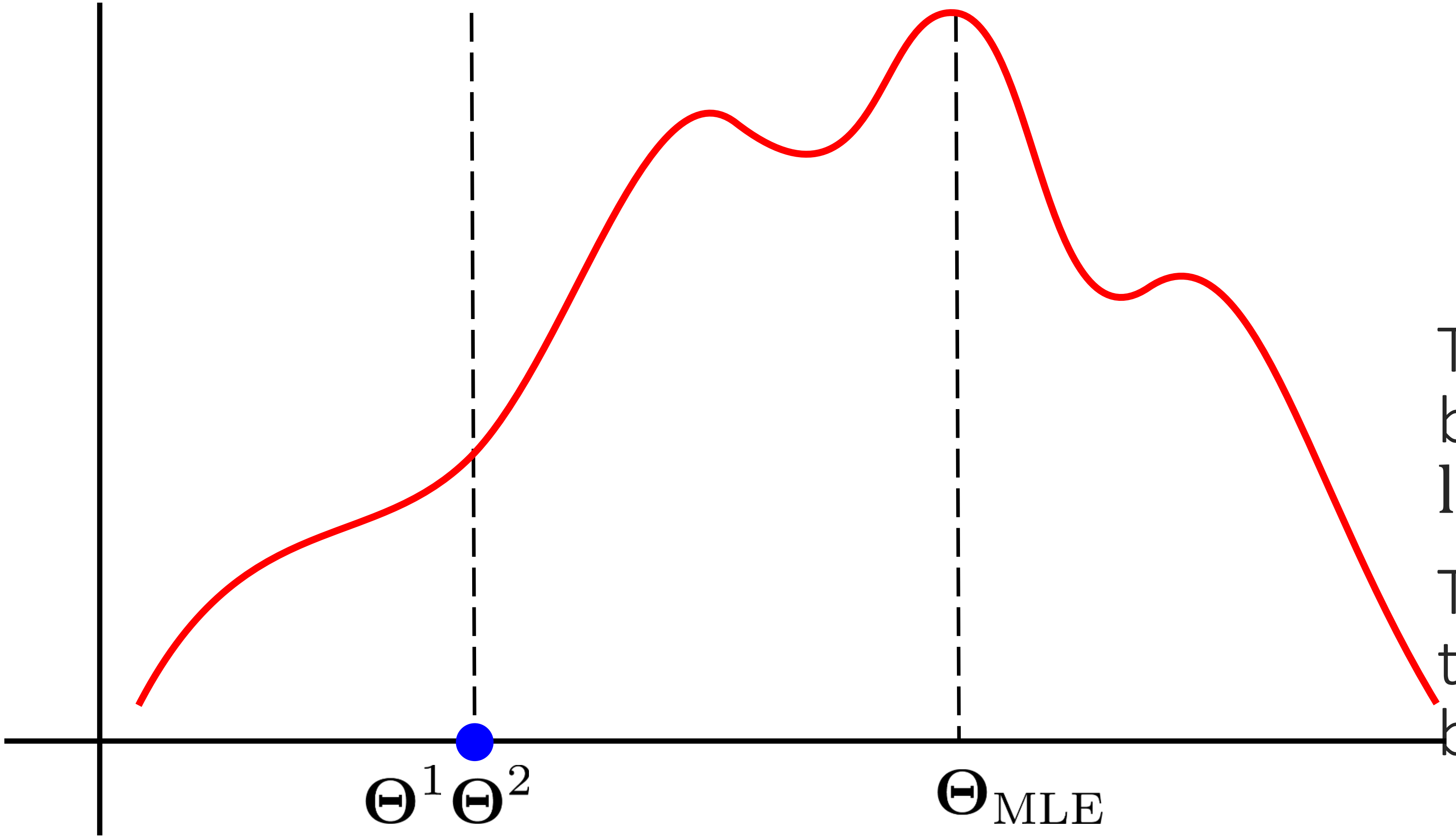
$$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM

🟥 $\log \mathbb{P}\left[X \mid \mathbf{\Theta}\right]$

🟩 $Q_1(\Theta)$



The Q_{t}-curves always lie below the red curve

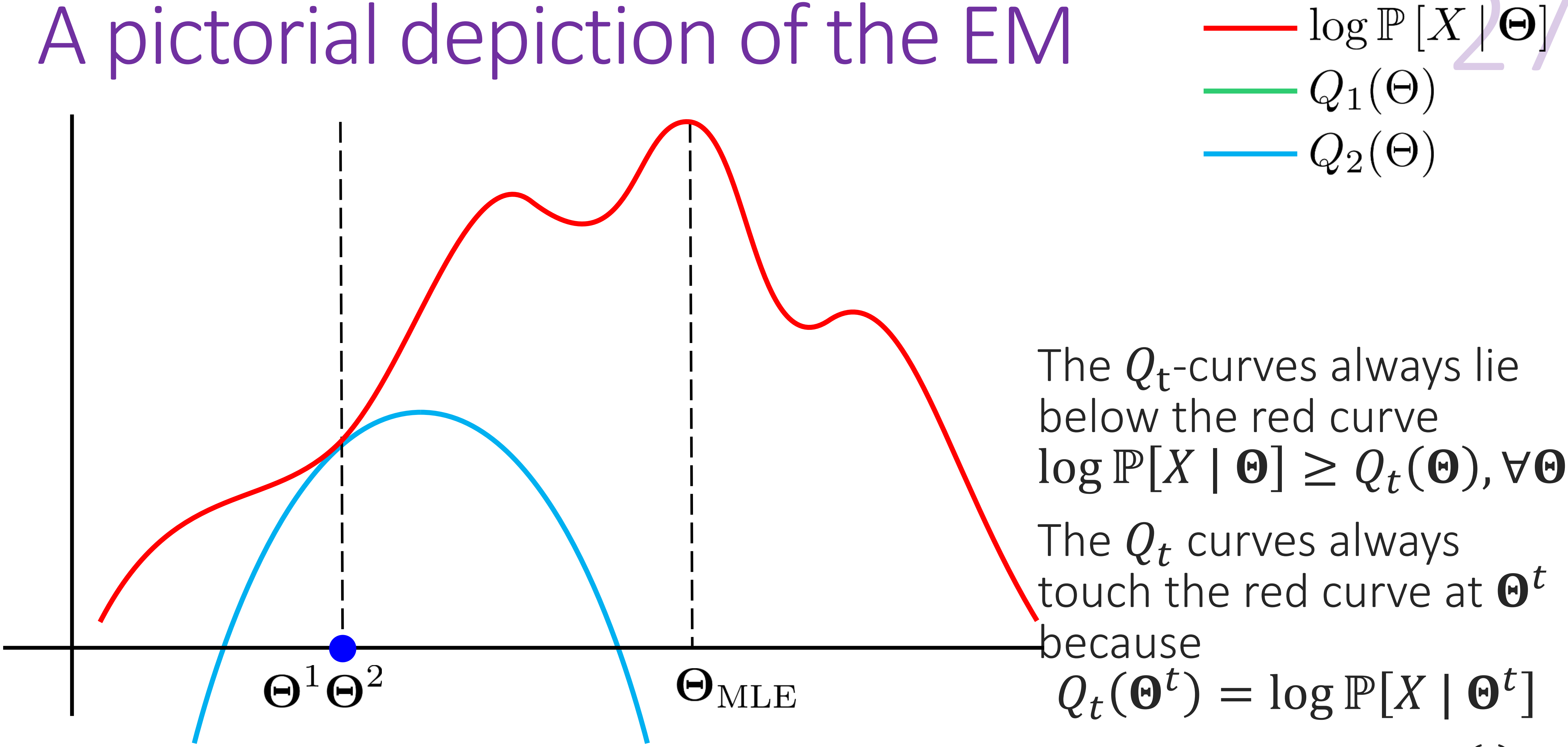
$$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_t-curves always lie below the red curve

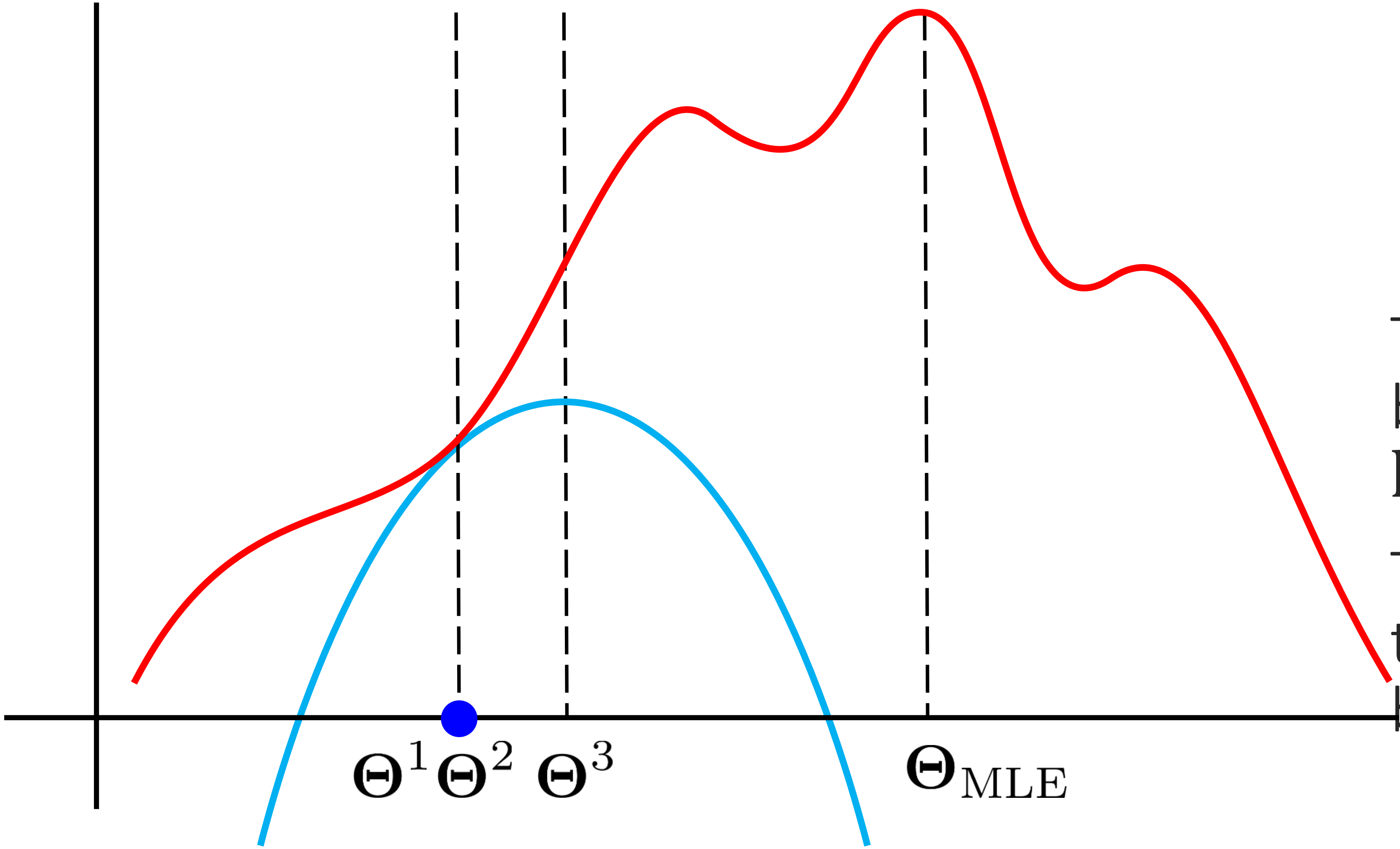
$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

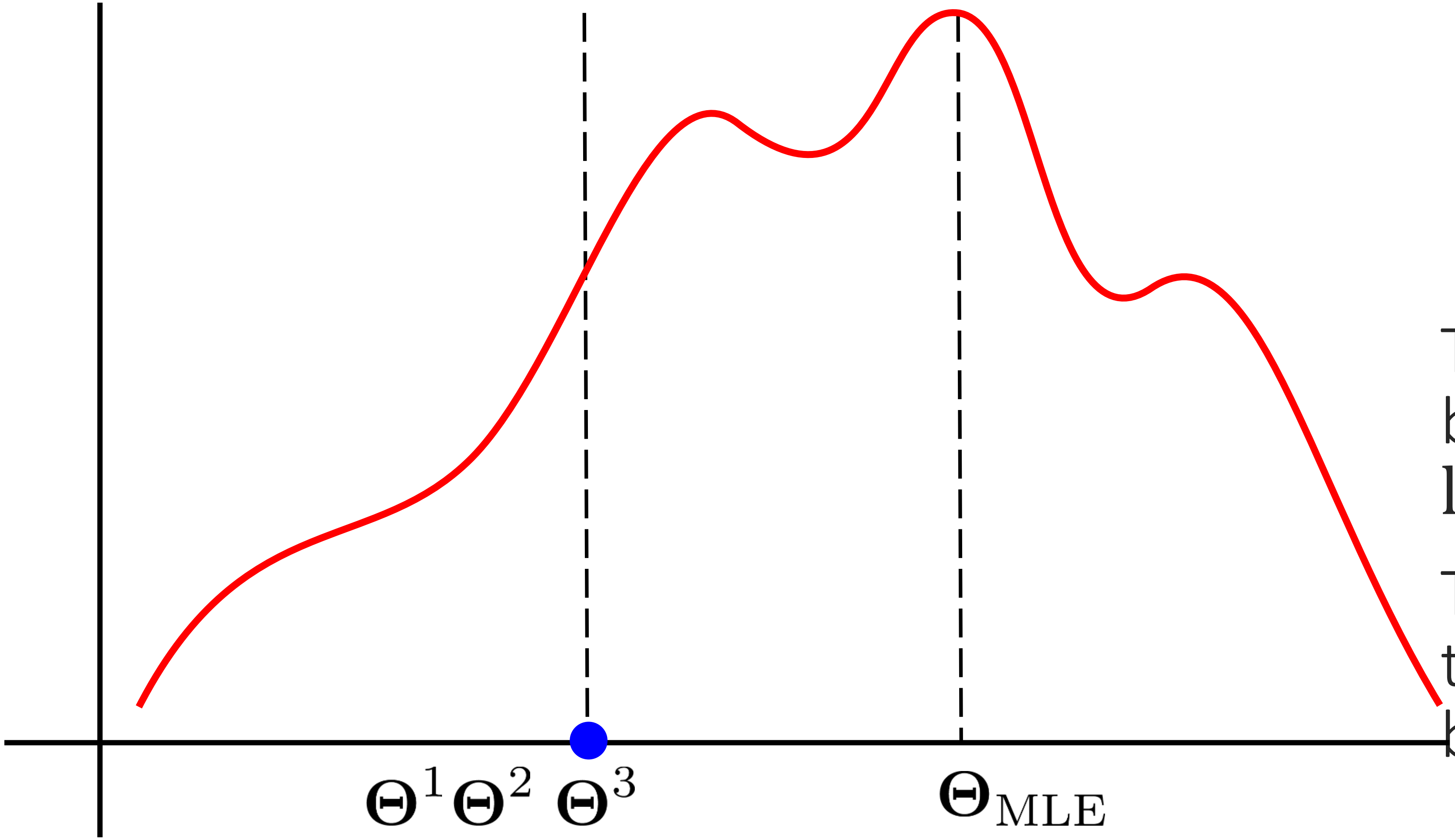
$$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



- $\log \mathbb{P}[X \mid \Theta]$
- $Q_1(\Theta)$
- $Q_2(\Theta)$

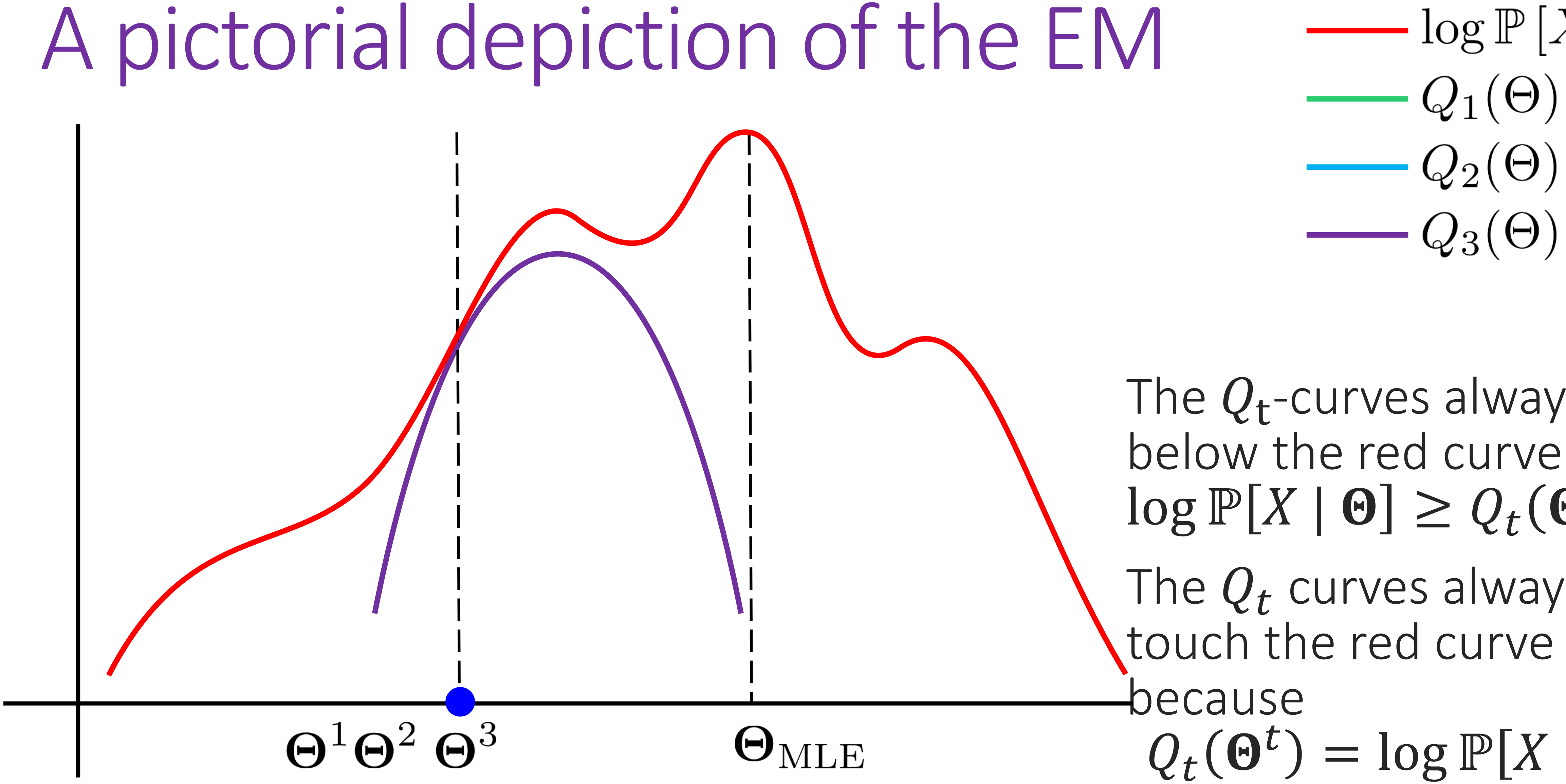
The Q_t-curves always lie below the red curve $\log \mathbb{P}[X \mid \boldsymbol{\Theta}] \geq Q_t(\boldsymbol{\Theta}), \forall \boldsymbol{\Theta}$

The Q_t curves always touch the red curve at $\boldsymbol{\Theta}^t$ because

$$Q_t(\boldsymbol{\Theta}^t) = \log \mathbb{P}[X \mid \boldsymbol{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

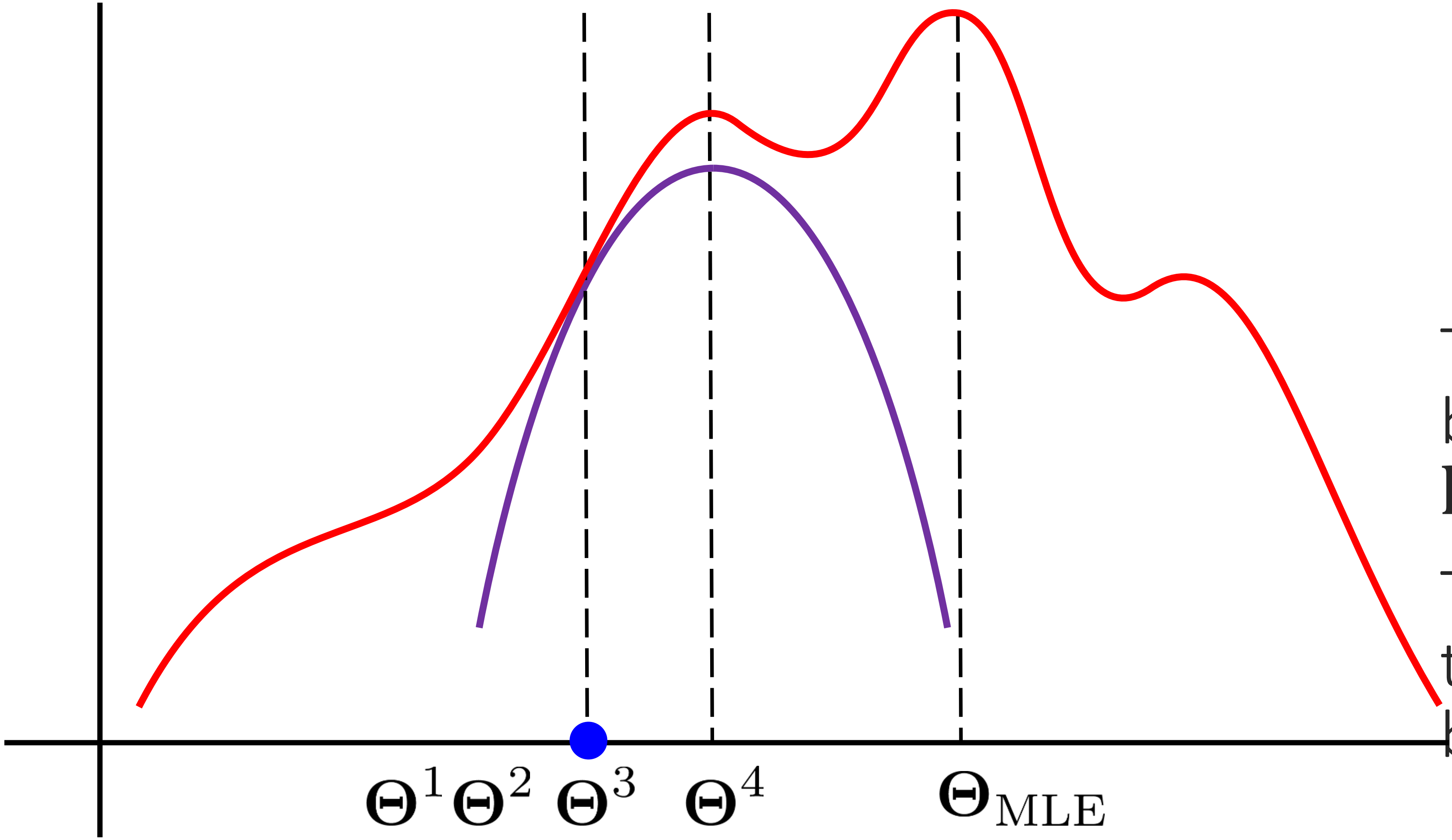
$$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve

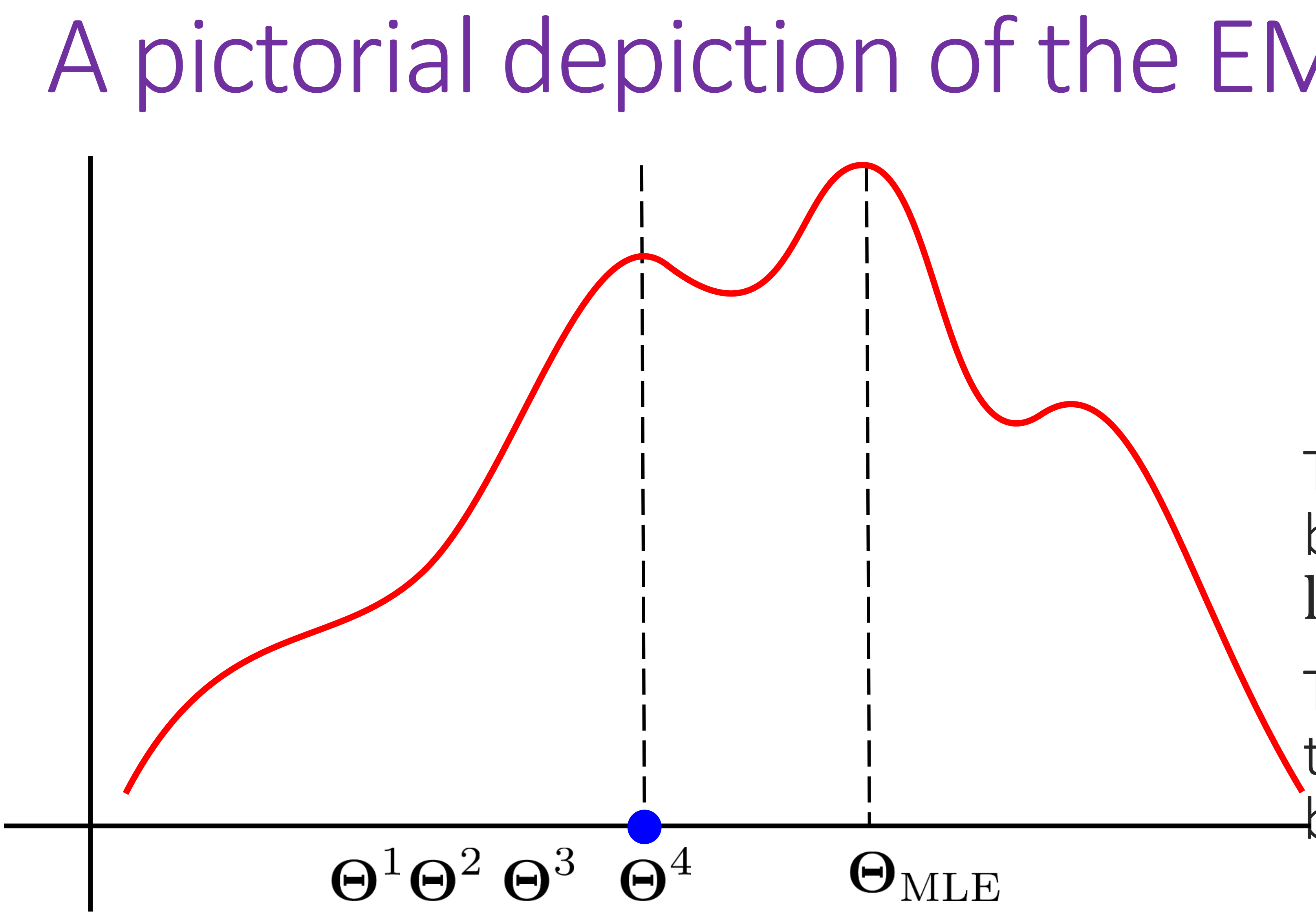
$$\log \mathbb{P}[X \mid \boldsymbol{\Theta}] \geq Q_t(\boldsymbol{\Theta}), \forall \boldsymbol{\Theta}$$

The Q_t curves always touch the red curve at $\boldsymbol{\Theta}^t$ because

$$Q_t(\boldsymbol{\Theta}^t) = \log \mathbb{P}[X \mid \boldsymbol{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_t-curves always lie below the red curve

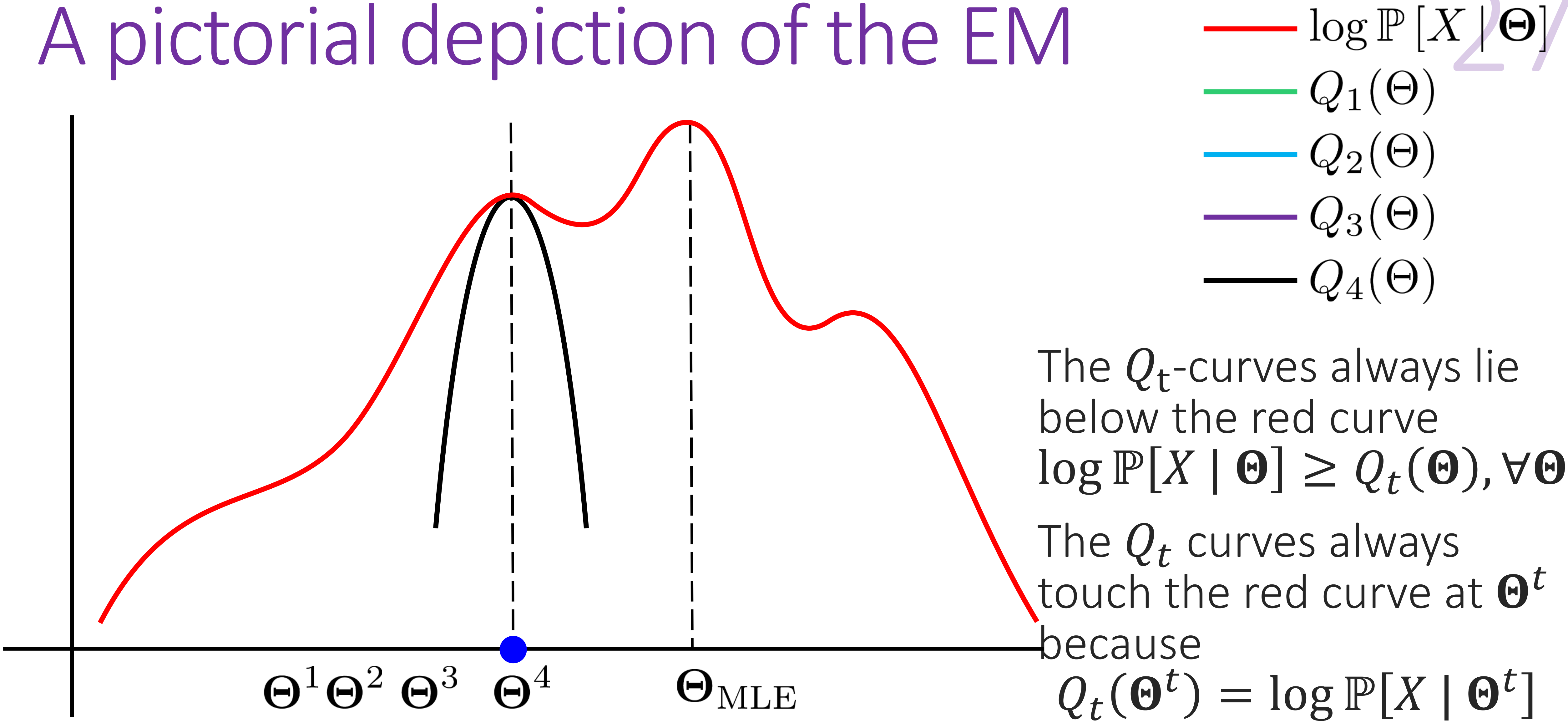
$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_t-curves always lie below the red curve

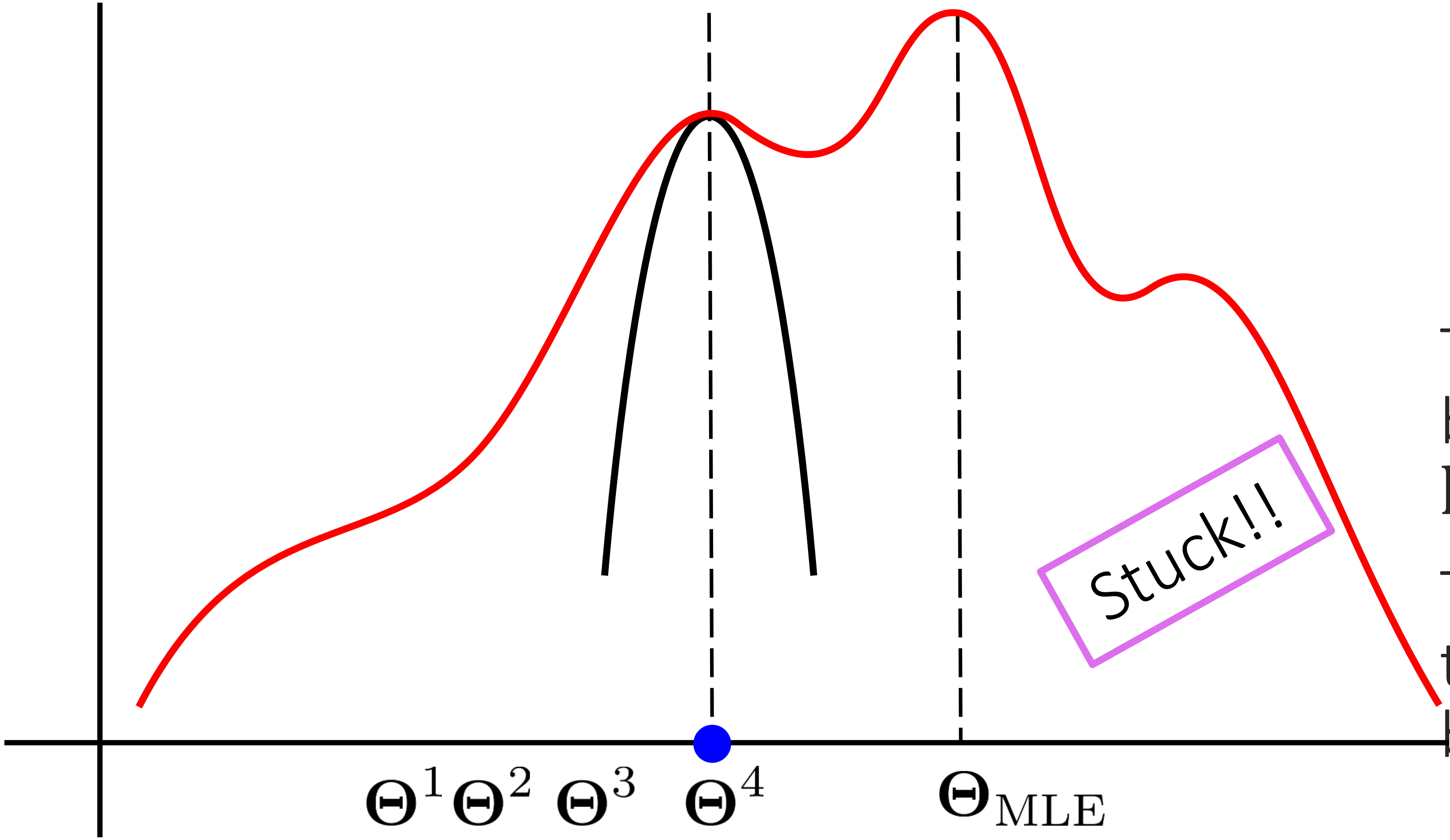
$\log \mathbb{P}[X \mid \boldsymbol{\Theta}] \geq Q_t(\boldsymbol{\Theta}), \forall \boldsymbol{\Theta}$

The Q_t curves always touch the red curve at $\boldsymbol{\Theta}^t$ because

$Q_t(\boldsymbol{\Theta}^t) = \log \mathbb{P}[X \mid \boldsymbol{\Theta}^t]$

M-step maximizes $Q_t(\cdot)$

A pictorial depiction of the EM



The Q_{t}-curves always lie below the red curve $\log \mathbb{P}[X \mid \boldsymbol{\Theta}] \geq Q_t(\boldsymbol{\Theta}), \forall \boldsymbol{\Theta}$

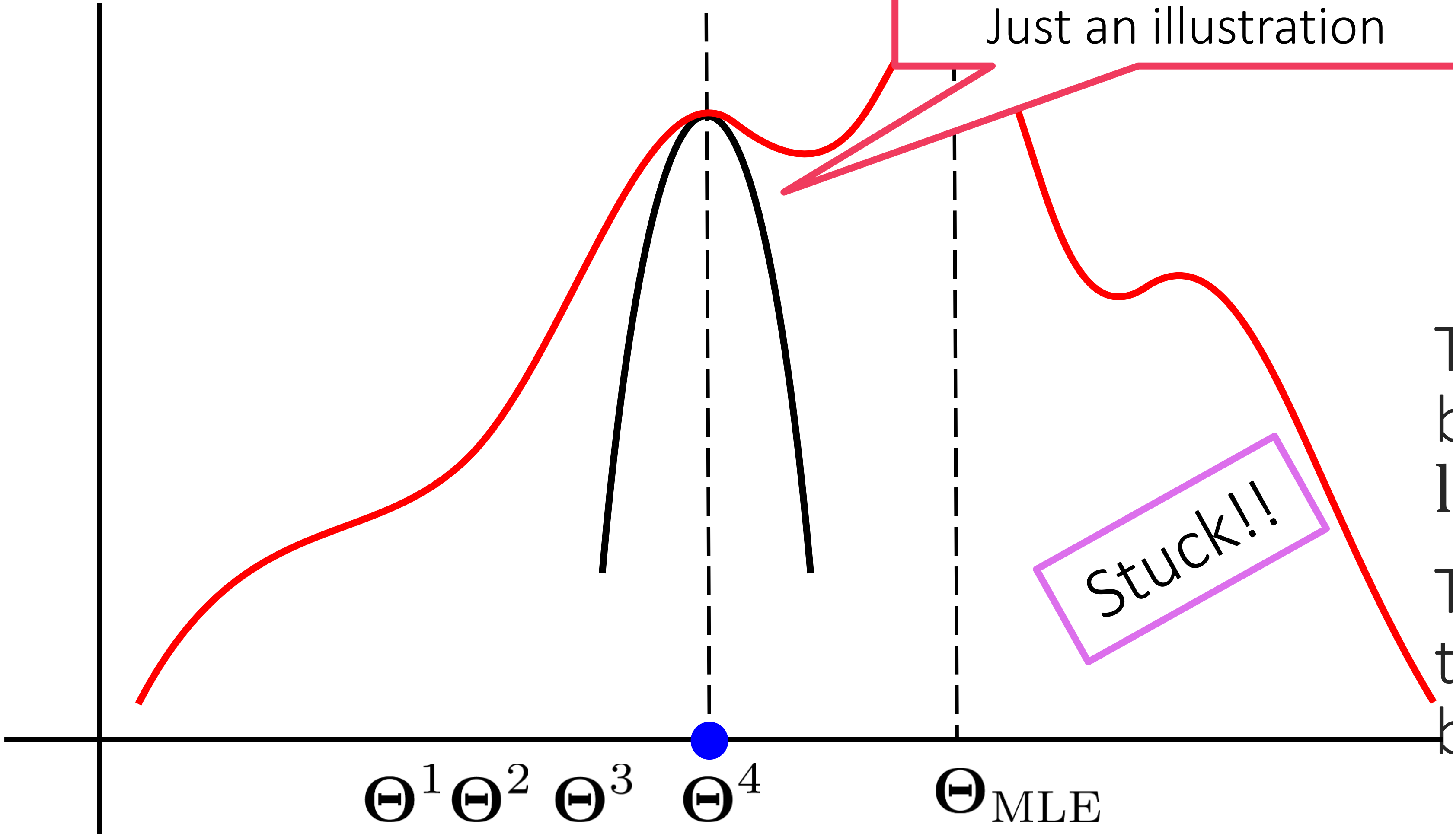
The Q_t curves always touch the red curve at $\boldsymbol{\Theta}^t$ because

$$Q_t(\boldsymbol{\Theta}^t) = \log \mathbb{P}[X \mid \boldsymbol{\Theta}^t]$$

M-step maximizes $Q_t(\cdot)$

A pictorial depicti

$Q_t(\cdot)$ is not necessarily an inverted quadratic fn. Just an illustration



The Q_t-curves always lie below the red curve

$\log \mathbb{P}[X \mid \mathbf{\Theta}] \geq Q_t(\mathbf{\Theta}), \forall \mathbf{\Theta}$

The Q_t curves always touch the red curve at $\mathbf{\Theta}^t$ because

$Q_t(\mathbf{\Theta}^t) = \log \mathbb{P}[X \mid \mathbf{\Theta}^t]$

M-step maximizes $Q_t(\cdot)$