CSE 250A. Principles of AI

Probabilistic Reasoning and Decision-Making

Lecture 11 – The EM Algorithm

Lawrence Saul
Department of Computer Science and Engineering
University of California, San Diego

Fall 2021

Outline

How to maximize $f(\vec{\theta})$?

1. **Gradient ascent**

$$\vec{\theta} \leftarrow \vec{\theta} + \eta \left(\frac{\partial f}{\partial \vec{\theta}} \right)$$

× Tedious to tune η.
× Not monotonically convergent.

2. **Newton's method**

$$\vec{\theta} \leftarrow \vec{\theta} - \mathbf{H}^{-1} \left(\frac{\partial f}{\partial \vec{\theta}} \right)$$

× Expensive for large problems.
× Fast but unstable.

3. **Auxiliary function**

$$\vec{\theta}_{\text{new}} = \underset{\vec{\theta}}{\operatorname{argmax}}\, Q(\vec{\theta}, \vec{\theta}_{\text{old}})$$

✓ No learning rate.
✓ Monotonically convergent.

Auxiliary functions

- **Definition**

  A function $Q(\vec{\theta}', \vec{\theta})$ is called an auxiliary function for $f(\vec{\theta})$ if it satisfies two properties:

  $$\text{(i)}\ Q(\vec{\theta}, \vec{\theta}) = f(\vec{\theta}) \text{ for all } \vec{\theta} \qquad \boxed{\textbf{equality}}$$

  $$\text{(ii)}\ Q(\vec{\theta}', \vec{\theta}) \leq f(\vec{\theta}') \text{ for all } \vec{\theta}, \vec{\theta}' \qquad \boxed{\textbf{lower bound}}$$

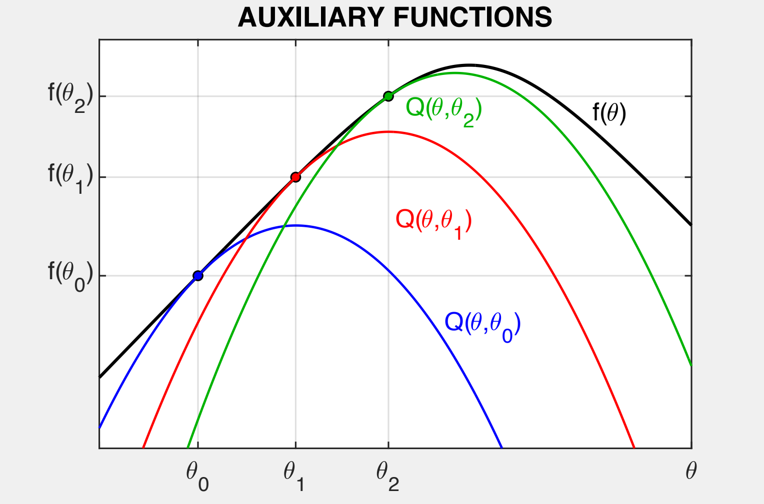
- **Theorem**

  Let $Q(\vec{\theta}', \vec{\theta})$ be an auxiliary function for $f(\vec{\theta})$.
  Then the update rule

  $$\vec{\theta}_{\text{new}} = \operatorname*{argmax}_{\vec{\theta}} Q(\vec{\theta}, \vec{\theta}_{\text{old}})$$

  converges monotonically with $f(\vec{\theta}_{\text{new}}) \geq f(\vec{\theta}_{\text{old}})$.

Visualization



Learning from incomplete data with tabular CPTs

- **Assumptions**

  The DAG is fixed over discrete nodes $\{X_1, \ldots, X_n\}$.
  The CPTs enumerate $P(X_i = x | \mathrm{pa}(X_i) = \pi)$ as lookup tables.
  IID data consists of T partially complete instantiations.

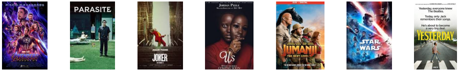
- **Notation**

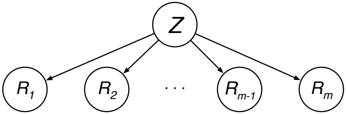
  H_t denotes the set of hidden nodes for the t^{th} example.
  V_t denotes the set of visible nodes for the t^{th} example.

- **Problem**

  How to choose CPTs to maximize $\mathcal{L} = \sum_{t=1}^{T} \log P(V_t = v_t)$,
  the incomplete-data log-likelihood?

Naive Bayes model with incomplete data





- **Movie recommender system**

$$
\begin{array}{lll}
Z & \in & \{1, 2, \ldots, k\} \quad \text{type of movie-goer} \\
R_i & \in & \{0, 1\} \quad \text{rating for } i^{\text{th}} \text{ movie}
\end{array}
$$

- **Incomplete data set**

student	Z	R_1	R_2	R_3	R_4	$\cdots$
1	**?**	0	1	1	**?**	$\cdots$
2	**?**	1	**?**	0	1	$\cdots$
3	**?**	0	0	**?**	1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T	**?**	**?**	1	0	**?**	$\cdots$

Note that the variable Z is **never observed**.

Outline

1. **Review**

2. **EM Algorithm**

   Procedural description and intuition

   Formal derivation and properties

Next week — many concrete examples ...

EM algorithm in a nutshell

- **If only the data weren't incomplete ...**

student	Z	R_1	R_2	$\cdots$
1	**?**	0	1	$\cdots$
2	**?**	1	**?**	$\cdots$
3	**?**	0	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T	**?**	**?**	**?**	$\cdots$

If the data were complete, we could easily estimate the CPTs. What can we do instead?

- **Here's a crazy idea ...**

Randomly initialize the CPTs with nonzero elements.
Use these CPTs to infer values for the **missing data**.
Re-estimate CPTs from the newly completed data.
Iterate the last two steps until convergence?

Amazingly, this is how EM works (more or less) ...

EM algorithm — overview

- **Initialize the CPTs**

  Assign random probabilities to all $P(X_i = x | \mathrm{pa}_i = \pi)$.

  Avoid zero probabilities (which cannot be unlearned).

  Different initializations may yield different results.

- **Iterate until convergence**

  [E-Step] Compute posterior probabilities $P(H_t = h | V_t = v_t)$.

  [M-Step] Update CPTs based on these probabilities.

E-step (Inference)

To fill in missing data, we must compute posterior probabilities.
But which probabilities, specifically, do we need?

At root nodes: $P(X_i = x | V_t = v_t)$
At other nodes: $P(X_i = x, \mathrm{pa}_i = \pi | V_t = v_t)$

These probabilities must be computed over a quadruple loop:

examples V_t	$t \in \{1, 2, \ldots, T\}$
nodes X_i	$i \in \{1, 2, \ldots, n\}$
values of $X_i = x$	e.g., $x \in \{0, 1\}$
values of $\mathrm{pa}_i = \pi$	e.g., $\pi \in \{0, 1\}^k$

The # of computations grows linearly in the size of the BN, and also in the amount of data (as expected).

M-step (Learning)

Next we use these posterior probabilities to update CPTs:

- **At root nodes**

$$P(X_i\!=\!x) \;\longleftarrow\; \frac{1}{T}\sum_{t=1}^{T} P(X_i\!=\!x|V_t\!=\!v_t)$$

- **At nodes with parents**

$$P(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) \;\longleftarrow\; \frac{\sum_{t=1} P(X_i\!=\!x, \mathrm{pa}_i\!=\!\pi|V_t\!=\!v_t)}{\sum_{t=1}^{T} P(\mathrm{pa}_i\!=\!\pi|V_t\!=\!v_t)}$$

Note that these are updates ($\longleftarrow$), not equalities ($=$).
The right hand sides depend on the current CPTs.

Formulas are great, but what about intuition?

Analogy to ML for complete data

- **Indicator functions**

$$I(x, x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}$$

- **Counts**

$$\begin{aligned}
\mathrm{count}(X_i = x) &= \sum_{t=1}^{T} I(x_{it}, x) \\
\mathrm{count}(\mathrm{pa}_i = \pi) &= \sum_{t=1}^{T} I(\mathrm{pa}_{it}, \pi) \\
\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi) &= \sum_{t=1}^{T} I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)
\end{aligned}$$

ML estimates for complete data

- **At root nodes**

$$P_{\mathrm{ML}}(X_i\!=\!x) \;=\; \frac{\mathrm{count}(X_i\!=\!x)}{T}$$

$$P_{\mathrm{ML}}(X_i\!=\!x) \;=\; \frac{1}{T}\sum_{t=1}^{T} I(x_{it}, x)$$

- **At nodes with parents**

$$P_{\mathrm{ML}}(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) \;=\; \frac{\mathrm{count}(X_i\!=\!x, \mathrm{pa}_i\!=\!\pi)}{\mathrm{count}(\mathrm{pa}_i\!=\!\pi)}$$

$$P_{\mathrm{ML}}(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) \;=\; \frac{\sum_{t=1}^{T} I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)}{\sum_{t=1}^{T} I(\mathrm{pa}_{it}, \pi)}$$

Intuition for EM updates — by analogy

- **At root nodes**

$$P_{\mathrm{ML}}(X_i\!=\!x) \;=\; \frac{1}{T}\sum_t I(x_{it}, x) \qquad \boxed{\textbf{ML for complete data}}$$

$$P(X_i\!=\!x) \;\leftarrow\; \frac{1}{T}\sum_t P(X_i\!=\!x|V_t\!=\!v_t) \qquad \boxed{\textbf{EM update}}$$

- **At nodes with parents**

$$P_{\mathrm{ML}}(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) \;=\; \frac{\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)}{\sum_t I(\mathrm{pa}_{it}, \pi)} \qquad \boxed{\textbf{ML for complete data}}$$

$$P(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) \;\leftarrow\; \frac{\sum_t P(X_i\!=\!x, \mathrm{pa}_i\!=\!\pi|V_t\!=\!v_t)}{\sum_t P(\mathrm{pa}_i\!=\!\pi|V_t\!=\!v_t)} \qquad \boxed{\textbf{EM update}}$$

- **Special case**

Consider a CPT whose nodes are fully observed.
EM updates in this case reduce to ML estimates for complete data.

EM updates

$$P(X_i\!=\!x) \quad \longleftarrow \quad \frac{1}{T}\sum_t P(X_i\!=\!x|V_t\!=\!v_t)$$

root nodes

$$P(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) \quad \longleftarrow \quad \frac{\sum_t P(X_i\!=\!x, \mathrm{pa}_i\!=\!\pi|V_t\!=\!v_t)}{\sum_t P(\mathrm{pa}_i\!=\!\pi|V_t\!=\!v_t)}$$

nodes with parents

Intuitively:

When the data is complete, we estimate the CPTs from observed counts.

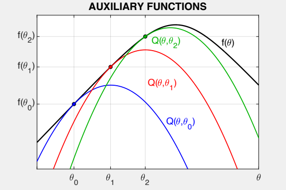
When the data is incomplete, we re-estimate the CPTs from expected counts.

These expected counts are computed from the posterior distributions $P(h|v_t)$.

Now versus later

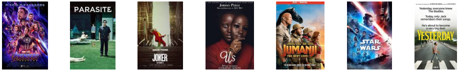
- **A reminder**

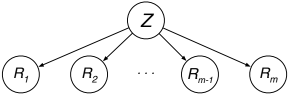
  Today's lecture is for developing intuition and proving key results.



- **A promise**

  The next lecture will be filled with practical examples and step-by-step algorithms.





Key properties of EM

- **No learning rate**

  The updates do not require the tuning of a learning rate ($\eta > 0$), as in most gradient-based methods.

- **Monotonic convergence**

  The updated CPTs from EM always increase the incomplete-data log-likelihood $\mathcal{L} = \sum_t \log P(V_t = v_t)$.

How do we prove this convergence?
By deriving an auxiliary function, of course.

Key inequality

Let $P(X)$ and $\tilde{P}(X)$ be different distributions over some set of nodes $X = \{X_1, X_2, \ldots, X_n\}$.

Let $V \subset X$ denote a subset of observed nodes.
Let $H \subset X$ denote the (complementary) subset of hidden nodes.

$$\begin{aligned}
\log \tilde{P}(v) &= 1 \cdot \log \tilde{P}(v) \\
&= \sum_h P(h|v) \log \tilde{P}(v) \\
&= \sum_h P(h|v) \log \frac{\tilde{P}(h,v)}{\tilde{P}(h|v)} \\
&= \sum_h P(h|v) \left[\log \frac{\tilde{P}(h,v)}{\tilde{P}(h|v)} + \log \frac{P(h|v)}{P(h|v)} \right]
\end{aligned}$$

Key inequality (con't)

Continuing the derivation:

$$\log \tilde{P}(v) = \sum_h P(h|v) \left[\log \frac{\tilde{P}(h,v)}{\tilde{P}(h|v)} + \log \frac{P(h|v)}{P(h|v)} \right]$$

$$= \sum_h P(h|v) \log \frac{\tilde{P}(h,v)}{P(h|v)} + \underbrace{\sum_h P(h|v) \log \frac{P(h|v)}{\tilde{P}(h|v)}}_{\textbf{KL distance!}}$$

$$\geq \sum_h P(h|v) \log \frac{\tilde{P}(h,v)}{P(h|v)} \qquad \boxed{\textbf{KL} \geq \textbf{0 by HW 1}}$$

This inequality holds for any instantiation v of observed nodes.
Now let's derive an auxiliary function for $\mathcal{L} = \sum_t \log P(v_t)$ ...

ML estimation for incomplete data

- **Notation**

  Let Θ denote the collection of CPTs in a BN.
  Let $\{v_t\}_{t=1}^{T}$ denote an incomplete data set for this BN.

- **Proposed objective and auxiliary functions**

$$\begin{aligned}\mathcal{L}(\Theta) &= \sum_t \log P(V_t = v_t) \\ Q(\tilde{\Theta}, \Theta) &= \sum_t \sum_h P(H_t = h | V_t = v_t) \log \frac{\tilde{P}(H_t = h, V_t = v_t)}{P(H_t = h | V_t = v_t)}\end{aligned}$$

- **What we need to check**

$$\begin{aligned}&\text{(i) } Q(\Theta, \Theta) = \mathcal{L}(\Theta) && \textbf{(equality)} \\ &\text{(ii) } Q(\tilde{\Theta}, \Theta) \leq \mathcal{L}(\tilde{\Theta}) && \textbf{(bound)}\end{aligned}$$

Auxiliary function properties

- **Auxiliary function**

$$Q(\tilde{\Theta}, \Theta) = \sum_t \sum_h P(H_t = h | V_t = v_t) \log \frac{\tilde{P}(H_t = h, V_t = v_t)}{P(H_t = h | V_t = v_t)}$$

- **Equality**

$$\begin{aligned} Q(\Theta, \Theta) &= \sum_t \sum_h P(H_t = h | V_t = v_t) \log \frac{P(H_t = h, V_t = v_t)}{P(H_t = h | V_t = v_t)} \\ &= \sum_t \sum_h P(H_t = h | V_t = v_t) \log P(V_t = v_t) \\ &= \sum_t \log P(V_t = v_t) \sum_h P(H_t = h | V_t = v_t) \\ &= \sum_t \log P(V_t = v_t) \cdot 1 \\ &= \mathcal{L}(\Theta) \quad \checkmark \end{aligned}$$

Auxiliary function properties (con't)

- **Bound**

$$
\begin{aligned}
Q(\tilde{\Theta}, \Theta) &= \sum_t \left[\sum_h P(H_t = h | V_t = v_t) \log \frac{\tilde{P}(H_t = h, V_t = v_t)}{P(H_t = h | V_t = v_t)} \right] \\
&\leq \sum_t \log \tilde{P}(V_t = v_t) \qquad \boxed{\textbf{by earlier inequality}} \\
&= \mathcal{L}(\tilde{\Theta}) \qquad \boxed{\textbf{by previous result}}
\end{aligned}
$$

We've shown that $Q(\tilde{\Theta}, \Theta)$ is an auxiliary function for $\mathcal{L}(\Theta)$. So what is the update derived from $Q(\tilde{\Theta}, \Theta)$?

It is exactly the update computed by the EM algorithm.

Formal statement of EM algorithm

- **E-step (Expectation)**

  Compute the auxiliary function, which can be viewed as a sum of expected values from the current CPTs:

$$
\begin{aligned}
Q(\tilde{\Theta}, \Theta) &= \sum_t \sum_h P(H_t = h | V_t = v_t) \log \frac{\tilde{P}(H_t = h, V_t = v_t)}{P(H_t = h | V_t = v_t)} \\
&= \sum_t \mathbf{E}_\Theta \left[\log \frac{\tilde{P}(H_t = h, V_t = v_t)}{P(H_t = h | V_t = v_t)} \middle| V_t = v_t \right]
\end{aligned}
$$

- **M-step (Maximization)**

  Choose the new CPTs by maximizing the first argument of the auxiliary function:

$$
\Theta_{\text{new}} = \underset{\Theta}{\text{argmax}} \left[Q(\Theta, \Theta_{\text{old}}) \right]
$$

Formal derivation of M-step

$$\operatorname*{argmax}_{\tilde{\Theta}} Q(\tilde{\Theta}, \Theta)$$

$$= \operatorname*{argmax}_{\tilde{\Theta}} \sum_t \sum_h P(h|v_t) \log \frac{\tilde{P}(h, v_t)}{P(h|v_t)}$$

$$= \operatorname*{argmax}_{\tilde{\Theta}} \sum_t \sum_h P(h|v_t) \log \tilde{P}(h, v_t)$$

$$= \operatorname*{argmax}_{\tilde{\Theta}} \sum_t \sum_h P(h|v_t) \log \prod_{i=1}^{n} \tilde{P}(X_i = x|\mathrm{pa}_i = \pi)\Big|_{H_t = h,\, V_t = v_t}$$

$$= \operatorname*{argmax}_{\tilde{\Theta}} \sum_i \sum_t \sum_h P(h|v_t) \log \tilde{P}(X_i = x|\mathrm{pa}_i = \pi)\Big|_{H_t = h,\, V_t = v_t}$$

$$= \operatorname*{argmax}_{\tilde{\Theta}} \sum_i \sum_t \sum_x \sum_\pi P(X_i = x, \mathrm{pa_i} = \pi|v_t) \log \tilde{P}(X_i = x|\mathrm{pa}_i = \pi)$$

Perhaps this looks bad, but you've solved this problem before ...

Complete versus incomplete data

- **Complete-data log-likelihood**

$$\mathcal{L}(\tilde{\Theta}) = \sum_i \sum_x \sum_\pi \mathrm{count}(X_i = x, \mathrm{pa_i} = \pi) \log \tilde{P}(X_i = x | \mathrm{pa}_i = \pi)$$

$$= \sum_i \sum_x \sum_\pi \left[\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi) \right] \log \tilde{P}(X_i = x | \mathrm{pa}_i = \pi)$$

- **Complete-data ML estimate**

$$P_{\mathrm{ML}}(X_i = x | \mathrm{pa}_i = \pi) = \frac{\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)}{\sum_t I(\mathrm{pa}_i, \pi)}$$

- **M-step for incomplete data**

$$\underset{\tilde{\Theta}}{\mathrm{argmax}} \sum_i \sum_x \sum_\pi \left[\sum_t P(X_i = x, \mathrm{pa}_i = \pi | v_t) \right] \log \tilde{P}(X_i = x | \mathrm{pa}_i = \pi)$$

Solution for EM update

- **ML estimation for complete data**

$$\mathcal{L}(\tilde{\Theta}) = \sum_i \sum_x \sum_\pi \left[\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi) \right] \log \tilde{P}(X_i = x | \mathrm{pa}_i = \pi)$$

$$\boxed{P_{\mathrm{ML}}(X_i = x | \mathrm{pa}_i = \pi) = \frac{\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)}{\sum_t I(\mathrm{pa}_i, \pi)}}$$

- **EM update for incomplete data**

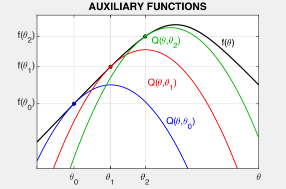
$$\underset{\tilde{\Theta}}{\operatorname{argmax}} \sum_i \sum_x \sum_\pi \left[\sum_t P(X_i = x, \mathrm{pa}_i = \pi | v_t) \right] \log \tilde{P}(X_i = x | \mathrm{pa}_i = \pi)$$

$$\boxed{P(X_i = x | \mathrm{pa}_i = \pi) \longleftarrow \frac{\sum_t P(X_i = x, \mathrm{pa}_i = \pi | v_t)}{\sum_t P(\mathrm{pa}_i = \pi | v_t)}}$$

Next lecture

- **A reminder**

  Today's lecture was for developing intuition and proving key results.



- **A promise**

  The next lecture will be filled with practical examples and step-by-step algorithms.