Latent Variable Models and EM Algorithm

CS772A: Probabilistic Machine Learning

Piyush Rai

Hybrid Inference (posterior infer. + point est.)

- In many models, we infer posterior on some unknowns and do point est. for others
- We have already seen MLE-II for lin reg. which alternates between
  - Inferring CP over the main parameter given the point estimates of hyperparams
  - Maximizing the marginal lik. to do point estimation for hyperparams

CP of w: $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \hat{\lambda}, \hat{\beta})$

$\{\hat{\lambda}, \hat{\beta}\} = \text{argmax}_{\lambda,\beta}\, p(\boldsymbol{y}|\boldsymbol{X}, \lambda, \beta)$

- The Expectation-Maximization algorithm (will see today) also does something similar
  - In E step, the CP of latent variables is inferred, given current point-est of params
  - M step maximizes expected complete data log-lik. to get point estimates of params
- If we can't (due to computational or other reasons) infer posterior over all unknowns, how to decide which variables to infer posterior on, and for which to do point-est?
- Usual approach: Infer posterior over local vars and point estimates for global vars
  - Reason: We typically have plenty of data to reliably estimate the global variables so it is okay even if we just do point estimation for those

Nomenclature/Notation Alert

- Why call some unknowns as parameters and others as latent variables?
- Well, no specific reason. Sort of a convention adopted by some algorithms
  - EM: Unknowns estimated in E step referred to as latent vars; those in M step as params
  - Usually: Latent vars – (Conditional) posterior computed; parameters – point estimation
- Some algos won't make such distinction and will infer posterior over all unknowns
- Sometimes the "global" or "local" unknown distinction makes it clear
  - Local variables = latent variables, global variables = parameters
- But remember that this nomenclature isn't really cast in stone, no need to be confused so long as you are clear as to what the role of each unknown is, and how we want to estimate it (posterior or point estimate) and using what type of inference algorithm

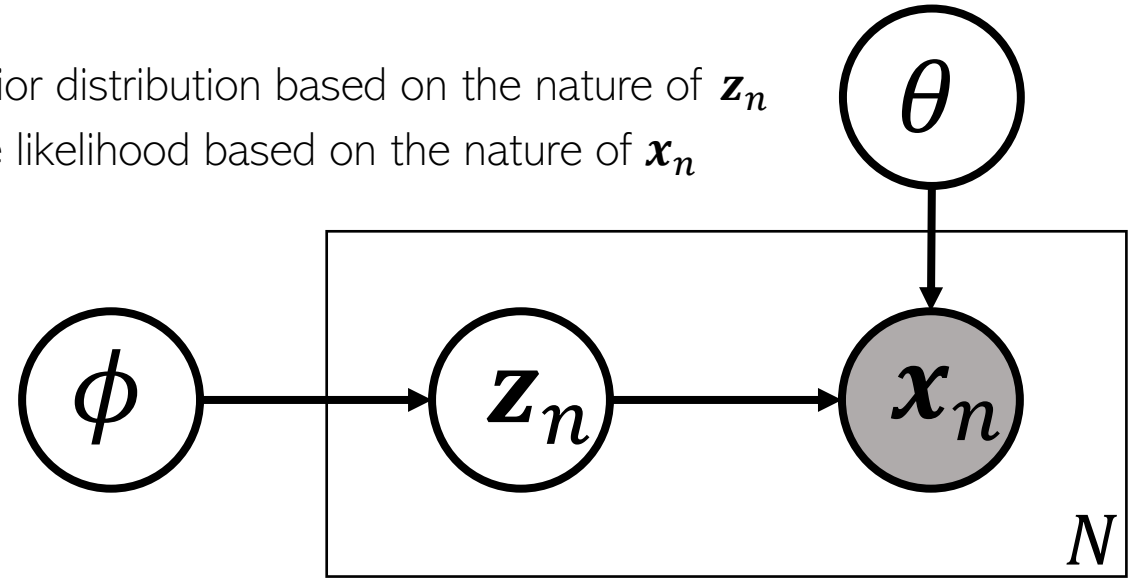
Inference/Parameter Estimation in Latent Variable Models using Expectation-Maximization (EM)

Parameter Estimation in Latent Variable Models

- Assume each observation $\boldsymbol{x}_n$ to be associated with a "local" latent variable $\boldsymbol{z}_n$

$p(\boldsymbol{z}_n|\boldsymbol{\phi})$: A suitable prior distribution based on the nature of $\boldsymbol{z}_n$

$p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\theta})$: A suitable likelihood based on the nature of $\boldsymbol{x}_n$



- Although we can do fully Bayesian inference for all the unknowns, suppose we only want a point estimate of the "global" parameters $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\phi})$ via MLE/MAP
- Such MLE/MAP problems in LVMs are difficult to solve in a "clean" way
  - Would typically require gradient based methods with no closed form updates for $\boldsymbol{\Theta}$
  - However, EM gives a clean way to obtain closed form updates for $\boldsymbol{\Theta}$

Why MLE/MAP of Params is Hard for LVMs?

- Suppose we want to estimate $\Theta = (\theta, \phi)$ via MLE. If we knew $\boldsymbol{z}_n$, we could solve

$$\Theta_{MLE} = \arg\max_{\Theta} \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) = \arg\max_{\Theta} \sum_{n=1}^{N} [\log p(\boldsymbol{z}_n|\phi) + \log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)]$$

Easy to solve

In particular, if they are exp-fam distributions

- Easy. Usually closed form if $p(\boldsymbol{z}_n|\phi)$ and $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)$ have simple forms

- However, since in LVMs, $\boldsymbol{z}_n$ is hidden, the MLE problem for Θ will be the following

Basically, the marginal likelihood after integrating out z_n

$$\Theta_{MLE} = \arg\max_{\Theta} \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \arg\max_{\Theta} \log p(\mathbf{X}|\Theta)$$

- $\log p(\boldsymbol{x}_n|\Theta)$ will not have a simple expression since $p(\boldsymbol{x}_n|\Theta)$ requires sum/integral

$$p(\boldsymbol{x}_n|\Theta) = \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) \quad \text{... or if } \boldsymbol{z}_n \text{ is continuous: } \quad p(\boldsymbol{x}_n|\Theta) = \int p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) d\boldsymbol{z}_n$$

- MLE now becomes difficult (basically MLE-II now), no closed form expression for Θ.

- Can we maximize some other quantity instead of $\log p(\boldsymbol{x}_n|\Theta)$ for this MLE?

An Important Identity

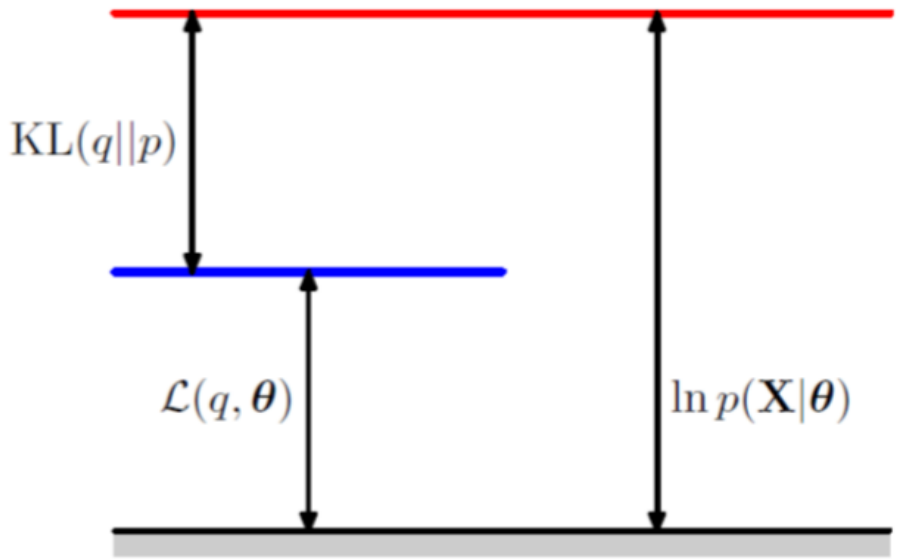
- Assume $p_z = p(\boldsymbol{Z}|\boldsymbol{X}, \Theta)$ and $q(\boldsymbol{Z})$ to be some prob distribution over $\boldsymbol{Z}$, then

$$\log p(\boldsymbol{X}|\Theta) = \mathcal{L}(q, \Theta) + KL(q||p_z)$$

Verify the identity

Assume $\boldsymbol{Z}$ discrete

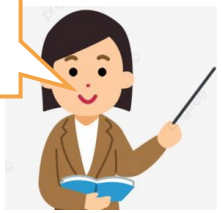
- In the above $\mathcal{L}(q, \Theta) = \sum_Z q(Z) \log\left\{\frac{p(X,Z|\Theta)}{q(Z)}\right\}$
- $KL(q||p_z) = -\sum_Z q(\boldsymbol{Z}) \log\left\{\frac{p(\boldsymbol{Z}|\boldsymbol{X},\Theta)}{q(\boldsymbol{Z})}\right\}$
- KL is always non-negative, so $\log p(\boldsymbol{X}|\Theta) \geq \mathcal{L}(q, \Theta)$
- Thus $\mathcal{L}(q, \Theta)$ is a lower-bound on $\log p(\boldsymbol{X}|\Theta)$
- Thus if we maximize $\mathcal{L}(q, \Theta)$, it will also improve $\log p(\boldsymbol{X}|\Theta)$
- Also, as we'll see, it's easier to maximize $\mathcal{L}(q, \Theta)$



Maximizing $\mathcal{L}(q, \Theta)$

Basically, log of marginal likelihood w.r.t. $\boldsymbol{\Theta}$ with $\boldsymbol{Z}$ integrated out

$\log p(\boldsymbol{X}|\Theta)$ is called Incomplete-Data Log Likelihood (ILL)



- $\mathcal{L}(q, \Theta)$ depends on q and Θ. We'll use ALT-OPT to maximize it
- Let's maximize $\mathcal{L}(q, \Theta)$ w.r.t. q with Θ fixed at some Θ^{old}

Since $\log p(\boldsymbol{X}|\Theta) = \mathcal{L}(q, \Theta) + KL(q||p_z)$ is constant when Θ is held fixed at Θ^{old}

$$\hat{q} = \text{argmax}_q \mathcal{L}(q, \Theta^{\text{old}}) = \text{argmin}_q KL(q||p_z) = p_z = p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}})$$

The posterior distribution of $\boldsymbol{Z}$ given current parameters Θ^{old}

- Now let's maximize $\mathcal{L}(q, \Theta)$ w.r.t. Θ with q fixed at $\hat{q} = p_z = p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}})$

$$\Theta^{\text{new}} = \text{argmax}_\Theta \mathcal{L}(\hat{q}, \Theta) = \text{argmax}_\Theta \sum_Z p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}}) \log \left\{ \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)}{p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}})} \right\}$$

$$= \text{argmax}_\Theta \sum_Z p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}}) \log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)$$

$$= \text{argmax}_\Theta \, \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}})}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)]$$

$$= \text{argmax}_\Theta \, Q(\Theta, \Theta^{\text{old}})$$

Maximization of expected CLL where the expectation is w.r.t. the posterior distribution of $\boldsymbol{Z}$ given current parameters Θ^{old}

Complete-Data Log Likelihood (CLL)

Much easier than maximizing ILL since CLL will have simple expressions (since it is akin to knowing $\boldsymbol{Z}$)

The Expectation-Maximization (EM) Algorithm

- ALT-OPT of $\mathcal{L}(q, \Theta)$ w.r.t. q and Θ gives the EM algorithm (Dempster, Laird, Rubin, 1977)

The EM Algorithm

Primarily designed for doing point estimation of the parameters Θ but also gives (CP of) latent variables z_n

Usually computing CP + expected CLL is referred to as the E step, and max. of exp-CLL w.r.t. Θ as the M step

1. Initialize Θ as $\Theta^{(0)}$, set $t = 1$
2. Step 1: Compute posterior of latent variables given current parameters $\Theta^{(t-1)}$

Conditional posterior of each latent variable z_n

Latent variables also assumed indep. a priori

$$p(\boldsymbol{z}_n^{(t)}|\boldsymbol{x}_n, \Theta^{(t-1)}) = \frac{p(\boldsymbol{z}_n^{(t)}|\Theta^{(t-1)})p(\boldsymbol{x}_n|\boldsymbol{z}_n^{(t)}, \Theta^{(t-1)})}{p(\boldsymbol{x}_n|\Theta^{(t-1)})} \propto \text{prior} \times \text{likelihood}$$

3. Step 2: Now maximize the expected complete data log-likelihood w.r.t. Θ

Assuming the (expected) CLL $\mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}})}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)]$ factorizes over all observations

$$\Theta^{(t)} = \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta^{(t-1)}) = \arg\max_{\Theta} \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n^{(t)}|\boldsymbol{x}_n, \Theta^{(t-1)})}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n^{(t)}|\Theta)]$$

4. If not yet converged, set $t = t + 1$ and go to step 2.

- Note: If we can take the MAP estimate $\hat{\boldsymbol{z}}_n$ of $\boldsymbol{z}_n$ (not full posterior) in Step 1 and maximize the CLL in Step 2 using that, i.e., do $\text{argmax}_{\Theta} \sum_{n=1}^{N} [\log p(\boldsymbol{x}_n, \hat{\boldsymbol{z}}_n^{(t)}|\Theta)]$ this will be ALT-OPT

The Expected CLL

- Expected CLL in EM is given by (assume observations are i.i.d.)

$$\mathcal{Q}(\Theta, \Theta^{old}) = \sum_{n=1}^{N} \mathbb{E}_{p(z_n|x_n,\Theta^{old})}[\log p(x_n, z_n|\Theta)]$$

$$= \sum_{n=1}^{N} \mathbb{E}_{p(z_n|x_n,\Theta^{old})}[\log p(x_n|z_n, \Theta) + \log p(z_n|\Theta)]$$

- If $p(\boldsymbol{z}_n|\Theta)$ and $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \Theta)$ are exp-family distributions, $\mathcal{Q}(\Theta, \Theta^{\mathbf{old}})$ has a very simple form
- In resulting expressions, replace terms containing $\boldsymbol{z}_n$'s by their respective expectations, e.g.,
  - $\boldsymbol{z}_n$ replaced by $\mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \widehat{\Theta})}[\boldsymbol{z}_n]$
  - $\boldsymbol{z}_n\boldsymbol{z}_n^\top$ replaced by $\mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \widehat{\Theta})}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- However, in some LVMs, these expectations are intractable to compute and need to be approximated (will see some examples later)

What's Going On?

Alternating between them until convergence to some local optima

KL becomes zero and $\mathcal{L}(q, \Theta)$ becomes equal to $\log p(\boldsymbol{X}|\Theta)$; thus their curves touch at current Θ

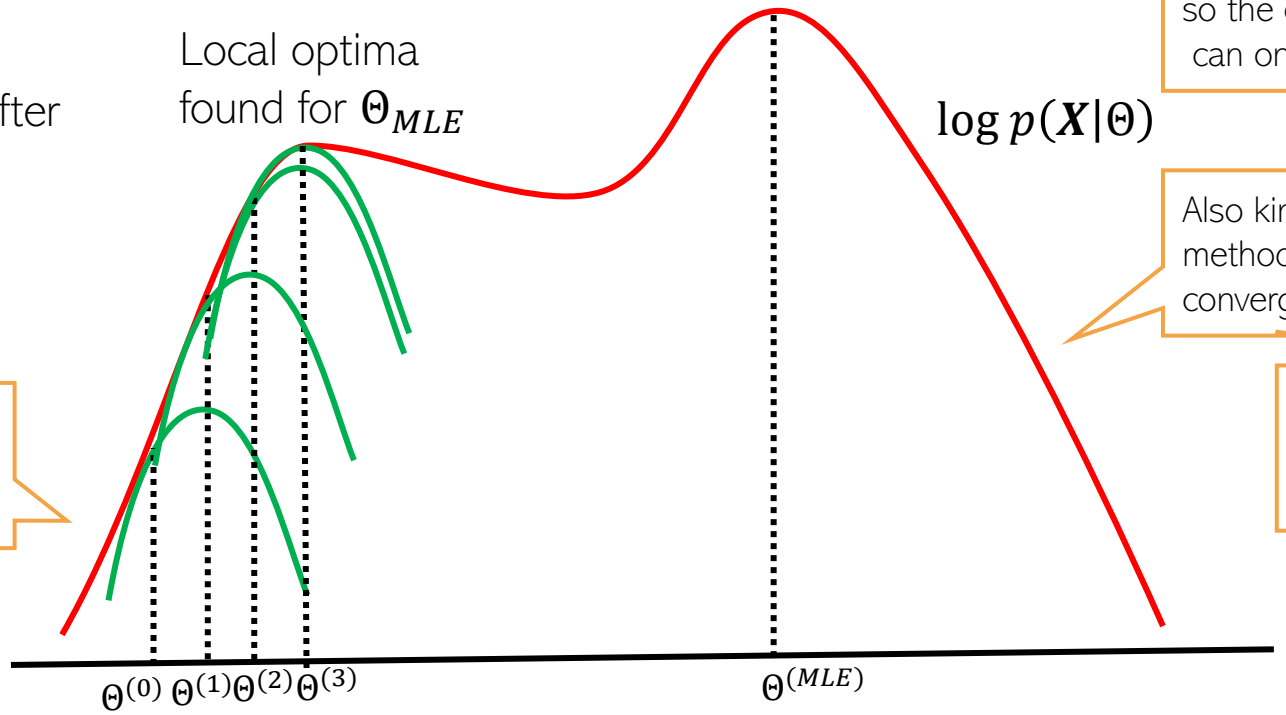
- As we saw, the maximization of lower bound $\mathcal{L}(q, \Theta)$ had two steps
- Step 1 finds the optimal q (call it $\hat{q}$) by setting it as the posterior of $\boldsymbol{Z}$ given current Θ
- Step 2 maximizes $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ which gives a new Θ.

Note that Θ only changes in Step 2 so the objective $\log p(\boldsymbol{X}|\Theta)$ can only change in Step 2

Green curve: $\mathcal{L}(\hat{q}, \Theta)$ after setting q to $\hat{q}$

Local optima found for Θ_{MLE}

$\log p(\boldsymbol{X}|\Theta)$



Also kind of similar to Newton's method (and has second order like convergence behavior in some cases)

Good initialization matters; otherwise would converge to a poor local optima

Unlike Newton's method, we don't construct and optimize a quadratic approximation, but a lower bound

Even though original MLE problem $\text{argmax}_{\Theta} \log p(\boldsymbol{X}|\Theta)$ could be solved using gradient methods, EM often works faster and has cleaner updates

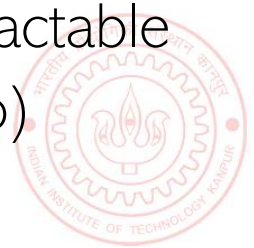
EM vs Gradient-based Methods

- Can also estimate params using gradient-based optimization instead of EM
  - We can usually explicitly sum over or integrate out the latent variables $\boldsymbol{Z}$, e.g.,

$$\mathcal{L}(\Theta) = \log p(\mathbf{X}|\Theta) = \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\Theta)$$

  - Now we can optimize $\mathcal{L}(\Theta)$ using first/second order optimization to find the optimal Θ
- EM is usually preferred over this approach because
  - The M step has often simple closed-form updates for the parameters Θ
  - Often constraints (e.g., PSD matrices) are automatically satisfied due to form of updates
  - In some cases†, EM usually converges faster (and often like second-order methods)
    - E.g., Example: Mixture of Gaussians with when the data is reasonably well-clustered
  - EM applies even when the explicit summing over/integrating out is expensive/intractable
  - EM also provides the conditional posterior over the latent variables Z (from E step)

†Optimization with EM and Expectation-Conjugate-Gradient (Salakhutdinov et al, 2003), On Convergence Properties of the EM Algorithm for Gaussian Mixtures (Xu and Jordan, 1996), Statistical guarantees for the EM algorithm: From population to sample-based analysis (Balakrishnan et al, 2017)



Some Applications of EM

- Mixture Models and Dimensionality Reduction/Representation Learning
  - Mixture Models: Mixture of Gaussians, Mixture of Experts, etc
  - Dim. Reduction/Representation Learning: Probabilistic PCA, Variational Autoencoders
- Problems with missing features or missing labels (which are treated as latent variables)
  - $\widehat{\Theta} = \text{argmax}_{\Theta} \log p(\boldsymbol{x}^{obs}|\Theta) = \text{argmax}_{\Theta} \log \int p([\boldsymbol{x}^{obs}, \boldsymbol{x}^{miss}]|\Theta) d\boldsymbol{x}^{miss}$
  - $\widehat{\Theta} = \text{argmax}_{\Theta} \sum_{n=1}^{N} \log p(x_n, y_n|\Theta) + \sum_{n=N+1}^{N+M} \log \sum_{c=1}^{K} p(x_n, y_n = c|\Theta)$
- Hyperparameter estimation in probabilistic models (an alternative to MLE-II)
  - MLE-II estimates hyperparams by maximizing the marginal likelihood, e.g.,

$$\{\hat{\lambda}, \hat{\beta}\} = \text{argmax}_{\lambda,\beta}\, p(\boldsymbol{y}|\boldsymbol{X}, \lambda, \beta) = \text{argmax}_{\lambda,\beta} \int p(\boldsymbol{y}|\boldsymbol{w}, \boldsymbol{X}, \beta) p(\boldsymbol{w}|\lambda) d\boldsymbol{w}$$

For a Bayesian linear regression model

  - With EM, can treat $\boldsymbol{w}$ as latent var, and $\boldsymbol{\lambda}, \boldsymbol{\beta}$ as "parameters"
    - E step will estimate the CP of $\boldsymbol{w}$ given current estimates of $\boldsymbol{\lambda}, \boldsymbol{\beta}$
    - M step will re-estimate $\boldsymbol{\lambda}, \boldsymbol{\beta}$ by maximizing the expected CLL

$$\mathbb{E}[\log p(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{X}, \beta, \lambda)] = \mathbb{E}[\log p(\boldsymbol{y}|\boldsymbol{w}, \boldsymbol{X}, \beta) + \log p(\boldsymbol{w}|\lambda)]$$

Expectations w.r.t. the CP of $\boldsymbol{w}$