Expectation Maximization (contd)

CS772A: Probabilistic Machine Learning

Piyush Rai

Expectation Maximization

Data

Latent variables

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

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

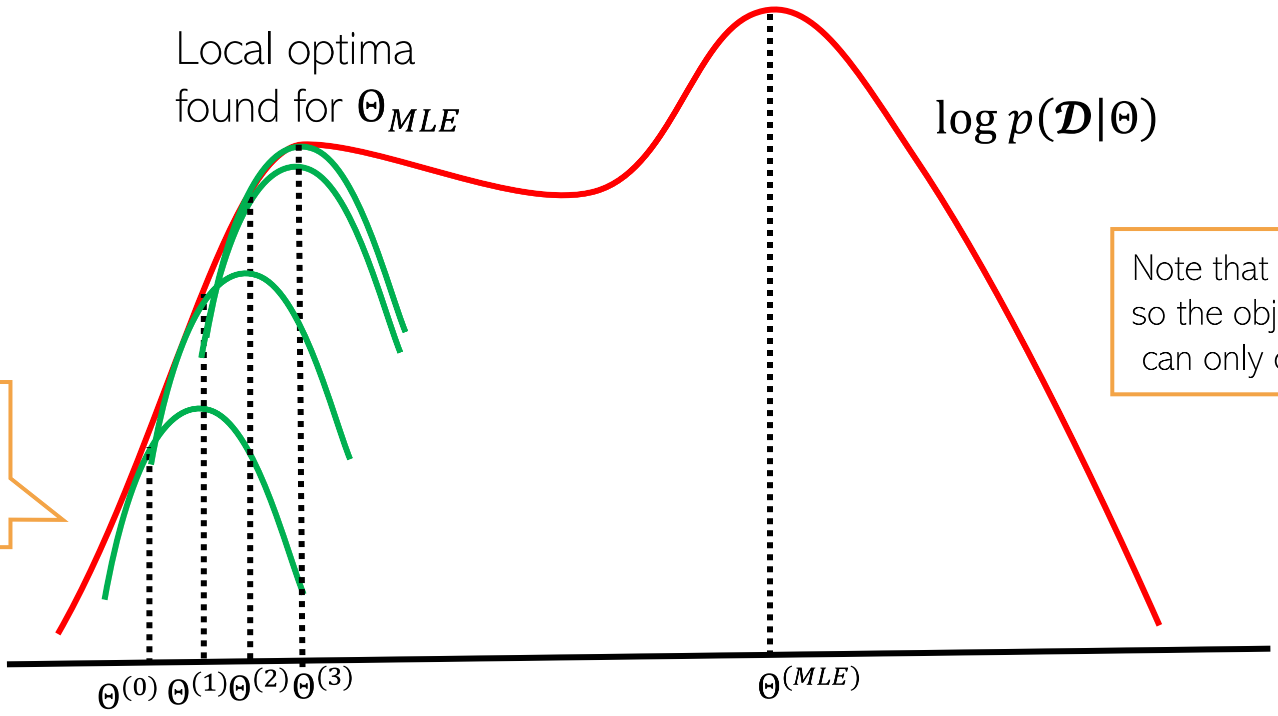
EM is guaranteed to converge!

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

- Maximization of lower bound $\mathcal{L}(q, \Theta)$ alternates between these two steps
  - Step 1 sets $q^{(t)} = p(\boldsymbol{Z}|\mathcal{D}, \Theta^{(t-1)})$ so $KL(q||p_Z)$ becomes zero, and red and green curves touch at $\Theta^{(t-1)}$
  - Step 2 sets $\Theta^{(t)} = \text{argmax}_\Theta\ \mathcal{L}(q^{(t)}, \Theta) = \text{argmax}_\Theta\ \mathbb{E}_{q^{(t)}}[\log p(\mathcal{D}, \boldsymbol{Z}|\Theta)] = \text{argmax}_\Theta\ \mathcal{Q}(\Theta, \Theta^{(t-1)})$

Red curve: $\log p(\mathcal{D}|\Theta)$

Green curve: $\mathcal{L}(q^{(t)}, \Theta)$



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

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

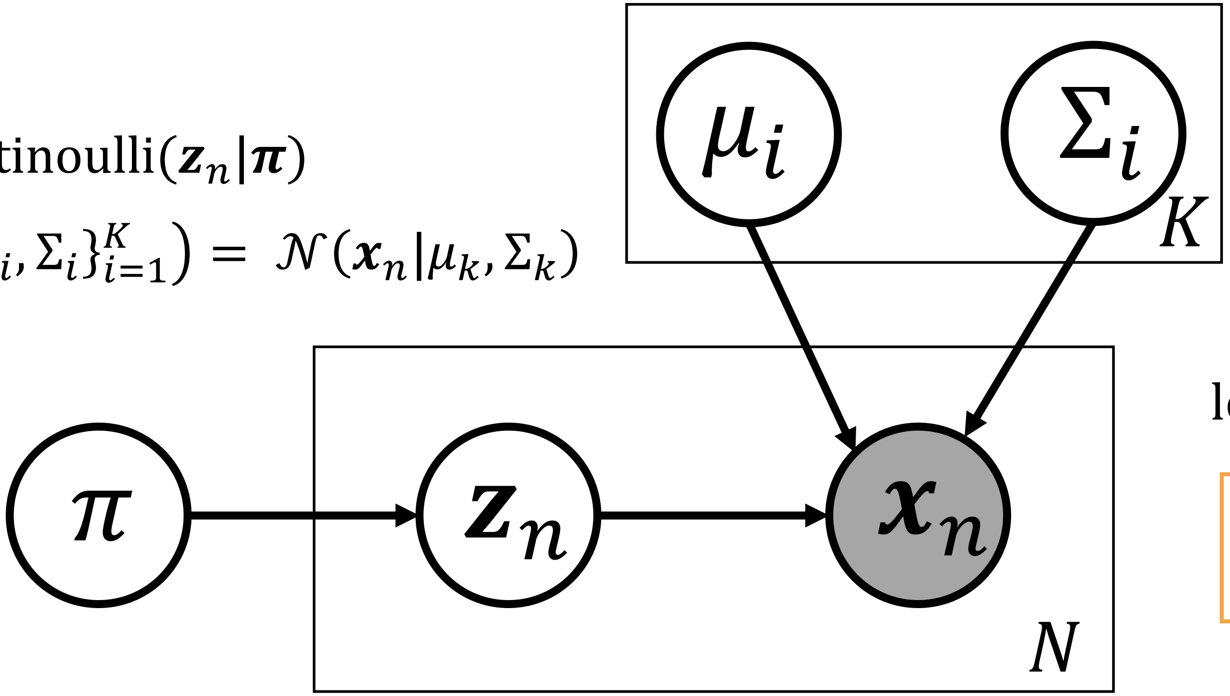
- EM is guaranteed to converge (possibly to a local optima)

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 $\boldsymbol{z}_n \in \{1,2,\dots,K\}$ 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

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

multinoulli

Gaussian

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

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

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

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

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

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"

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

Same as writing $z_n = k$

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

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

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

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

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

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

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

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

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

4. Go to step 2 if not converged

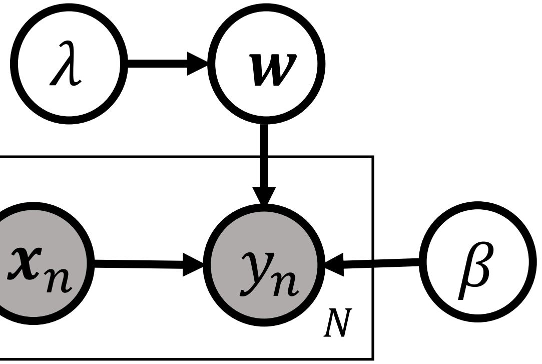
Bayesian Linear Regression (Revisited)

$N \times D$ 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. $\boldsymbol{w}, \beta, \lambda$ 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 $\quad (\hat{\beta}, \hat{\lambda}) = \text{argmax}_{\beta,\lambda} \log p(\boldsymbol{y}|\boldsymbol{X}, \beta, \lambda)$

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

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

EM for Bayesian Linear Regression

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

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 $\boldsymbol{w}$

Extra: MLE-II for Bayesian Lin. Reg.

- The MLE-II problem for Bayesian linear regression

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

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

- 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.

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

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

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

where

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

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

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

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

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

  - 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. $\boldsymbol{\Theta}$

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

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

- The above problem may be difficult to solve if one/both of the following is true
  1. CP $p\left(\boldsymbol{Z}\middle|\Theta^{(t-1)}, \mathcal{D}\right)$ 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 $p\left(\boldsymbol{Z}\middle|\Theta^{(t-1)}, \mathcal{D}\right)$

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

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

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

- Monte-Carlo approximation is commonly used in such problems