CPSC 440: Advanced Machine Learning

More EM

Mark Schmidt

University of British Columbia

Winter 2021

Last Time: Expectation Maximization

- EM considers learning with observed data O and hidden data H.
  - Treating the hidden/missing data H as nuissance variables.
- In this case the "marginal" log-likelihood has a nasty form,

$$\log p(O \mid \Theta) = \log\left(\sum_H p(O, H \mid \Theta)\right).$$

- EM applies when "complete" likelihood, $p(O, H \mid \Theta)$, has a nice form.
- EM iterations take the form of a weighted "complete" NLL,

$$\Theta^{t+1} = \underset{\Theta}{\text{argmax}}\left\{\sum_H \alpha_H^t \log p(O, H \mid \Theta)\right\},$$

  for a specific choice of the convex combination coefficients α_H^t (today).
- We looked at the simple form of the EM update for Gaussian mixture models.
  - Video: `https://www.youtube.com/watch?v=B36fzChfyGU`

Digression: z^i vs r_c^i vs π_c for Mixtures

- For mixtures models we have discussed the quantities z^i, r_c^i, and π_c.
  - Many students (myself included) get these confused when learning.
- Mixtures assume each example x^i is generated by exactly one of the mixtures.
  - And I use "mixture" and "cluster" interchangeably.
- z^i is a nuisance parameter that is mixture number that generated example i.
  - So if $k = 3$ then z^i is either 1, 2, or 3.
- π_c is a parameter giving our estimate of the proportion of examples in cluster c.
  - So if $\pi_2 = 0.3$, we think that 30% of our examples come from cluster 2.
- r_c^i is the probability that example i came from mixture c (given parameters).
  - It's a quantity that appears when doing calculations with mixture models.
    - In EM, but also when you want to guess which cluster generated an example.

Expectation Maximization Bound

- Each iteration of EM and imputation optimize the approximation:

$$\Theta^{t+1} \in \underset{\Theta}{\text{argmin}} - \sum_{H} \alpha_H^t \log p(O, H \mid \Theta).$$

  where the probabilities α_H^t are updated after each iteration t.

- Imputation sets $\alpha_H^t = 1$ for the most likely H given Θ^t (all other $\alpha_H^t = 0$).
  - It assumes that the imputations are correct, then optimizes with the guess

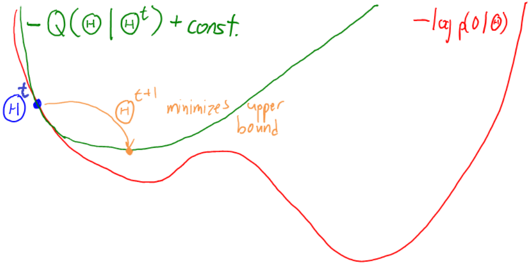
- In EM we set $\alpha_H^t = p(H \mid O, \Theta^t)$, weighting H by probability given Θ^t.
  - It weighs different imputations by their probability, then optimizes.

Expectation Maximization as Bound Optimization

- We'll show that the EM approximation minimizes an upper bound,

$$-\underbrace{\log p(O \mid \Theta)}_{\text{what we want}} \leq \underbrace{-\sum_{H} p(H \mid O, \Theta^t) \log p(O, H \mid \Theta)}_{Q(\Theta \mid \Theta^t):\ \text{what we optimize}} + \text{const.},$$

- Geometry of expectation maximization as "optimizing an upper bound":
  - At each iteration t we optimize a bound on the function.



Expectation Maximization (EM)

- So EM starts with Θ^0 and sets Θ^{t+1} to maximize $Q(\Theta \mid \Theta^t)$.
- This is typically written as two steps:
  1. E-step: Define expectation of complete log-likelihood given last parameters Θ^t,

$$Q(\Theta \mid \Theta^t) = \sum_H \underbrace{p(H \mid O, \Theta^t)}_{\text{fixed weights } \alpha_H^t} \underbrace{\log p(O, H \mid \Theta)}_{\text{nice term}}$$
$$= \mathbb{E}_{H \mid O, \Theta^t}[\log p(O, H \mid \Theta)],$$

     which is a weighted version of the "nice" $\log p(O, H)$ values.
     - For mixtures of Gaussians, E-step updates r_c^i (like clustering step in k-means).
  2. M-step: Maximize this expectation to generate new parameters Θ^{t+1},

$$\Theta^{t+1} = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta \mid \Theta^t).$$

     - For mixture of Gaussians, M-step updates π_c, μ, and Σ_c (like mean in k-means).
- But I don't like the terms "E-step" and "M-step".
  - For mixture models it separates into two steps, but for many models it doesn't.

Expectation Maximization for Mixture Models

- In the case of a mixture model with extra "cluster" variables z^i, EM uses

$$Q(\Theta \mid \Theta^t) = \mathbb{E}_{z \mid X, \Theta^t}[\log p(X, z \mid \Theta)]$$

$$= \sum_{z^1=1}^{k} \sum_{z^2=1}^{k} \cdots \sum_{z^n=1}^{k} \underbrace{p(z \mid X, \Theta^t)}_{\alpha_z} \underbrace{\log p(X, z \mid \Theta)}_{\text{"nice"}} \quad (k^n \text{ terms})$$

$$= \sum_{z^1=1}^{k} \sum_{z^2=1}^{k} \cdots \sum_{z^n=1}^{k} \left(\prod_{i=1}^{n} p(z^i \mid x^i, \Theta^t)\right) \left(\sum_{i=1}^{n} \log p(x^i, z^i \mid \Theta)\right)$$

$$= \text{(see EM notes, tedious use of distributive law and independences)}$$

$$= \sum_{i=1}^{n} \sum_{z^i=1}^{k} p(z^i \mid x^i, \Theta^t) \log p(x^i, z^i \mid \Theta) \quad (nk \text{ terms}).$$

- Sum over k^n clusterings turns into sum over nk 1-example assignments.
  - Same simplification happens for semi-supervised learning, we'll discuss why later.

Expectation Maximization for Mixture Models

- In the case of a mixture model with extra "cluster" variables z^i EM uses

$$Q(\Theta \mid \Theta^t) = \sum_{i=1}^{n} \sum_{z^i=1}^{k} \underbrace{p(z^i \mid x^i, \Theta^t)}_{r_c^i} \log p(x^i, z^i \mid \Theta).$$

- This is just a weighted version of the usual log-likelihood.
  - Update is solution of a weighted Gaussian, weighted Bernoulli, and so on.
    - Closed-form solution in these simple cases.

- To derive the simple EM updates that were shown for mixture of Gaussians:
  - Take gradient of above and set it to 0, then solve for π_c, μ_c and Σ_c.
    - Then you re-compute responsibilities and repeat.

Discussing of EM for Mixtures of Gaussians

- EM and mixture models are used in a ton of applications.
  - One of the default unsupervised learning methods.
  - Not just for mixture models:
    - Semi-supervised learning.
    - Density estimation with missing values in matrix.
- EM usually doesn't reach global optimum.
  - Classic solution: restart the algorithm from different initializations.
  - Lots of work in CS theory on getting better initializations (like "k-means++").
- MLE for some clusters may not exist (e.g., only responsible for one point).
  - Use MAP estimates or remove these clusters.
- EM does not fix "propagation of errors" from imputation approach.
  - But it reduces problem by incorporating a "confidence" over different imputations.
- Can you make it robust?
  - Use mixture of Laplace of student t distributions.
  - Don't have closed-form EM steps: compute responsibilities then need to optimize.

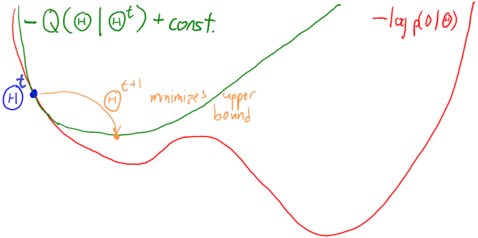
Outline

Monotonicity of EM

- Classic result is that EM iterations are monotonic:

$$\log p(O \mid \Theta^{t+1}) \geq \log p(O \mid \Theta^{t}),$$

- We don't need a step-size and this is useful for debugging.
- We can show this by proving that the below picture is "correct":



- The Q function leads to a global bound on the original function.
- At Θ^t the bound matches original function.
  - So if you improve on the Q function, you improve on the original function.

Monotonicity of EM

- Let's show that the Q function gives a global upper bound on NLL:

$$
\begin{aligned}
-\log p(O \mid \Theta) &= -\log\left(\sum_H p(O, H \mid \Theta)\right) && \text{(marginalization rule)}\\
&= -\log\left(\sum_H \alpha_H \frac{p(O, H \mid \Theta)}{\alpha_H}\right) && \text{(for } \alpha_H \neq 0\text{)}\\
&\leq -\sum_H \alpha_H \log\left(\frac{p(O, H \mid \Theta)}{\alpha_H}\right),
\end{aligned}
$$

because $-\log(z)$ is convex and the α_H are a convex combination.

Monotonicity of EM

- Using that log turns multiplication into addition we get

$$
\begin{aligned}
-\log p(O \mid \Theta) &\leq -\sum_H \alpha_H \log\left(\frac{p(O, H \mid \Theta)}{\alpha_H}\right) \\
&= \underbrace{-\sum_H \alpha_H \log p(O, H \mid \Theta)}_{Q(\Theta \mid \Theta^t)} + \underbrace{\sum_H \alpha_H \log \alpha_H}_{\text{negative entropy}} \\
&= -Q(\Theta \mid \Theta^t) - \text{entropy}(\alpha),
\end{aligned}
$$

  so we have the first part of the picture, $-\log p(O \mid \Theta^{t+1}) \leq -Q(\Theta|\Theta^t) + \text{const}$.
  - Entropy is a measure of how "random" the α_H values are.
  - Q behaves more like true objective for H that are more "predictable".

- Now we need to show that this holds with equality at Θ^t.

Bound on Progress of Expectation Maximization

- To show equality at Θ^t we use definition of conditional probability,

$$p(H \mid O, \Theta^t) = \frac{p(O, H \mid \Theta^t)}{p(O \mid \Theta^t)} \quad \text{or} \quad \log p(O \mid \Theta^t) = \log p(O, H \mid \Theta^t) - \log p(H \mid O, \Theta^t)$$

- Multiply by α_H and summing over H values,

$$\sum_H \alpha_H \log p(O \mid \Theta^t) = \underbrace{\sum_H \alpha_H \log p(O, H \mid \Theta^t}_{Q(\Theta^t \mid \Theta^t)} - \sum_H \alpha_H \log \underbrace{p(H \mid O, \Theta^t)}_{\alpha_H}.$$

- Which gives the result we want:

$$\log p(O \mid \Theta^t) \underbrace{\sum_H \alpha_H}_{=1} = Q(\Theta^t \mid \Theta^t) + \text{entropy}(\alpha),$$

Summary

- Expectation maximization:
  - Optimization with MAR variables, when knowing MAR variables make problem easy.
  - Instead of imputation, works with "soft" assignments to nuisance variables.
  - Maximizes log-likelihood, weighted by all imputations of hidden variables.
- Monotonicity of EM: EM is guaranteed not to decrease likelihood.
- Next time: generalizing histograms?

Alternate View of EM as BCD

- We showed that given α the M-step minimizes in Θ the function

$$F(\Theta, \alpha) = -\mathbb{E}_\alpha[\log p(O, H \mid \Theta)] - \text{entropy}(\alpha).$$

- The E-step minimizes this function in terms of α given Θ.
  - Setting $\alpha_H = p(H \mid O, \Theta)$ minimizes it.
- Note that F is not the NLL, but F and the NLL have same stationary points.
- From this perspective, we can view EM as a block coordinate descent method.
- This perspective is also useful if you want to do approximate E-steps.

Alternate View of EM as KL-Proximal

- Using definitions of expectation and entropy and α in the last slide gives

$$\begin{aligned}
F(\Theta,\alpha) &= -\sum_H p(H \mid O,\theta^t)\log p(O,H\mid\Theta) + \sum_H p(H\mid O,\theta^t)\log p(H\mid O,\theta^t)\\
&= -\sum_H p(H\mid O,\theta^t)\log\frac{p(O,H\mid\theta)}{p(H\mid O,\theta^t)}\\
&= -\sum_H p(H\mid O,\theta^t)\log\frac{p(H\mid O,\theta)p(O\mid\theta)}{p(H\mid O,\theta^t)}\\
&= -\sum_H \log p(O\mid\Theta) - \sum_H p(H\mid O,\theta^t)\log\frac{p(H\mid O,\theta)}{p(H\mid O,\theta^t)}\\
&= NLL(\Theta) + \mathsf{KL}(p(H\mid O,\theta^t)\ ||\ p(H\mid O,\theta)).
\end{aligned}$$

- From this perspective, we can view EM as a "proximal point" method.
  - Classical proximal point method uses $\frac{1}{2}\|\theta^t-\theta\|^2$, EM uses KL divergence.
- From this view we can see that EM doesn't depend on parameterization of Θ.
- If we linearize NLL and we multiply KL term by $1/\alpha_k$ (step-size), we get the natural gradient method.