Latent Variable Models and EM

Slides adapted from CS772A, Piyush Rai, IITK

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 Conditional Posterior (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

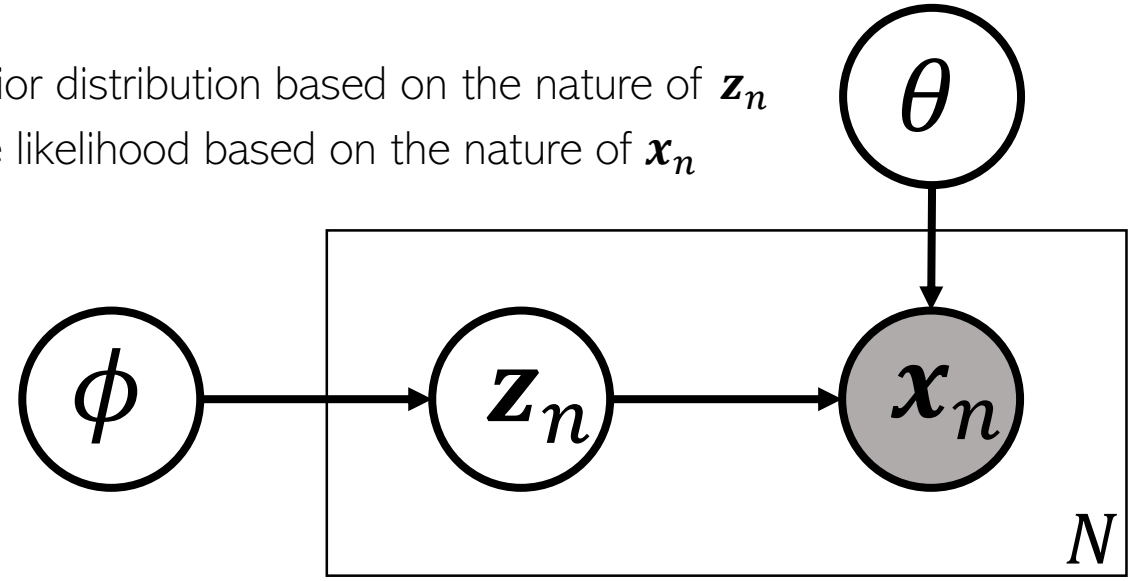
Inference/Parameter Estimation in Latent Variable Models using Expectation-Maximization (EM)

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 $\Theta = (\boldsymbol{\theta}, \boldsymbol{\phi})$ 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 $\Theta = (\theta, \phi)$ 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?

MLE-II

Sometimes our model has two layers of unknowns:

- **Parameters / weights** w (local variables).
- **Hyperparameters** λ, β (global parameters controlling prior distributions, variances, etc.).

- Example: **Bayesian linear regression**
  - Data likelihood: $p(y \mid \boldsymbol{X}, w, \beta)$
  - Prior: $p(w \mid \lambda)$
  - Unknowns: w, β, λ
- If we were fully Bayesian, we'd put priors on everything and integrate

MLE-II

In **MLE-II**

- Integrate out the local parameter w to compute the **marginal likelihood** of hyperparameters:

$$p(y \mid \boldsymbol{X}, \lambda, \beta) = \int p(y \mid \boldsymbol{X}, w, \beta)\, p(w \mid \lambda)\, dw$$

- Then maximize w.r.t. hyperparameters:

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

- This is **MLE-II** (Maximum Likelihood Estimation, stage II).

MLE-I: fit parameters (w) by maximizing likelihood of data.

MLE-II: fit hyperparameters (λ,β) by maximizing the marginal likelihood (a.k.a. “evidence”) after integrating out parameters w.

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

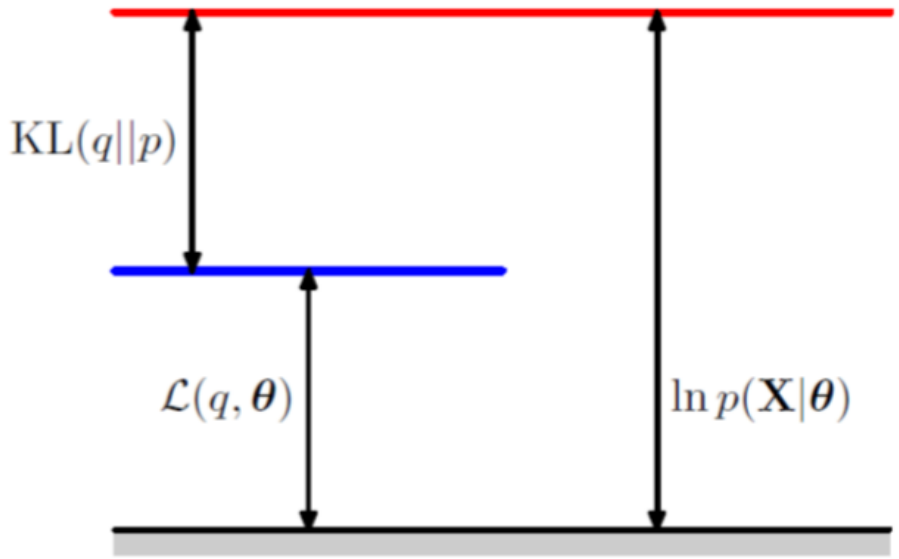
Assume $\boldsymbol{Z}$ discrete

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

Lower bound

KL divergence

Derivation

$$\mathrm{KL}\big(q(Z) \,\|\, p(Z|X,\Theta)\big) = \sum_{Z} q(Z) \log \frac{q(Z)}{p(Z|X,\Theta)}.$$

Intuition

KL is a measure of how far our guess distribution $q(Z)$ is from the true posterior $p(Z|X,\Theta)$. If they're equal, KL = 0.

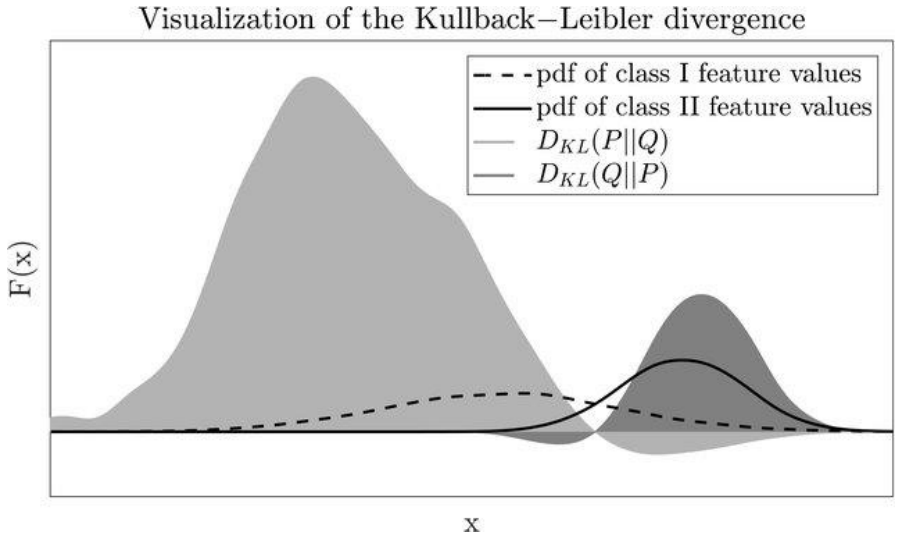
Deriving Lower bound

KL divergence

$$\mathrm{KL}\big(q(Z) \,\|\, p(Z|X,\Theta)\big) = \sum_{Z} q(Z) \log \frac{q(Z)}{p(Z|X,\Theta)}.$$

Intuition

KL is a measure of how far our guess distribution $q(Z)$ is from the true posterior $p(Z|X,\Theta)$. If they're equal, KL = 0.



Deriving Lower bound

Substitute Bayes' rule for the posterior

$$p(Z|X,\Theta) = \frac{p(X,Z|\Theta)}{p(X|\Theta)}.$$

So,

$$\mathrm{KL}(q\|p) = \sum_{Z} q(Z) \log \frac{q(Z)\, p(X|\Theta)}{p(X,Z|\Theta)}$$

We replace the "posterior" with things we know:

- joint $p(X, Z \mid \Theta)$ and
- evidence (marginal likelihood) $p(X \mid \Theta)$

Deriving Lower bound

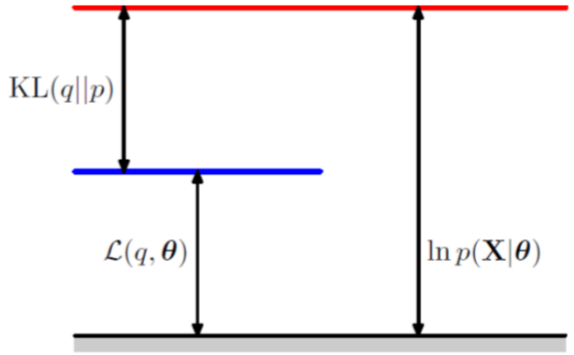
Expand the terms

$$\mathrm{KL}(q\|p) = \sum_Z q(Z)\log q(Z) - \sum_Z q(Z)\log p(X,Z|\Theta) + \log p(X|\Theta)$$

Intuition : We now have three pieces

- an entropy-like term $\sum q \log q$
- a data-fit term $\sum q \log p(X, Z \mid \Theta)$
- Constant log-likelihood of the data $\log p(X \mid \Theta)$

Deriving Lower bound



Rearrange to isolate $\log p(X \mid \Theta)$

$$\log p(X|\Theta) = \underbrace{\sum_Z q(Z)\left[\log p(X, Z|\Theta) - \log q(Z)\right]}_{\mathcal{L}(q,\Theta)} + \mathrm{KL}(q\|p)$$

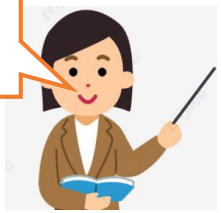
Intuition : We've decomposed the log-likelihood into two parts:

1. **ELBO ($\mathcal{L}(q, \Theta)$)** – how well our chosen $q(Z)$ explains the joint data + latent variables, adjusted by its own complexity.
2. **KL gap** – the "price" we pay because $q(Z)$ isn't the true posterior.

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)

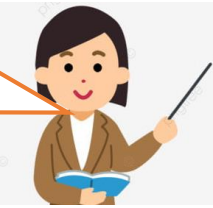
What's Going On?

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.

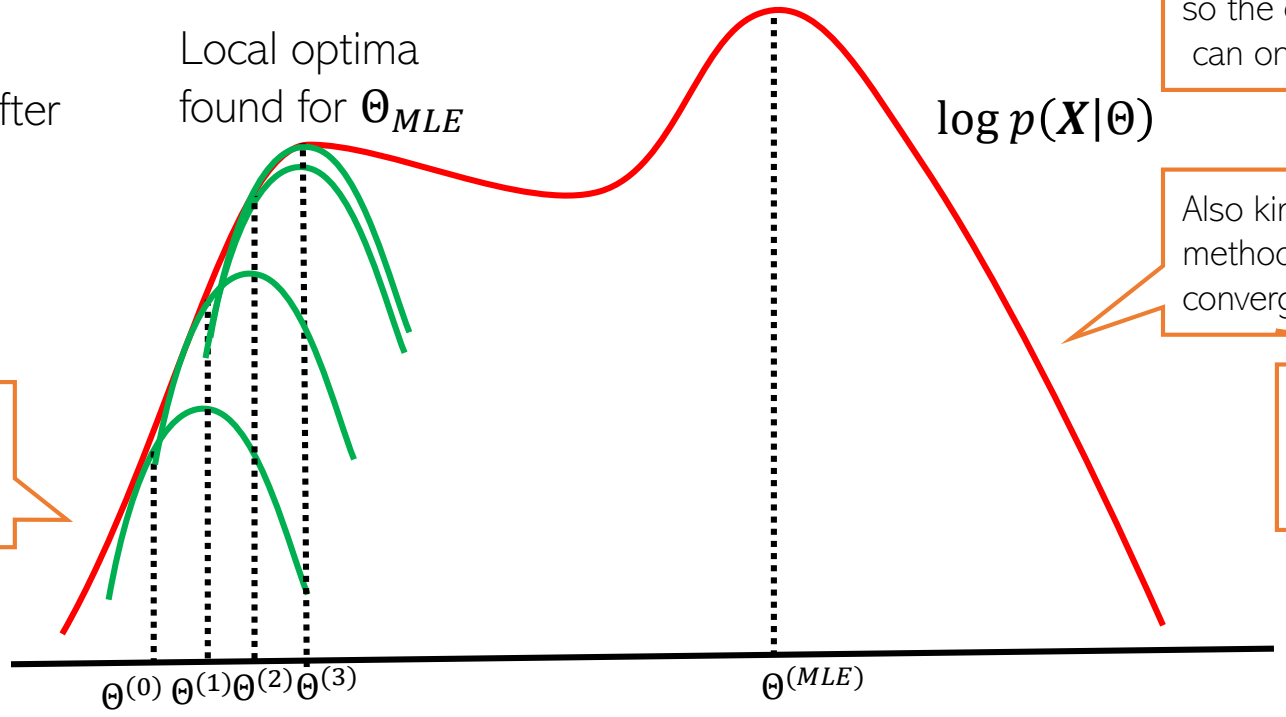
Note that Θ only changes in Step 2 so the objective $\log p(\boldsymbol{X}|\Theta)$ can only change in Step 2



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)
  - $\widehat{\Theta} = \operatorname{argmax}_{\Theta} \log p(\boldsymbol{x}^{obs}|\Theta) = \operatorname{argmax}_{\Theta} \log \int p([\boldsymbol{x}^{obs}, \boldsymbol{x}^{miss}]|\Theta) d\boldsymbol{x}^{miss}$
  - $\widehat{\Theta} = \operatorname{argmax}_{\Theta} \sum_{n=1}^{N} \log p(x_n, y_n|\Theta) + \sum_{n=N+1}^{N+M} \log \sum_{c=1}^{K} p(x_n, y_n = c|\Theta)$
- Hyperparameter estimation in probabilistic models (an alternative to MLE-II)
  - MLE-II estimates hyperparams by maximizing the marginal likelihood, e.g.,

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

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

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

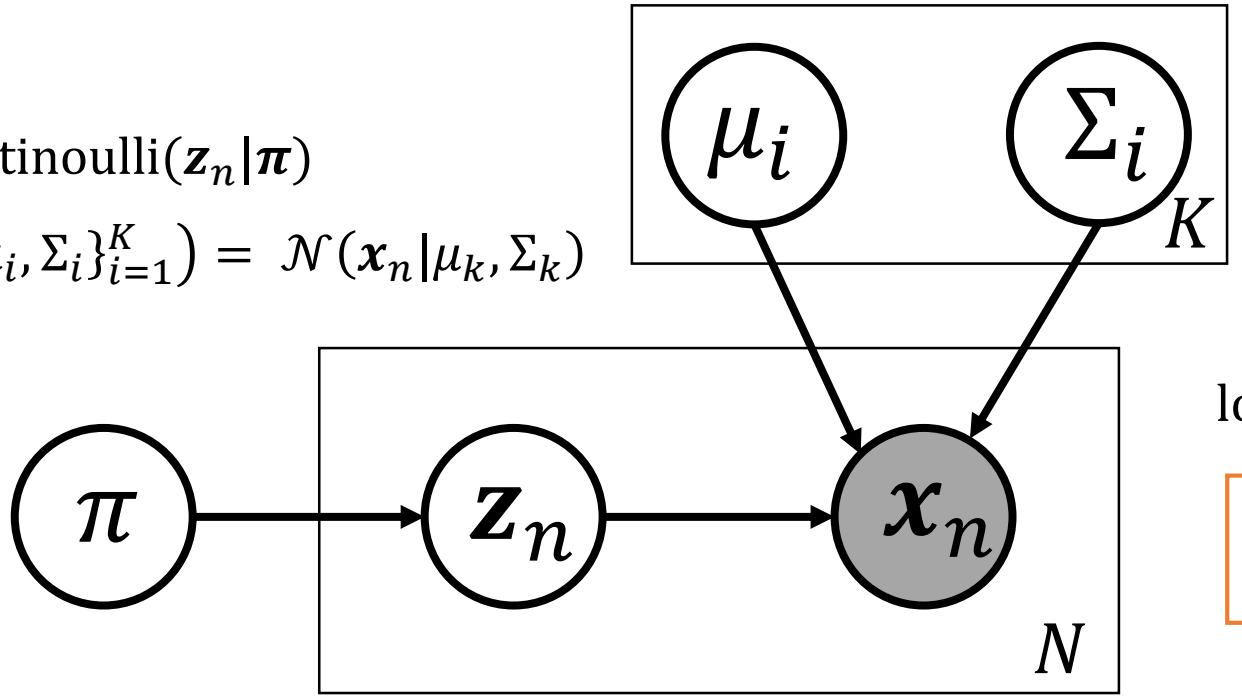
Expectations w.r.t. the CP of $\boldsymbol{w}$

Gaussian Mixture Model (GMM)

- N observations $\{\boldsymbol{x}_n\}_{n=1}^N$ each from one of the K Gaussians $\{\mathcal{N}(\mu_i, \Sigma_i)\}_{i=1}^K$
- We don't know which Gaussian each observation $\boldsymbol{x}_n$ comes from
- Assume $\boldsymbol{z}_n \in \{1,2,\ldots,K\}$ denotes which Gaussian generated $\boldsymbol{x}_n$
- Suppose we want to do point estimation for the parameters $\{\mu_i, \Sigma_i\}_{i=1}^K$

$$p(\boldsymbol{z}_n|\boldsymbol{\pi}) = \text{multinoulli}(\boldsymbol{z}_n|\boldsymbol{\pi})$$

$$p\left(\boldsymbol{x}_n \middle| \boldsymbol{z}_n = k, \{\mu_i, \Sigma_i\}_{i=1}^K\right) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$



$$p\left(\boldsymbol{x}_n \middle| \{\pi_i, \mu_i, \Sigma_i\}_{i=1}^K\right) = \sum_{i=1}^K \pi_i \mathcal{N}(\boldsymbol{x}_n|\mu_i, \Sigma_i)$$

$$\log p(\boldsymbol{x}_n|\Theta) = \log \sum_{i=1}^K \pi_i \mathcal{N}(x_n|\mu_i, \Sigma_i)$$

Can use gradient based optimization for MLE of Θ but the update equations are a bit complicated

EM would give simpler updates

Detour: MLE for GMM when $\boldsymbol{Z}$ is known

GMM then is just like generative classification with Gaussian class conditionals

- Derivation of the MLE solution for $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ when $\boldsymbol{Z}$ is known

$$\hat{\Theta} = \operatorname{argmax}_\Theta p(\boldsymbol{X}, \boldsymbol{Z}|\Theta) = \operatorname{argmax}_\Theta \prod_{n=1}^N p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$$

multinoulli: $p(\boldsymbol{z}_n|\Theta)$; Gaussian: $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \Theta)$

$$= \operatorname{argmax}_\Theta \prod_{n=1}^N p(\boldsymbol{z}_n|\Theta)\, p(\boldsymbol{x}_n|\boldsymbol{z}_n, \Theta)$$

$$= \operatorname{argmax}_\Theta \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \prod_{k=1}^K p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \Theta)^{z_{nk}}$$

$$= \operatorname{argmax}_\Theta \prod_{n=1}^N \prod_{k=1}^K [\pi_k p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \Theta)]^{z_{nk}}$$

$$= \operatorname{argmax}_\Theta \log \prod_{n=1}^N \prod_{k=1}^K [\pi_k p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \Theta)]^{z_{nk}}$$

$$= \operatorname{argmax}_\Theta \sum_{n=1}^N \sum_{k=1}^K z_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

In general, in models with probability distributions from the exponential family, the MLE problem will usually have a simple analytic form

Also, due to the form of the likelihood (Gaussian) and prior (multinoulli), the MLE problem had a nice separable structure after taking the log

Can see that, when estimating the parameters of the k^{th} Gaussian (π_k, μ_k, Σ_k), we only will only need training examples from the k^{th} class, i.e., examples for which $z_{nk} = 1$

EM for Gaussian Mixture Model (GMM)

1. Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ as $\Theta^{(0)}$. Set $t = 1$
2. Set CP $q^{(t)} = p(\boldsymbol{Z}|\boldsymbol{X}, \Theta^{(t-1)})$. Assuming i.i.d. data, this means computing $\forall n, k$

Probability of data point n belonging to the k-th Gaussian

"Soft-clustering"

Same as writing $z_n = k$

$$p(\boldsymbol{z}_{nk} = 1|\boldsymbol{x}_n, \Theta^{(t-1)}) \propto p(\boldsymbol{z}_{nk} = 1|\Theta^{(t-1)})\, p(\boldsymbol{x}_n|\boldsymbol{z}_{nk} = 1, \Theta^{(t-1)})$$
$$= \pi_k^{(t-1)} \mathcal{N}\left(\boldsymbol{x}_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)}\right)$$

EM for GMM does two things: soft-clustering and estimating the density $p(X|\Theta)$

3. Set $\Theta^{(t)} = \text{argmax}_\Theta\, \mathbb{E}_{q^{(t)}}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)] = \text{argmax}_\Theta\, \mathcal{Q}(\Theta, \Theta^{(t-1)})$

This only required expectation for EM for GMM is $\mathbb{E}[z_{nk}]$ which can be computed easily using the CP of z_n

$$\Theta^{(t)} = \text{argmax}_\Theta \sum_{n=1}^N \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \Theta^{(t-1)})}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)]$$
$$= \text{argmax}_\Theta\, \mathbb{E}\left[\sum_{n=1}^N \sum_{k=1}^K z_{nk}\left[\log \pi_k^{(t-1)} + \log \mathcal{N}\left(\boldsymbol{x}_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)}\right)\right]\right]$$
$$= \text{argmax}_\Theta \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}[z_{nk}][\log \pi_k^{(t-1)} + \log \mathcal{N}\left(\boldsymbol{x}_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)}\right)]$$

$$\pi_k^{(t)} = \frac{1}{N}\sum_{n=1}^N \mathbb{E}[z_{nk}]$$
$$\mu_k^{(t)} = \frac{1}{N_k}\sum_{n=1}^N \mathbb{E}[z_{nk}]\boldsymbol{x}_n$$
$$\Sigma_k^{(t)} = \frac{1}{N_k}\sum_{n=1}^N \mathbb{E}[z_{nk}](\boldsymbol{x}_n - \mu_k^{(t)})(\boldsymbol{x}_n - \mu_k^{(t)})^\top$$

$N_k = \sum_{n=1}^N \mathbb{E}[z_{nk}]$ denotes the effective number of points from k-th Gaussian

4. Go to step 2 if not converged

EM for GMM: The Full Algorithm

- The EM algo for GMM required $\mathbb{E}[z_{nk}]$. Note $z_{nk} \in \{0,1\}$

Reason: $\sum_{k=1}^{K} \gamma_{nk} = 1$

Need to normalize: $\mathbb{E}[z_{nk}] = \frac{\hat{\pi}_k \mathcal{N}(x_n|\hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{\ell=1}^{K} \hat{\pi}_\ell \mathcal{N}(x_n|\hat{\mu}_\ell, \hat{\Sigma}_\ell)}$

$$\mathbb{E}[z_{nk}] = \gamma_{nk} = 0 \times p(z_{nk} = 0|x_n, \hat{\Theta}) + 1 \times p(z_{nk} = 1|x_n, \hat{\Theta}) = p(z_{nk} = 1|x_n, \hat{\Theta}) \propto \hat{\pi}_k \mathcal{N}(x_n|\hat{\mu}_k, \hat{\Sigma}_k)$$

EM for Gaussian Mixture Model

1. Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ as $\Theta^{(0)}$, set $t = 1$
2. E step: compute the expectation of each z_n (we need it in M step)

Soft K-means, which is more of a heuristic to get soft-clustering, also gave us probabilities but doesn't account for cluster shapes or fraction of points in each cluster

Accounts for fraction of points in each cluster

Accounts for cluster shapes (since each cluster is a Gaussian

$$\mathbb{E}[z_{nk}^{(t)}] = \gamma_{nk}^{(t)} = \frac{\pi_k^{(t-1)} \mathcal{N}(x_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)})}{\sum_{\ell=1}^{K} \pi_\ell^{(t-1)} \mathcal{N}(x_n|\mu_\ell^{(t-1)}, \Sigma_\ell^{(t-1)})} \quad \forall n, k$$

3. Given "responsibilities" $\gamma_{nk} = \mathbb{E}[z_{nk}]$, and $N_k = \sum_{n=1}^{N} \gamma_{nk}$, re-estimate Θ via MLE

Effective number of points in the k^{th} cluster

M-step:

$$\mu_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}^{(t)} x_n$$

$$\Sigma_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}^{(t)} (x_n - \mu_k^{(t)})(x_n - \mu_k^{(t)})^\top$$

$$\pi_k^{(t)} = \frac{N_k}{N}$$

4. Set $t = t + 1$ and go to step 2 if not yet converged

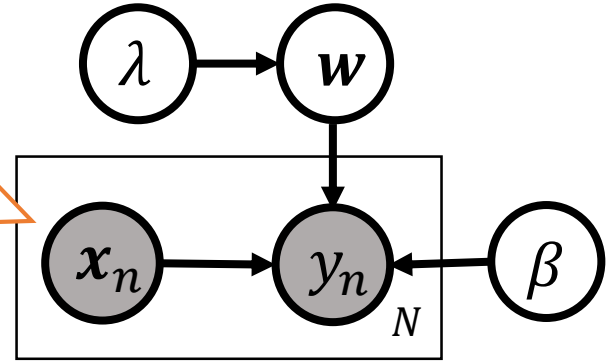
Bayesian Linear Regression (Revisited)

$N \times D$ input matrix

$N \times 1$ responses

- N observations $(\boldsymbol{X}, \boldsymbol{y}) = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$ from a lin-reg model with weights $\boldsymbol{w}$
- Suppose the hyperparameters are also unknown, so need to estimate $\boldsymbol{w}, \beta, \lambda$

$$p(y_n|\boldsymbol{x}_n, \boldsymbol{w}, \beta) = \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1}) \quad p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\mathbf{I})$$



In this latent variable model, there are no local variables. w, β, λ are all "global"

CP of $\boldsymbol{w}$: $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \beta, \lambda) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\boldsymbol{\Sigma} = (\beta \boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \qquad \boldsymbol{\mu} = \beta \boldsymbol{A}^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

Many ways to optimize the marginal likelihood in MLE-II, e.g., gradient descent

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

EM solves the MLE-II problem by optimizing a lower bound on the log marginal likelihood, and gives simple update equations for β, λ

Expected CLL

Data

$\boldsymbol{w}$ treated as latent variable here

EM $(\hat{\beta}, \hat{\lambda}) = \text{argmax}_{\beta,\lambda} \mathbb{E}_{p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y},\beta^{(t-1)},\lambda^{(t-1)})}[\log p(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{X}, \beta, \lambda)]$

$= \text{argmax}_{\beta,\lambda} \mathbb{E}_{p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y},\beta^{(t-1)},\lambda^{(t-1)})}[\log p(\boldsymbol{y}|\boldsymbol{w}, \boldsymbol{X}, \beta) + \log p(\boldsymbol{w}|\lambda)]$

EM for Bayesian Linear Regression

$$(\beta^{(t)}, \lambda^{(t)}) = \text{argmax}_{\beta,\lambda}\, \mathbb{E}\left[\log p\left(\boldsymbol{y}, \boldsymbol{w} | \boldsymbol{X}, \beta^{(t-1)}, \lambda^{(t-1)}\right)\right]$$

1. Initialize β as $\beta^{(0)}$ and λ as $\lambda^{(0)}$. Set $t = 1$
2. Update the CP of $\boldsymbol{w}$ as

$$p\left(\boldsymbol{w} | \boldsymbol{X}, \boldsymbol{y}, \beta^{(t-1)}, \lambda^{(t-1)}\right) = \mathcal{N}(\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})$$

$$\boldsymbol{\Sigma}^{(t)} = \left(\beta^{(t-1)} \boldsymbol{X}^\top \boldsymbol{X} + \lambda^{(t-1)} \boldsymbol{I}\right)^{-1} \qquad \boldsymbol{\mu}^{(t)} = \beta^{(t-1)} \boldsymbol{\Sigma}^{(t)} \boldsymbol{X}^\top \boldsymbol{y}$$

3. Update β and λ as

$$\lambda^{(t)} = \frac{D}{\mathbb{E}[\boldsymbol{w}^\top \boldsymbol{w}]} = \frac{D}{\boldsymbol{\mu}^{(t)^\top} \boldsymbol{\mu}^{(t)} + \text{trace}(\boldsymbol{\Sigma}^{(t)})}$$

Intuition: If weights are large, λ decreases (weaker prior). If weights are small, λ increases (stronger shrinkage).

$$\beta^{(t)} = \frac{N}{\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\mu}^{(t)}\right\|^2 + \text{trace}(\boldsymbol{X}^\top \boldsymbol{\Sigma}^{(t)} \boldsymbol{X})}$$

Intuition: β (noise precision) increases when residuals are small (model fits well), decreases when residuals are large (data is noisy).

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

EM: Some other examples

- Problems with missing features (which are treated as latent variables)
  - Suppose each input $\boldsymbol{x}_n$ has two parts - observed and missing: $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs}, \boldsymbol{x}_n^{miss}]$
  - For such problems, MLE for a model $p(\boldsymbol{X}|\Theta)$, assuming i.i.d. data, would have the form

$$\hat{\Theta} = \text{argmax}_\Theta \sum_{n=1}^{N} \log p(\boldsymbol{x}_n^{obs}|\Theta)$$

Suppose we are estimating the mean/covariance of a multivariate Gaussian given N input, with some inputs observations may have missing features

$$= \text{argmax}_\Theta \sum_{n=1}^{N} \log \int p([\boldsymbol{x}_n^{obs}, \boldsymbol{x}_n^{miss}]|\Theta) d\boldsymbol{x}_n^{miss}$$

  - Here $\boldsymbol{x}_n^{miss}$ can be treated as a latent variable
  - The CP will be $p(\boldsymbol{x}_n^{miss} \mid \boldsymbol{x}_n^{obs}, \Theta)$
  - Using the CP, compute expected CLL and maximize it w.r.t. Θ

An example of semi-supervised learning

- Problems with missing labels (which are treated as latent variables)

This part is like GMM, thus EM can be used

$$\hat{\Theta} = \text{argmax}_\Theta \sum_{n=1}^{N} \log p(x_n, y_n|\Theta) + \sum_{n=N+1}^{N+M} \log \sum_{c=1}^{K} p(x_n, y_n = c|\Theta)$$

EM when CP and/or expectation is intractable

- EM solves the following step for estimating Θ

$$\Theta^{(t)} = \text{argmax}_{\Theta}\, \mathbb{E}_{q^{(t)}}[\log p(\boldsymbol{\mathcal{D}}, \boldsymbol{Z}|\Theta)] = \text{argmax}_{\Theta} \int \log p(\boldsymbol{\mathcal{D}}, \boldsymbol{Z}|\Theta)\, p\left(\boldsymbol{Z}\middle|\Theta^{(t-1)}, \boldsymbol{\mathcal{D}}\right) d\boldsymbol{Z}$$

- The above problem may be difficult to solve if one/both of the following is true
  1. CP $p\left(\boldsymbol{Z}\middle|\Theta^{(t-1)}, \boldsymbol{\mathcal{D}}\right)$ can't be computed exactly (Solution: Need to approximate the CP)
  2. Integral for the expectation is intractable (Solution: Use Monte Carlo approximation)
     - Draw $\boldsymbol{M}$ i.i.d. samples of $\boldsymbol{Z}$ from the current (exact/approximate) CP $p\left(\boldsymbol{Z}\middle|\Theta^{(t-1)}, \boldsymbol{\mathcal{D}}\right)$

$$\left\{\boldsymbol{Z}^{(i)}\right\}_{i=1}^{M} \sim p\left(\boldsymbol{Z}\middle|\Theta^{(t-1)}, \boldsymbol{\mathcal{D}}\right)$$

     - Use these samples to get a Monte-Carlo approximation of expected CLL and maximize

$$\Theta^{(t)} = \text{argmax}_{\Theta}\, \frac{1}{M} \sum_{i=1}^{M} \log p\left(\boldsymbol{\mathcal{D}}, \boldsymbol{Z}^{(i)}\middle|\Theta\right)$$

- Monte-Carlo approximation is commonly used in such problems

EM: Some Final Comments

- The E and M steps may not always be possible to perform exactly. Some reasons
  - The conditional posterior of latent variables $p(Z|X,\Theta)$ may not be easy to compute
    - Will need to approximate $p(Z|X,\Theta)$ using methods such as MCMC or variational inference
  - Even if $p(Z|X,\Theta)$ is easy, the expected CLL may not be easy to compute

$$\mathbb{E}[\log p(\mathbf{X},\mathbf{Z}|\Theta)] = \int \log p(\mathbf{X},\mathbf{Z}|\Theta)p(\mathbf{Z}|\mathbf{X},\Theta)d\mathbf{Z}$$

Results in Monte-Carlo EM

Can often be approximated by Monte-Carlo using sample from the CP of Z

  - Maximization of the expected CLL may not be possible in closed form
- EM works even if the M step is only solved approximately (Generalized EM)
- If M step has multiple parameters whose updates depend on each other, they are updated in an alternating fashion - called Expectation Conditional Maximization (ECM)
- Other advanced probabilistic inference algos are based on ideas similar to EM
  - E.g., Variational EM, Variational Bayes (VB) inference, a.k.a. Variational Inference (VI)