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 $\Theta$ 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 $H \equiv \{H_1, H_2, \ldots, H_m\}$.
  - 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

**Expectation Maximization Bound**

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

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

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

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

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

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

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

where $Q(\Theta \mid \Theta^t)$ is the upper bound.
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).
So EM starts with $\Theta^0$ and sets $\Theta^{t+1}$ 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 $\Theta^t$,

   $$Q(\Theta \mid \Theta^t) = \sum_H p(H \mid O, \Theta^t) \log p(O, H \mid \Theta)$$

   where

   $$\sum_H \underbrace{p(H \mid O, \Theta^t)}_{\text{fixed weights } \alpha^t_H} \underbrace{\log p(O, H \mid \Theta)}_{\text{nice term}}$$

   $$= \mathbb{E}_{H \mid O, \Theta^t} \left[ \log p(O, H \mid \Theta) \right],$$

   which is a weighted version of the “nice” $\log p(O, H)$ values.

2. **M-step**: Maximize this expectation to generate new parameters $\Theta^{t+1}$,

   $$\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta \mid \Theta^t).$$
expectation maximization

**Expectation Maximization for Mixture Models**

- In the case of a *mixture model* with extra “cluster” variables $z^i$, EM uses

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

  \[
  = \sum_{z^1=1}^k \sum_{z^2=1}^k \cdots \sum_{z^n=1}^k p(z \mid X, \Theta^t) \log p(X, z \mid \Theta)
  \]

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

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

  \[
  = \sum_{i=1}^n \sum_{z^i=1}^k p(z^i \mid x^i, \Theta^t) \log p(x^i, z^i \mid \Theta).
  \]

- **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

Expectation Maximization for Mixture Models

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

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

- 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,

$$r^i_c \triangleq p(z^i = c \mid x^i, \Theta^t) = \frac{p(x^i \mid z^i = c, \Theta^t)p(z^i = c \mid \Theta^t)}{p(x^i \mid \Theta^t)}$$

  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 $r^i_c = 1$ for most likely cluster and 0 otherwise.
Expectation Maximization for Mixture of Gaussians

- For mixture of Gaussians, E-step computes all $r^i_c$ and M-step minimizes the weighted NLL:

$$\pi^{t+1}_c = \frac{1}{n} \sum_{i=1}^{n} r^i_c$$  
(proportion of examples soft-assigned to cluster $c$)

$$\mu_{c}^{t+1} = \frac{\sum_{i=1}^{n} r^i_c x^i}{\sum_{i=1}^{n} r^i_c}$$  
(mean of examples soft-assigned to cluster $c$)

$$\Sigma_{c}^{t+1} = \frac{\sum_{i=1}^{n} r^i_c (x^i - \mu_{c}^{t+1})(x^i - \mu_{c}^{t+1})^T}{\sum_{i=1}^{n} r^i_c}$$  
(covariance of examples soft-assigned to $c$).

- 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
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 \( x^i \).