Speech Recognition
Lecture 7: Expectation-Maximization Algorithm, Hidden Markov Models

Cyril Allauzen
Google, NYU Courant Institute
allauzen@cs.nyu.edu
Slide Credit: Mehryar Mohri
This Lecture

- Expectation-Maximization (EM) algorithm
- Hidden-Markov Models
Latent Variables

**Definition:** unobserved or hidden variables, as opposed to those directly available at training and test time.

- example: mixture models, HMMs.

**Why latent variables?**

- naturally unavailable variables: e.g., did the patient take their medicine?
- modeling: latent variables introduced to model dependencies.
ML with Latent Variables

Problem:

• with fully observed variables, ML is often straightforward.

• with latent variables, log-likelihood contains a sum (harder to find the best parameter values):

\[ L(\theta, x) = \log p_\theta[x] = \log \sum_z p_\theta(x, z) = \log \sum_z p_\theta[z|x]p_\theta[x]. \]

Idea: use current parameter values to estimate latent variables and use those to re-estimate parameter values → EM algorithm.
**EM Idea**

- **Maximize expectation:**

\[
\sum_z p_\theta[z|x] \log p_{\theta'}[x, z] = \mathbb{E}_{z \sim p_\theta[\cdot|x]} \left[ \log p_{\theta'}[x, z] \right].
\]

- **Iterations:**
  - **compute** \( p_\theta[z|x] \) **for current value of** \( \theta \).
  - **compute expectation and find maximizing** \( \theta' \).
EM Algorithm

(Dempster, Laird, and Rubin, 1977; Wu, 1983)

- **Algorithm**: maximum-likelihood meta algorithm for models with latent variables.
  - **E-step**: $q^{t+1} \leftarrow p_{\theta^t}[z|x]$.
  - **M-step**: $\theta^{t+1} \leftarrow \arg \max_{\theta} \sum_z q^{t+1}(z|x) \log p_{\theta}[x, z]$.

- **Interpretation**:
  - **E-step**: posterior probability of latent variables given observed variables and current parameter.
  - **M-step**: maximum-likelihood parameter given all data.
EM Theorem

**Theorem:** for any $x$, and parameter values $\theta$ and $\theta'$,

$$L(\theta', x) - L(\theta, x) \geq \sum_z p_\theta[z|x] \log p_{\theta'}[x, z] - \sum_z p_\theta[z|x] \log p_\theta[x, z].$$

**Proof:**

$$L(\theta', x) - L(\theta, x) = \log p_{\theta'}[x] - \log p_\theta[x]$$

$$= \sum_z p_\theta[z|x] \log p_{\theta'}[x] - \sum_z p_\theta[z|x] \log p_\theta[x]$$

$$= \sum_z p_\theta[z|x] \log \frac{p_{\theta'}[x, z]}{p_{\theta'}[z|x]} - \sum_z p_\theta[z|x] \log \frac{p_\theta[x, z]}{p_\theta[z|x]}$$

$$= \sum_z p_\theta[z|x] \log \frac{p_{\theta}[z|x]}{p_{\theta'}[z|x]} + \sum_z p_\theta[z|x] \log p_{\theta'}[x, z] - \sum_z p_\theta[z|x] \log p_\theta[x, z]$$

$$= D(p_\theta[z|x] || p_{\theta'}[z|x]) + \sum_z p_\theta[z|x] \log p_{\theta'}[x, z] - \sum_z p_\theta[z|x] \log p_\theta[x, z]$$

$$\geq \sum_z p_\theta[z|x] \log p_{\theta'}[x, z] - \sum_z p_\theta[z|x] \log p_\theta[x, z].$$
EM Algorithm - Notes

**Applications:**
- mixture models, e.g., Gaussian mixtures. Latent variables: which model generated points.
- HMMs (Baum-Welch algorithm).

**Notes:**
- positive: each iteration increases likelihood. No parameter tuning.
- negative: can converge to local optima. Note that likelihood could converge but not parameter. Dependence on initial parameter.
EM - Extensions and Variants

(Dempster et al., 1977; Jamshidian and Jennrich, 1993; Wu, 1983)

- **Generalized EM (GEM):** maximization is not necessary at all steps since any (strict) increase of the auxiliary function guarantees an increase of the log-likelihood.

- **Sparse EM:** posterior probabilities computed at only some points in E-step.

(Dempster et al., 1977; Jamshidian and Jennrich, 1993; Wu, 1983)
This Lecture

- Expectation-Maximization (EM) algorithm
- Hidden-Markov Models
Motivation

- **Data**: sample of $m$ sequences over alphabet $\Sigma$ drawn i.i.d. according to some distribution $D$:

$$x^{i_1}, x^{i_2}, \ldots, x^{i_k}, \quad i = 1, \ldots, m.$$ 

- **Problem**: find sequence model that best estimates distribution $D$.

- **Latent variables**: observed sequences may have been generated by model with states and transitions that are hidden or unobserved.
Example

- **Observations:** sequences of Heads and Tails.
  
  observed sequence: $H, T, T, H, T, H, T, H, H, T, T, \ldots, H.$
  
  unobserved paths: $(0, H, 0), (0, T, 1), (1, T, 1), \ldots, (2, H, 2).$

- **Model:**

[Diagram showing states and transitions with probabilities]

$H/.3$  $T/.4$  $H/.6$  $H/.1$

$T/.7$  $1$  $T/.5$

$T/.4$  $0$  $2$
Hidden Markov Models

- **Definition**: probabilistic automata, generative view.
  - **discrete case**: finite alphabet $\Sigma$.
  - **transition probability**: $\Pr[\text{transition } e]$.
  - **emission probability**: $\Pr[\text{emission } a|e]$.
  - **simplification**: for us, $w[e] = \Pr[a|e] \Pr[e]$.

- **Illustration**:

```
0  \rightarrow  a/.3  \rightarrow  a/.2  \rightarrow  \cdots
  \downarrow a/.2  \rightarrow  \cdots
  \downarrow b/.1  \rightarrow  \cdots
  \downarrow c/.2  \rightarrow  \cdots
  \downarrow .5
```
Other Models

- **Outputs at states** (instead of transitions): equivalent model, dual representation.

- **Non-discrete case**: outputs in $\mathbb{R}^N$.
  - fixed probabilities replaced by distributions, e.g., Gaussian mixtures.
  - application to acoustic modeling.
Three Problems

Problem 1: given sequence $x_1, \ldots, x_k$ and hidden Markov model $p_\theta$ compute $p_\theta[x_1, \ldots, x_k]$.

Problem 2: given sequence $x_1, \ldots, x_k$ and hidden Markov model $p_\theta$ find most likely path $\pi = e_1, \ldots, e_r$ that generated that sequence.

Problem 3: estimate parameters $\theta$ of a hidden Markov model via ML: $\theta_* = \arg\max_\theta p_\theta[x]$.

(Rabiner, 1989)
Algorithm: let $X$ represent a finite automaton representing the sequence $x = x_1 \cdots x_k$ and let $H$ denote the current hidden Markov model.

Then, $p_{\theta}[x] = \sum_{\pi \in X \circ H} w[\pi]$.

Thus, it can be computed using composition and a shortest-distance algorithm in time $O(|X||H|)$. 
P2: Most Likely Path

- **Algorithm**: shortest-path problem in \((\max, \times)\) semiring.

  - any shortest-path algorithm applied to the result of the composition \(X \circ H\). In particular, since the automaton is acyclic, with a linear-time shortest-path algorithm, the total complexity is in \(O(|X \circ H|) = O(|X||H|)\).

  - a traditional solution is to use the Viterbi algorithm.
P3: Parameter Estimation

- **Baum-Welch algorithm**: special case of EM applied to HMMs.
  - Here, $\theta$ represents the transition weights $w[e]$.
  - The latent or hidden variables are paths $\pi$.
  - M-step:
    \[
    \arg\max_\theta \sum_{\pi} p_{\theta'}[\pi|x] \log p_{\theta}[x, \pi]
    \]
    subject to $\forall q \in Q, \sum_{e \in E[q]} w_{\theta}[e] = 1$. 

(Baum, 1972)
P3: Parameter Estimation

Using Lagrange multipliers $\lambda_q, q \in Q$, the problem consists of setting the following partial derivatives to zero:

$$\frac{\partial}{\partial w_\theta[e]} \left[ \sum_\pi p_\theta' [\pi | x] \log p_\theta [x, \pi] - \sum_q \lambda_q \sum_{e \in E[q]} w_\theta[e] \right] = 0$$

$$\Leftrightarrow \sum_\pi p_\theta' [\pi | x] \frac{\partial \log p_\theta [x, \pi]}{\partial w_\theta[e]} - \lambda_{\text{orig}}[e] = 0.$$ 

$p_\theta[x, \pi] \neq 0$ only for paths $\pi$ labeled with $x : \pi \in \Pi(x)$.

In that case, $p_\theta[x, \pi] = \prod w_\theta[e]^{\mid \pi \mid_e}$, where $\mid \pi \mid_e$ is the number of occurrences of $e$ in $\pi$. 
Thus, the equation with partial derivatives can be rewritten as

$$
\sum_{\pi \in \Pi(x)} p_{\theta'} [\pi | x] \frac{|\pi|_e}{w_{\theta}[e]} = \lambda_{orig}(e)
$$

$$
\iff w_{\theta}[e] = \frac{1}{\lambda_{orig}(e)} \sum_{\pi \in \Pi(x)} p_{\theta'} [\pi | x]|\pi|_e
$$

$$
\iff w_{\theta}[e] = \frac{1}{\lambda_{orig}(e)p_{\theta'}[x]} \sum_{\pi \in \Pi(x)} p_{\theta'} [x, \pi]|\pi|_e
$$

$$
\iff w_{\theta}[e] \propto \sum_{\pi \in \Pi(x)} p_{\theta'} [x, \pi]|\pi|_e.
$$
P3: Parameter Estimation

- But, \[ \sum_{\pi \in \Pi(x)} p_{\theta'}[x, \pi]|\pi|_e = \sum_{i=1}^{k} \alpha_{i-1}(\theta') w_{\theta'}[e] \beta_i(\theta') . \]

- Thus, with \[ w_{\theta}[e] \propto \sum_{i=1}^{k} \alpha_{i-1}(\theta') w_{\theta'}[e] \beta_i(\theta') \]

Can be computed using forward-backward algorithms, or, for us, shortest-distance algorithms.
References


References