CPSC 540: Machine Learning

Expectation Maximization

Mark Schmidt

University of British Columbia

Winter 2018

Last Time: Learning with MAR Values

- We discussed learning with "missing at random" values in data:

$$X = \begin{bmatrix} 1.33 & 0.45 & -0.05 & -1.08 & ? \\ 1.49 & 2.36 & -1.29 & -0.80 & ? \\ -0.35 & -1.38 & -2.89 & -0.10 & ? \\ 0.10 & -1.29 & 0.64 & -0.46 & ? \\ 0.79 & 0.25 & -0.47 & -0.18 & ? \\ 2.93 & -1.56 & -1.11 & -0.81 & ? \\ -1.15 & 0.22 & -0.11 & -0.25 & ? \end{bmatrix}$$

- Imputation approach:
  - Guess the most likely value of each ?, fit model with these values (and repeat).

- K-means clustering algorithm is a special case:
  - Mixture of Gaussians with $\Sigma_c = I$ and ? being the cluster ($? \in \{1, 2, \cdots, k\}$).

Parameters, Hyper-Parameters, and Nuisance Parameters

- Are the ? values "parameters" or "hyper-parameters"?
- Parameters:
  - Variables in our model that we optimize based on the training set.
- Hyper-Parameters
  - Variables that control model complexity, typically set using validation set.
  - Often become degenerate if we set these based on training data.
  - We sometimes add optimization parameters in here like step-size.
- Nuisance Parameters
  - Not part of the model and not really controlling complexity.
  - An alternative to optimizing ("imputation") is to integrate over these values.
    - Consider all possible imputations, and weight them by their probability.

Expectation Maximization Notation

- Expectation maximization (EM) is an optimization algorithm for MAR values:
  - Applies to problems that are easy to solve with "complete" data (i.e., you knew ?).
  - Allows probabilistic or "soft" assignments to MAR (or other nuisance) variables.

- EM is among the most cited paper in statistics.
  - Imputation approach is sometimes called "hard" EM.

- EM notation: we use O as observed variables and H as hidden (?) variables.
  - Semi-supervised learning: observe $O = \{X, y, \bar{X}\}$ but don't observe $H = \{\bar{y}\}$.
  - Mixture models: observe data $O = \{X\}$ but don't observe clusters $H = \{z^i\}_{i=1}^n$.

- We use Θ as parameters we want to optimize.

Complete Data and Marginal Likelihoods

- Assume observing H makes "complete" likelihood $p(O, H \mid \Theta)$ "nice".
  - It has a closed-form MLE, gives a convex NLL, or something like that.

- From marginalization rule, likelihood of O in terms of "complete" likelihood is

$$p(O \mid \Theta) = \sum_{H_1} \sum_{H_2} \cdots \sum_{H_m} p(O, H \mid \Theta) = \sum_{H} \underbrace{p(O, H \mid \Theta)}_{\text{“complete likelihood”}} .$$

  where we sum (or integrate) over all possible $H \equiv \{H_1, H_2, \dots, H_m\}$.
  - For mixture models, this sums over all possible clusterings.

- The negative log-likelihood thus has the form

$$-\log p(O \mid \Theta) = -\log \left(\sum_{H} p(O, H \mid \Theta) \right),$$

- which has a sum inside the log.
  - This does not preserve convexity: minimizing it is usually NP-hard.

Expectation Maximization Bound

- To compute Θ^{t+1}, the approximation used by EM and hard-EM is

$$-\log\left(\sum_H p(O, H \mid \Theta)\right) \approx -\sum_H \alpha_H^t \log p(O, H \mid \Theta)$$

where α_H^t is a probability for the assignment H to the hidden variables.
  - Note that α_H^t changes on each iteration t.

- In hard-EM we set $\alpha_H^t = 1$ for the most likely H given Θ^t (all other $\alpha_H^t = 0$).

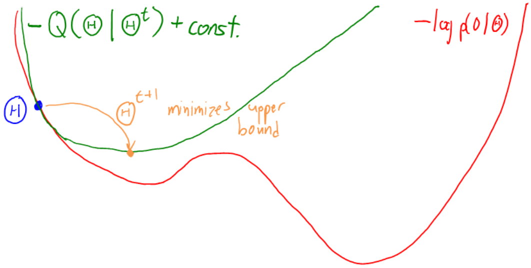
- In soft-EM we set $\alpha_H^t = p(H \mid O, \Theta^t)$, weighting H by probability given Θ^t.

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

$$-\log p(O \mid \Theta) \leq \underbrace{-\sum_H p(H \mid O, \Theta^t) \log p(O, H \mid \Theta)}_{Q(\Theta \mid \Theta^t)} + \text{const.},$$

Expectation Maximization as Bound Optimization

- Expectation maximization is a "bound-optimization" method:
  - At each iteration t we optimize a bound on the function.



- In gradient descent, our bound came from Lipschitz-continuity of the gradient.
- In EM, our bound comes from expectation over hidden variables (non-quadratic).

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,
$$
\begin{aligned}
Q(\Theta \mid \Theta^t) &= \sum_H \underbrace{p(H \mid O, \Theta^t)}_{\text{fixed weights } \alpha_H^t} \underbrace{\log p(O, H \mid \Theta)}_{\text{nice term}} \\
&= \mathbb{E}_{H \mid O, \Theta^t}[\log p(O, H \mid \Theta)],
\end{aligned}
$$
     which is a weighted version of the "nice" $\log p(O, H)$ values.
  2. M-step: Maximize this expectation to generate new parameters Θ^{t+1},
$$
\Theta^{t+1} = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta \mid \Theta^t).
$$

Expectation Maximization for Mixture Models

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

$$
\begin{aligned}
Q(\Theta \mid \Theta^t) &= \mathbb{E}_{z \mid X, \Theta}[\log p(X, z \mid \Theta)] \\
&= \sum_{z^1=1}^{k} \sum_{z^2=1}^{k} \cdots \sum_{z^n=1}^{k} \underbrace{p(z \mid X, \Theta^t)}_{\alpha_z} \underbrace{\log p(X, z \mid \Theta)}_{\text{"nice"}} \\
&= \sum_{z^1=1}^{k} \sum_{z^2=1}^{k} \cdots \sum_{z^n=1}^{k} \left(\prod_{i=1}^{n} p(z^i \mid x^i, \Theta^t) \right) \left(\sum_{i=1}^{n} \log p(x^i, z^i \mid \Theta) \right) \\
&= \text{(see EM notes, tedious use of distributive law and independences)} \\
&= \sum_{i=1}^{n} \sum_{z^i=1}^{k} p(z^i \mid x^i, \Theta^t) \log p(x^i, z^i \mid \Theta).
\end{aligned}
$$

- 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 likelihood.
  - We just need to do MLE in weighted Gaussian, weighted Bernoulli, etc.

- We typically write update in terms of responsibilitites,

$$r_c^i \triangleq p(z^i = c \mid x^i, \Theta^t) = \frac{p(x^i \mid z^i = c, \Theta^t) p(z^i = c \mid \Theta^t)}{p(x^i \mid \Theta^t)} \quad \text{(Bayes rule)},$$

the probability that cluster c generated x^i.
  - By marginalization rule, $p(x^i \mid \Theta^t) = \sum_{c=1}^{k} p(x^i \mid z^i = c, \Theta^t) p(z^i = c' \mid \Theta^t)$.
  - We get k-means if $r_c^i = 1$ for most likely cluster and 0 otherwise.

Expectation Maximization for Mixture of Gaussians

- For mixture of Gaussians, E-step computes all r_c^i and M-step minimizes the weighted NLL:

$$\pi_c^{t+1} = \frac{1}{n}\sum_{i=1}^{n} r_c^i \qquad \text{(proportion of examples soft-assigned to cluster } c\text{)}$$

$$\mu_c^{t+1} = \frac{\sum_{i=1}^{n} r_c^i x^i}{\sum_{i=1}^{n} r_c^i} \qquad \text{(mean of examples soft-assigned to cluster } c\text{)}$$

$$\Sigma_c^{t+1} = \frac{\sum_{i=1}^{n} r_c^i (x^i - \mu_c^{t+1})(x^i - \mu_c^{t+1})^T}{\sum_{i=1}^{n} r_c^i} \qquad \text{(covariance of examples soft-assigned to } c\text{).}$$

- Now you would compute new responsibilities and repeat.
  - Notice that there is no step-size.

- EM for fitting mixture of Gaussians in action:
  `https://www.youtube.com/watch?v=B36fzChfyGU`

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.
- EM usually doesn't reach global optimum.
  - Classic solution: restart the algorithm from different initializations.
- MLE for some clusters may not exist (e.g., only responsible for one point).
  - Use MAP estimates or remove these clusters.
- How do you choose number of mixtures k?
  - Use cross-validation or other model selection criteria.
- Can you make it robust?
  - Use mixture of Laplace of student t distributions.
- Are there alternatives to EM?
  - Could use gradient descent on NLL.
  - Spectral and other recent methods have some global guarantees.

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.
- Next time: the sad truth about rain in Vancouver.

Generative Mixture Models and Mixture of Experts

- Classic generative model for supervised learning uses

$$p(y^i \mid x^i) \propto p(x^i \mid y^i)p(y^i),$$

  and typically $p(x^i \mid y^i)$ is assumed Gaussian (LDA) or independent (naive Bayes).
- But we could allow more flexibility by using a mixture model,

$$p(x^i \mid y^i) = \sum_{c=1}^{k} p(z^i = c \mid y^i)p(x^i \mid z^i = c, y^i).$$

- Another variation is a mixture of discriminative models (like logistic regression),

$$p(y^i \mid x^i) = \sum_{c=1}^{k} p(z^i = c \mid x^i)p(y^i \mid z^i = c, x^i).$$

- Called a "mixture of experts" model:
  - Each regression model becomes an "expert" for certain values of x^i.