LVMs and EM Algorithm (contd), Variational Inference

CS772A: Probabilistic Machine Learning

Piyush Rai

Expectation Maximization

Data

Latent variables

- EM is a method to optimize $\log p(\mathcal{D}|\Theta) = \log \sum_Z p(\mathcal{D}, Z|\Theta)$ for point estimation of Θ
- EM optimizes $\mathcal{L}(q,\Theta) = \sum_Z q(Z)\log\left\{\frac{p(\mathcal{D},Z|\Theta)}{q(Z)}\right\}$, which is a lower bound on $\log p(X|\Theta)$

Computing the CP of latent variables

Maximizing the expected CLL

1. Initialize Θ as $\Theta^{(0)}$ somehow (e.g., randomly), set $t = 1$
2. Set $q^{(t)} = p\left(Z|\mathcal{D}, \Theta^{(t-1)}\right) \propto p\left(\mathcal{D}|Z, \Theta^{(t-1)}\right)p(Z|\Theta^{(t-1)})$
3. Set $\Theta^{(t)} = \text{argmax}_\Theta\ \mathbb{E}_{q^{(t)}}[\log p(\mathcal{D}, Z|\Theta)] = \text{argmax}_\Theta\ \mathcal{Q}(\Theta, \Theta^{(t-1)})$
4. If not converged, set $t = t + 1$ and go to step 2

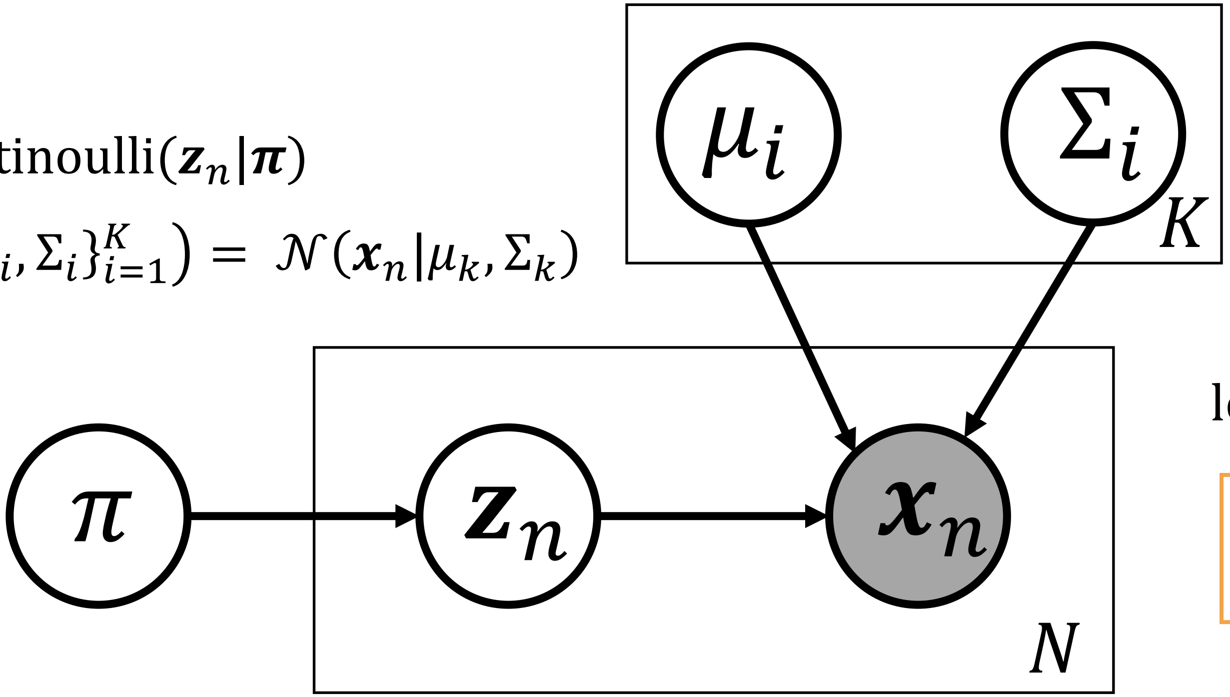
- CP $q^{(t)}$ in step 2 and expectation in step 3 may not be tractable. May need approximations

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 $\{\mu_i, \Sigma_i\}_{i=1}^K$

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

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



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

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

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

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

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

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

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

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

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

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

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 $\forall n, k$

$$p(\boldsymbol{z}_{nk} = 1|\boldsymbol{x}_n, \Theta^{(t-1)}) \propto p(\boldsymbol{z}_{nk} = 1|\Theta^{(t-1)})\, p(\boldsymbol{x}_n|\boldsymbol{z}_{nk} = 1, \Theta^{(t-1)})$$

$$= \pi_k^{(t-1)} \mathcal{N}\left(\boldsymbol{x}_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)}\right)$$

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

"Soft-clustering"

Same as writing $z_n = k$

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

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

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

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

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

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

$$\pi_k^{(t)} = \frac{1}{N}\sum_{n=1}^N \mathbb{E}[z_{nk}]$$

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

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

$N_k = \sum_{n=1}^N \mathbb{E}[z_{nk}]$ 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 $z_{nk} \in \{0,1\}$

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

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

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

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 $\boldsymbol{z}_n$ (we need it in M step)

$$\mathbb{E}[z_{nk}^{(t)}] = \gamma_{nk}^{(t)} = \frac{\pi_k^{(t-1)} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{\ell=1}^{K} \pi_\ell^{(t-1)} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell^{(t-1)}, \boldsymbol{\Sigma}_\ell^{(t-1)})} \quad \forall n, k$$

Accounts for fraction of points in each cluster

Accounts for cluster shapes (since each cluster is a Gaussian

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

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:

$$\boldsymbol{\mu}_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}^{(t)} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}^{(t)} (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})^\top$$

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

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

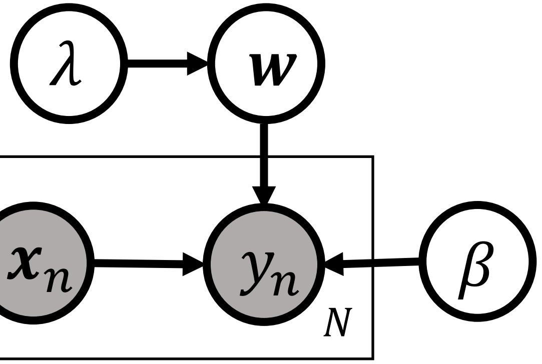
Bayesian Linear Regression (Revisited)

$N \times D$ 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 $\boldsymbol{w}, \beta, \lambda$

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



In this latent variable model, there are no local variables. $\boldsymbol{w}, \beta, \lambda$ 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 $\quad (\hat{\beta}, \hat{\lambda}) = \operatorname{argmax}_{\beta,\lambda} \log p(\boldsymbol{y}|\boldsymbol{X}, \beta, \lambda)$

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

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

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

EM for Bayesian Linear Regression

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

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

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

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

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

Note the dependence: CP of $\boldsymbol{w}$ depends on current values of β, λ and their update depends on the CP on $\boldsymbol{w}$

Less common but another alternative: Compute CP of β and λ in step 2, and compute MLE on $\boldsymbol{w}$ in step 3. That would amount to doing MLE-II for $\boldsymbol{w}$

MLE-II for Bayesian Lin. Reg.

- The MLE-II problem for Bayesian linear regression

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

$$= \text{argmax}_{\beta,\lambda}\, (2\pi)^{-\frac{N}{2}}|\beta^{-1}\mathbf{I}+\lambda^{-1}\boldsymbol{X}^\top\boldsymbol{X}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{y}^\top(\beta^{-1}\mathbf{I}+\lambda^{-1}\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{y}\right)$$

- This objective doesn't have a closed form solution
- Solved using iterative/alternating optimization
  - Gradient descent for $\boldsymbol{\lambda},\boldsymbol{\beta}$
  - Alternating optimization ($\boldsymbol{\lambda},\boldsymbol{\beta}$ and the mean/covariance of the CP depend on each other) - similar to EM but with some differences – next slide
- EM is also a way to do MLE-II but EM doesn't optimize the marginal likelihood but a lower bound on the marginal likelihood

An algorithm for MLE-II for Bayesian Lin. Reg.

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

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

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

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

3. Update β, λ as

RHS depends on β and λ. Thus it is an implicit solution (though still in closed form)

$$\lambda^{(t)} = \frac{\gamma^{(t)}}{\boldsymbol{\mu}^{(t)^\top}\boldsymbol{\mu}^{(t)}}$$

RHS depends on β and λ. Thus it is an implicit solution (though still in closed form)

$$\beta^{(t)} = \frac{N - \gamma^{(t)}}{\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\mu}^{(t)}\right\|^2}$$

In practice, we can compute them in the beginning for $\boldsymbol{X}^\top\boldsymbol{X}$ and multiply by $\beta^{(t-1)}$ in this iteration to get $\left\{\eta_d^{(t)}\right\}_{d=1}^D$

In each iteration, we need to compute the eigenvalues

where

$$\left\{\eta_d^{(t)}\right\}_{d=1}^D = \text{eigvals}(\beta^{(t-1)}\boldsymbol{X}^\top\boldsymbol{X})$$

$$\gamma^{(t)} = \sum_{d=1}^D \frac{\eta_d^{(t)}}{\lambda^{(t-1)} + \eta_d^{(t)}}$$

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

Note that this MLE-II procedure for Bayesian linear regression looks very similar to the EM algo for BLR

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. $\boldsymbol{\Theta}$
- Problems with missing labels (which are treated as latent variables)

An example of semi-supervised learning

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 $\boldsymbol{\Theta}$

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

- 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 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

      Results in Monte-Carlo EM
  - Even if $p(Z|X,\Theta)$ is easy, the expected CLL may not be easy to compute

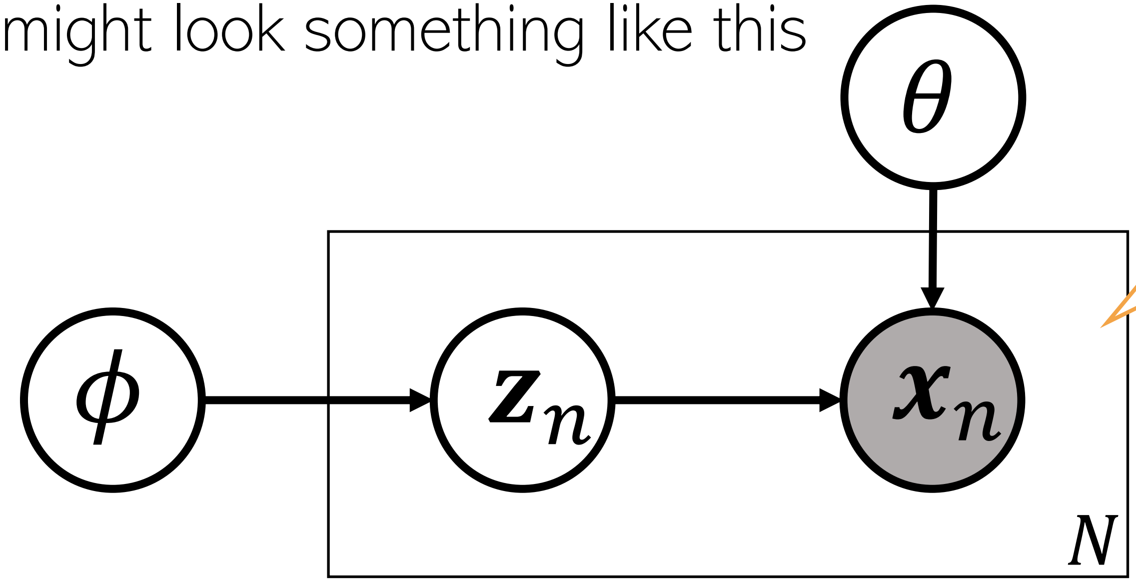
$$\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)] = \int \log p(\mathbf{X}, \mathbf{Z}|\Theta) p(\mathbf{Z}|\mathbf{X}, \Theta) d\mathbf{Z}$$

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)

Variational Inference (VI)

- Assume a latent variable model with data $\boldsymbol{\mathcal{D}}$ and latent variables $\boldsymbol{Z}$
- A simple setting might look something like this



This setting is just one example. VI is applicable in more general and more complex probabilistic models with and without latent variables

- Assume the likelihood is $p(\boldsymbol{\mathcal{D}}|\boldsymbol{Z},\Theta)$ and prior is $p(\boldsymbol{Z}|\Theta)$. Want posterior over $\boldsymbol{Z}$
- $\Theta = (\theta, \phi)$ denotes the other parameters that define the likelihood and the prior
- For now, assume Θ is known and only $\boldsymbol{Z}$ is unknown (the Θ unknown case later)
- Assume CP $p(\boldsymbol{Z}|\boldsymbol{\mathcal{D}},\Theta)$ is intractable

Variational Inference (VI)

- Assuming $p(\boldsymbol{Z}|\boldsymbol{\mathcal{D}},\Theta)$ is intractable, VI approximates it by a distr $q(\boldsymbol{Z}|\phi)$ or $q_\phi(\boldsymbol{Z})$

Find the optimal ϕ which makes our approximation $q(\boldsymbol{Z}|\phi)$ as closed as possible to the true posterior $p(\boldsymbol{Z}|\boldsymbol{\mathcal{D}})$

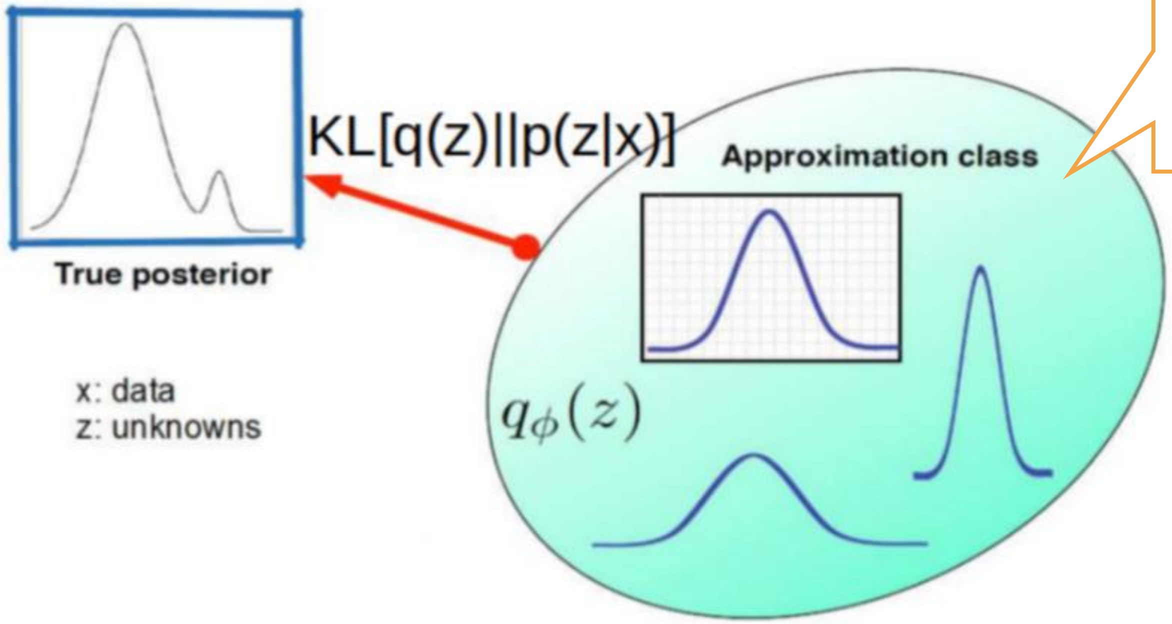
Kullback Leibler divergence $\mathrm{KL}[q||p]$ between q and p

Also possible to use $\mathrm{KL}[p||q]$ or divergences other than KL

$$\phi^* = \operatorname{argmin}_\phi \mathrm{KL}[q_\phi(\boldsymbol{Z})||p(\boldsymbol{Z}|\boldsymbol{\mathcal{D}},\Theta)]$$

q_ϕ defines a class of distributions parametrized by ϕ sometimes called "variational parameters"

Name "variational" comes from Physics and refers to problems where we are optimizing functions of distributions (here the function is the KL divergence)



Variational Inference (VI)

- The optimization problem

$$\phi^* = \text{argmin}_\phi \, \text{KL}[q_\phi(\boldsymbol{Z}) || p(\boldsymbol{Z}|\boldsymbol{\mathcal{D}}, \Theta)]$$

$$= \text{argmin}_\phi \, \mathbb{E}_{q_\phi(\boldsymbol{Z})}\left[\log q_\phi(\boldsymbol{Z}) - \log \frac{p(\boldsymbol{\mathcal{D}}|\boldsymbol{Z}, \Theta)p(\boldsymbol{Z}|\Theta)}{p(\boldsymbol{\mathcal{D}}|\Theta)}\right]$$

$$= \text{argmin}_\phi \, \mathbb{E}_{q_\phi(\boldsymbol{Z})}\left[\log q_\phi(\boldsymbol{Z}) - \log p(\boldsymbol{\mathcal{D}}|\boldsymbol{Z}, \Theta) - \log p(\boldsymbol{Z}|\Theta)\right] + \log p(\boldsymbol{\mathcal{D}}|\Theta)$$

- Since $\log p(\boldsymbol{\mathcal{D}}|\Theta)$ is independent of ϕ, the optimization problem becomes

$$\phi^* = \text{argmin}_\phi \, \mathbb{E}_{q_\phi(\boldsymbol{Z})}\left[\log q_\phi(\boldsymbol{Z}) - \log p(\boldsymbol{\mathcal{D}}|\boldsymbol{Z}, \Theta) - \log p(\boldsymbol{Z}|\Theta)\right]$$

$$\phi^* = \text{argmin}_\phi \, \mathbb{E}_{q_\phi(\boldsymbol{Z})}\left[\log q_\phi(\boldsymbol{Z}) - \log p(\boldsymbol{\mathcal{D}}, \boldsymbol{Z}|\Theta)\right]$$

$$\phi^* = \text{argmax}_\phi \, \mathbb{E}_{q_\phi(\boldsymbol{Z})}\left[\log p(\boldsymbol{\mathcal{D}}, \boldsymbol{Z}|\Theta) - \log q_\phi(\boldsymbol{Z})\right] = \text{argmax} \, \mathcal{L}(\phi, \Theta)$$

- Note that $\mathcal{L}(\phi, \Theta) \leq \log p(\boldsymbol{\mathcal{D}}|\Theta)$ and is called "Evidence Lower Bound" (ELBO)

The ELBO

- The ELBO is defined as

$$\mathcal{L}(\phi, \Theta) = \mathbb{E}_{q_\phi(\boldsymbol{Z})}[\log p(\boldsymbol{\mathcal{D}}, \boldsymbol{Z}|\Theta) - \log q_\phi(\boldsymbol{Z})]$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{Z})}[\log p(\boldsymbol{\mathcal{D}}, \boldsymbol{Z}|\Theta)] + \text{H}[q_\phi(\boldsymbol{Z})]$$

- Thus maximizing the ELBO w.r.t. ϕ gives us a $q_\phi(\boldsymbol{Z})$ which
  - Maximizes the expected joint probability of data and latent variables
  - Has a high entropy
- We can also write the ELBO as follows

$$\mathcal{L}(\phi, \Theta) = \mathbb{E}_{q_\phi(\boldsymbol{Z})}[\log p(\boldsymbol{\mathcal{D}}|\boldsymbol{Z}, \Theta)] - \text{KL}[q_\phi(\boldsymbol{Z})||p(\boldsymbol{Z}|\Theta)]$$

- Thus maximizing the ELBO w.r.t. ϕ will give us a $q_\phi(\boldsymbol{Z})$ which
  - Explains the data $\boldsymbol{\mathcal{D}}$ well, i.e., gives it large expected probability $\mathbb{E}_q[\log p(\boldsymbol{\mathcal{D}}|\boldsymbol{Z}, \Theta)]$
  - Is close to the prior $p(\boldsymbol{Z})$, i.e. is simple/regularized (small $\text{KL}[q_\phi(\boldsymbol{Z})||p(\boldsymbol{Z}|\Theta)]$)

Maximizing the ELBO

Unknown Θ case later

- We need to maximize the ELBO w.r.t. ϕ (for now, assuming Θ is known)

$$\mathcal{L}(\phi, \Theta) = \mathbb{E}_{q_\phi(\boldsymbol{Z})}[\log p(\mathcal{D}|\boldsymbol{Z}, \Theta)] - \mathrm{KL}[q_\phi(\boldsymbol{Z})||p(\boldsymbol{Z}|\Theta)]$$

- The general approach to maximize ELBO is based on gradient-based methods
  - Assume some suitable/convenient form for $q_\phi(\boldsymbol{Z})$, e.g., $\mathcal{N}(\boldsymbol{Z}|\mu, \Sigma)$ so $\phi = (\mu, \Sigma)$
  - Maximize the ELBO w.r.t. ϕ using gradient ascent

$$\phi_{t+1} = \phi_t + \eta_t \nabla_{\phi_t} \mathcal{L}(\phi, \Theta)$$

- Note: Expectations in ELBO and ELBO's gradients w.r.t. ϕ may not be easy
  - Will see methods to handle such issues later
  - Assuming simple forms for $q_\phi(\boldsymbol{Z})$ also helps (we can use random variable transformation methods to transform the simple form to more expressive ones – will see later)