Expectation Maximization (Contd)

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

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

The Expected CLL

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

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

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

Was indeed the case of GMM: $p(\boldsymbol{z}_n|\Theta)$ was multinoulli, $p(\boldsymbol{x}_n|\boldsymbol{z}_n,\Theta)$ was Gaussian

- If $p(\boldsymbol{z}_n|\Theta)$ and $p(\boldsymbol{x}_n|\boldsymbol{z}_n,\Theta)$ are exp-family distributions, $\mathcal{Q}(\Theta, \Theta^{\text{old}})$ 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 $\mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n,\hat{\Theta})}[\boldsymbol{z}_n]$
  - $\boldsymbol{z}_n\boldsymbol{z}_n^\top$ replaced by $\mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n,\hat{\Theta})}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- However, in some LVMs, these expectations are intractable to compute and need to be approximated (will see some examples later)

What's Going On?

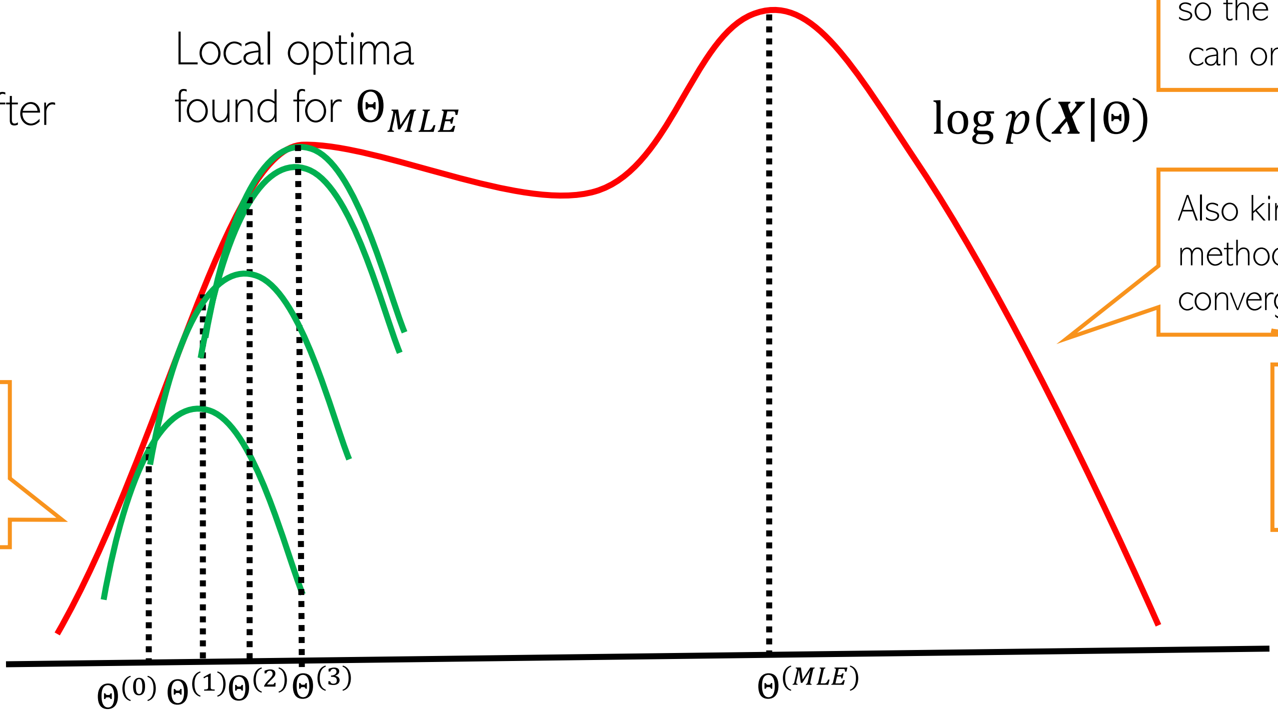
Alternating between them until convergence to some local optima

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 $\hat{q}$



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

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

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

Online/Incremental EM

Have to do it in each iteration of EM

- Computing CP of latent variable $\boldsymbol{z}_n$ of each observation $\boldsymbol{x}_n$ can be expensive
  - Before we do the M step to update $\boldsymbol{\Theta}$, we must wait for all CPs to be computed ☹

- Recall that, for i.i.d. case, the expected CLL is often a sum over all data points

$$\mathcal{Q}(\Theta, \Theta^{old}) = \mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta) = \sum_{n=1}^{N} \mathbb{E}[\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)] + \mathbb{E}[\log p(\boldsymbol{z}_n|\phi)]$$

Only requires CP for the latent variables from this minibatch of observations

- Can compute this quantity recursively using small minibatches of data

Expected CLL from iteration $t-1$

Expected CLL from this minibatch

$$\mathcal{Q}_t = (1-\gamma_t)\mathcal{Q}_{t-1} + \gamma_t \left[\sum_{n=1}^{N_t} \mathbb{E}[\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)] + \mathbb{E}[\log p(\boldsymbol{z}_n|\phi)]\right]$$

where $\gamma_t = (1+t)^{-\kappa}, 0.5 < \kappa \leq 1$ is a decaying learning rate

- MLE on above $\mathcal{Q}_t$ can be shown to be equivalent to a simple recursive updates for $\boldsymbol{\Theta}$

$$\Theta^{(t)} = (1-\gamma_t) \times \Theta^{(t-1)} + \gamma_t \times \arg\max_{\Theta} \underbrace{\mathcal{Q}(\Theta, \Theta^{t-1})}_{\substack{\text{computed using only} \\ \text{the } N_t \text{ examples} \\ \text{from this minibatch}}}$$

How M Step uses Sufficient Statistics

- Recall the batch EM algorithm for a K component Gaussian mixture model
  - Cluster id z_n s.t. $z_{nk} = 1$ if x_n belongs to cluster k, and 0 otherwise
  - The conditional posterior of z_{nk} is $p(z_{nk} = 1 | x_n, \Theta) \propto \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)$
- Denoting current iteration by t, and expectation computed in E step: $\mathbb{E}\left[z_{nk}^{(t)}\right] = \gamma_{nk}^{(t)}$
- The M step updates for params $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ are

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

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

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

- Each update depends on sum of expected sufficient statistics (ESS). For each x_n, z_n

ESS for $\boldsymbol{\mu}_k$ is $\gamma_{nk}^{(t)} \boldsymbol{x}_n$; ESS for $\boldsymbol{\Sigma}_k$ is $\gamma_{nk}^{(t)} (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})^\top$; ESS for π_k is $\gamma_{nk}^{(t)}$

Batch EM in terms of Sufficient Statistics

- Denote the sum of ESS as $\mathbf{S} = \sum_{n=1}^{N} \boldsymbol{s}_n$ where each ESS

SS for $\boldsymbol{x}_n$ and $\boldsymbol{z}_n$

$$\boldsymbol{s}_n = \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\Theta, \boldsymbol{x}_n)\phi(\boldsymbol{x}_n, \boldsymbol{z}_n)$$

- M step updates of Θ are like computing a function of $\mathbf{S}$, i.e., $\Theta = f(\mathbf{S})$

Batch EM in terms of ESS

- Initialize $\mathbf{S}$ and compute parameters $\Theta = f(\mathbf{S})$
- For $t = 1 : T$ (or until convergence)
  - $\mathbf{S}^{new} = 0$ (fresh sum of ESS; will be computed in this iteration)
  - For $n = 1 : N$

$$\begin{aligned} \boldsymbol{s}_n &= \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \Theta)\phi(\boldsymbol{x}_n, \boldsymbol{z}_n) = \mathbb{E}[\phi(\boldsymbol{x}_n, \boldsymbol{z}_n)] \\ \mathbf{S}^{new} &= \mathbf{S}^{new} + \boldsymbol{s}_n \end{aligned}$$

  - $\mathbf{S} = \mathbf{S}^{new}$
  - Recompute parameters $\Theta = f(\mathbf{S})$

- Note: In general, there may be more than one sum of ESS (one for each param)
  - E.g., for GMM, one for π_k, one for μ_k, one for Σ_k

Online EM in terms of Sufficient Statistics

- Works in a similar way as batch EM except we perform online updates for $\mathbf{S}$
- Can be done in one of the two manners (Liang and Klein, 2009)
  - Stepwise EM (based on recursively updating the sum of ESS)
  - Incremental EM (based on deleting old and adding new ESS of each data point)

Online EM as Stepwise EM

- Initialize the sum of ESS $\mathbf{S}$ and compute $\Theta = f(\mathbf{S})$
- For $t = 1 : T$ (or until convergence)
  - Set "learning rate" γ_t, pick a random example n and compute its sufficient statistics

$$
\begin{aligned}
\boldsymbol{s}_n &= \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \Theta)\phi(\boldsymbol{x}_n, \boldsymbol{z}_n) \\
\mathbf{S} &= (1-\gamma_t)\mathbf{S} + \gamma_t \boldsymbol{s}_n
\end{aligned}
$$

  - Recompute $\Theta = f(\mathbf{S})$

Online EM in terms of Sufficient Statistics

- The other Online EM approach "Incremental EM" needs no learning rate

Online EM as Incremental EM

- Initialize each ESS $\boldsymbol{s}_n$, $n = 1, \ldots, N$, $\mathbf{S} = \sum_{n=1}^{N} \boldsymbol{s}_n$, and compute $\Theta = f(\mathbf{S})$
- For $t = 1 : T$ (or until convergence)
    - Pick a random example n and update its exp. sufficient statistics

$$\begin{aligned} \boldsymbol{s}_n^{new} &= \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n | \boldsymbol{x}_n, \Theta) \phi(\boldsymbol{x}_n, \boldsymbol{z}_n) \\ \mathbf{S} &= \mathbf{S} + \boldsymbol{s}_n^{new} - \boldsymbol{s}_n \\ \boldsymbol{s}_n &= \boldsymbol{s}_n^{new} \end{aligned}$$

    - Recompute $\Theta = f(\mathbf{S})$

- However, incremental EM requires keeping a track of sum of ESS $\boldsymbol{S}$ as well as each $\boldsymbol{s}_n$
- In practice, stepwise EM outperforms batch EM as well as incremental EM on many problems (can refer to Liang and Klein, 2009 for some examples of models where these algos were tried)

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 (each data-point comes from one of K mixture components)
  - Examples: Mixture of Gaussians, Mixture of Experts, etc
- Latent variable models for dimensionality reduction or representation learning
  - Probabilistic PCA, Factor Analysis, Variational Autoencoders, etc
- Problems models with missing features/labels (treated as latent variables)
  - An example of problem with missing labels: Semi-supervised learning
- 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

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

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

EM: Some 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

Results in Monte-Carlo EM

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

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 Bayes (VB) inference, a.k.a. Variational Inference (VI)

Coming Up Next

- Variational Inference