Latent Variable Models (LVMs) and EM for Inference in LVMs

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

Plan

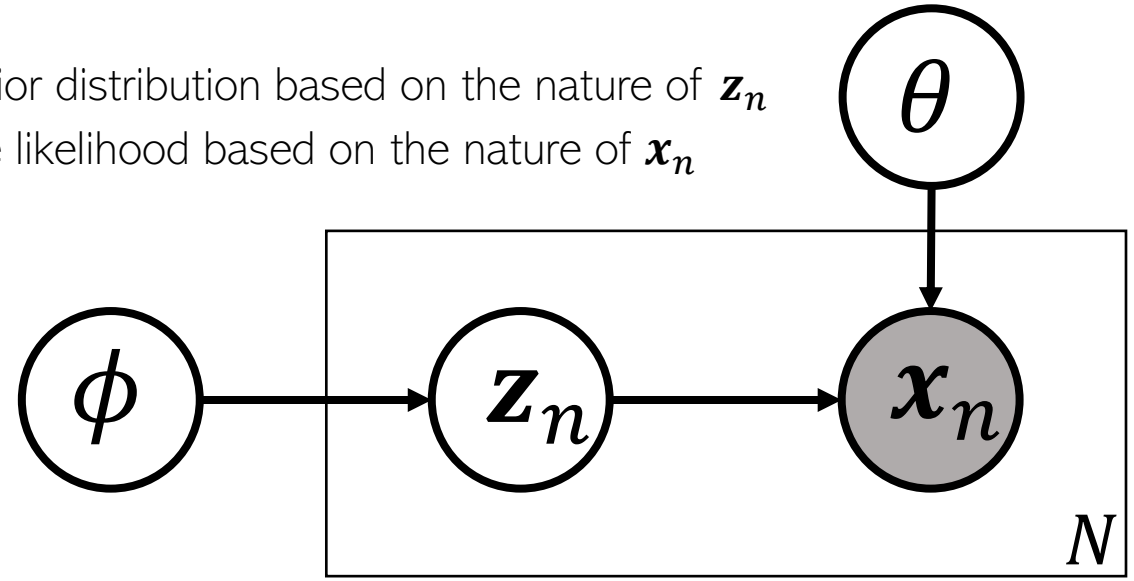
- Latent Variable Models (LVM)
  - The basic formulation of LVMs (specific models later)
  - Parameter Estimation in LVMs
- Expectation Maximization algorithm for param-est/inference LVMs

Latent Variable Models

- Application 1: Can use these to model latent properties/features of data, e.g.,
  - Cluster assignment of each observation (in mixture models)
  - Low-dim rep. or "code" of each observation (e.g., prob. PCA, variational autoencoders, etc)
  - Topic assignment of each word (in topic models such as Latent Dirichlet Allocation)

$p(\boldsymbol{z}_n|\boldsymbol{\phi})$: 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 $\boldsymbol{x}_n$



- In such apps, latent variables ($\boldsymbol{z}_n$'s) are called "local variables" (specific to individual obs.) and other unknown parameters/hyperparams ($\boldsymbol{\theta}, \boldsymbol{\phi}$ above) are called "global var"

Latent Variable Models

- Application 2: Sometimes, augmenting a model by latent variables simplifies inference
  - These latent variables aren't part of the original model definition (hence called latent)
- Some of the popular examples of such augmentation include
  - In Probit regression for binary classification, we can model each label $y_n \in \{0,1\}$ as

$$y_n = \mathbb{I}[z_n > 0] \qquad \text{where} \qquad z_n \sim \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n, 1) \text{ is an auxiliary latent variable}$$

.. and use EM etc, to infer the unknowns $\boldsymbol{w}$ and $\boldsymbol{z_n}$'s (MLAPP 11.4.6, EM for Probit Regression)

  - Many sparse priors on weights can be thought of as Gaussian "scale-mixtures"

$$\text{Laplace}(w_d|0, 1/\gamma) = \frac{\gamma}{2}\exp(-\gamma|w_d|) = \int \mathcal{N}(w_d|0, \tau_d^2)\text{Gamma}(\tau_d^2|1, \gamma^2/2)d\tau_d^2$$

Already talked about this when discussing sparse priors

.. where $\boldsymbol{\tau_d}$'s are latent vars. Can use EM to infer $\boldsymbol{w}$, $\boldsymbol{\tau}$ (MLAPP 13.4.4 - EM for LASSO)

- Such augmentations can often make a non-conjugate model a locally conjugate one
  - Conditional posteriors of the unknowns often have closed form in such cases

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
  - Usual distinction: Latent vars – posterior inferred; parameters – point estimation done
- 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

Hybrid Inference (posterior infer. + point est.)

- In many models, we infer posterior on some unknowns and point estimation for others
- We have already seen that MLE-II based inference does that
  - Maximize the marginal likelihood to do point estimation for hyperparams
  - Infer CP over the main parameter given the point estimates of hyperparams

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

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

- 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

Akin to maximizing marg-lik

- 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 (recall the schools problem in HW1)

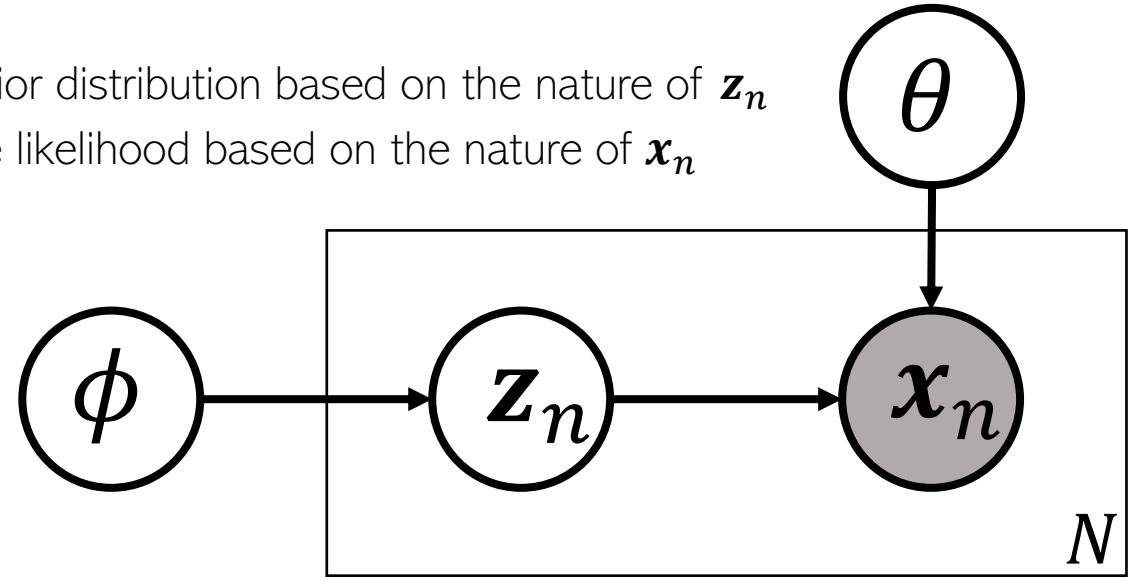
Inference/Parameter Estimation in Latent Variable Models using Expectation-Maximization (EM)

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 $\Theta = (\boldsymbol{\theta}, \boldsymbol{\phi})$ 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 Θ
  - However, EM gives a clean way to obtain closed form updates for Θ

Why MLE/MAP of Params is Hard for LVMs?

- Suppose we want to estimate parameters Θ 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 $p(\boldsymbol{z}_n|\boldsymbol{\phi})$ and $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\theta})$ have simple forms

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

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

- $\log p(\boldsymbol{x}_n|\Theta)$ 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, 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

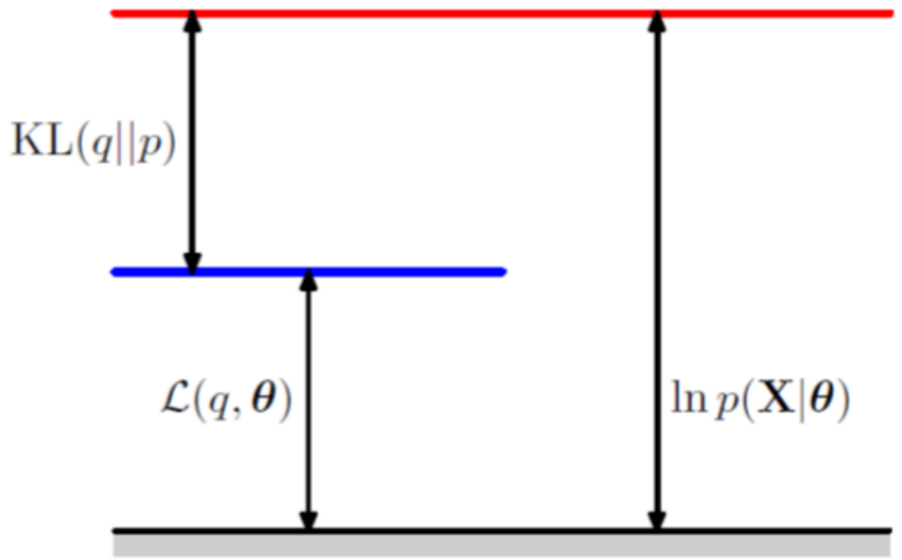
Assume $\boldsymbol{Z}$ discrete

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

$\log p(\boldsymbol{X}|\Theta)$ 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 $\hat{q} = p_z = p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{\text{old}})$

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

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

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

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

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 $\boldsymbol{Z}$)

Coming Up Next

- The EM algorithm and its properties