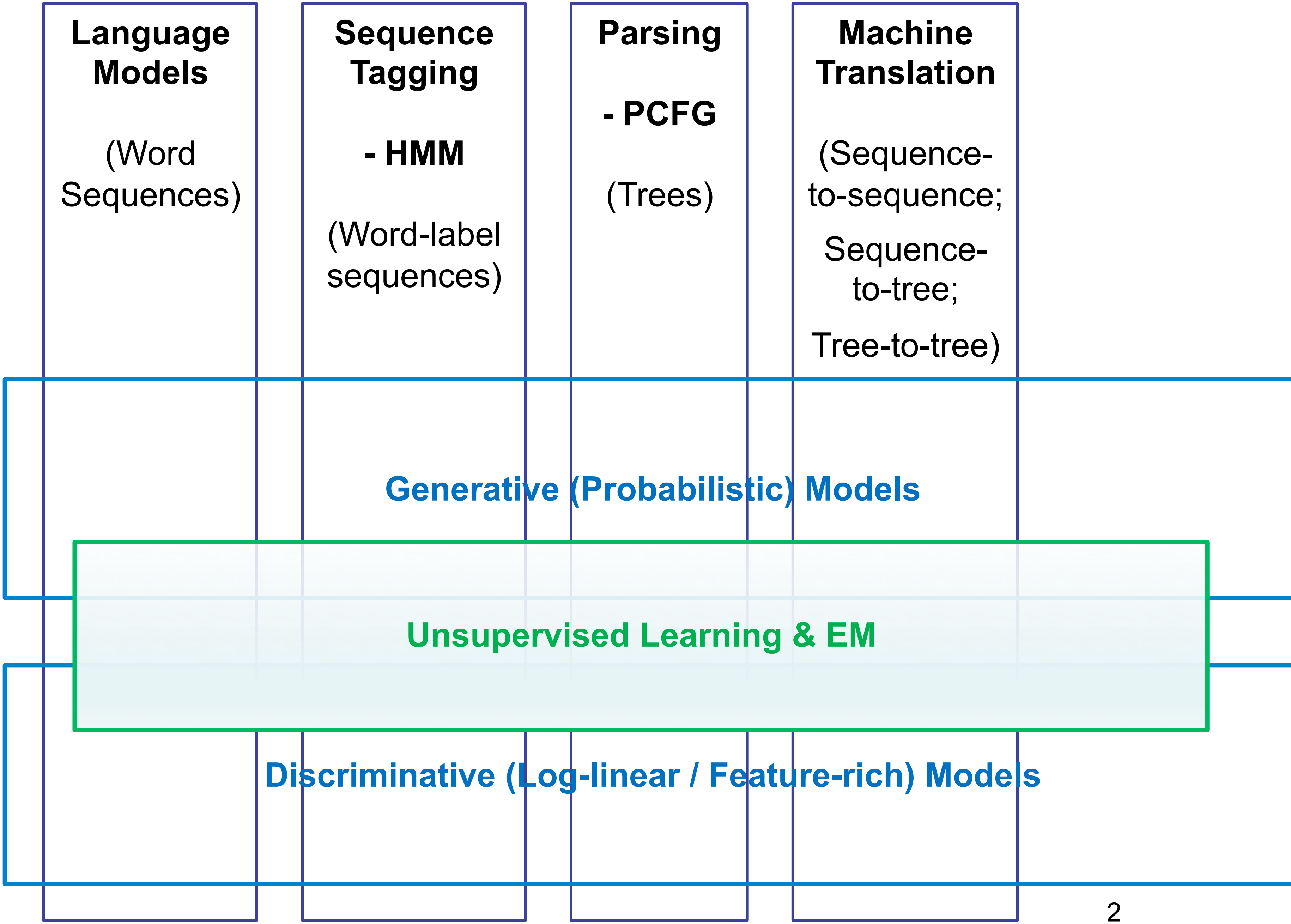
CSE 517
Natural Language Processing
Winter 2015

Expectation Maximization

Yejin Choi

University of Washington


Language Models
(Word Sequences)
Sequence Tagging
- HMM
(Word-label sequences)
Parsing
- PCFG
(Trees)
Machine Translation
(Sequence-to-sequence;
Sequence-to-tree;
Tree-to-tree)
Generative (Probabilistic) Models
Unsupervised Learning & EM
Discriminative (Log-linear / Feature-rich) Models

(Recap) Expectation Maximization for HMM

- Initialize transition and emission parameters
  - Random, uniform, or more informed initialization
- Iterate until convergence
  - **E-Step:**
    - Compute expected counts

$$\text{(expected) count(NN)} = \sum_i p(y_i = \text{NN} | x_1 \ldots x_n)$$

$$\text{(expected) count(NN} \rightarrow \text{VB)} = \sum_i p(y_i = \text{NN}, y_{i+1} = \text{VB} | x_1 \ldots x_n)$$

$$\text{(expected) count(NN} \rightarrow \text{apple)} = \sum_i p(y_i = \text{NN}, x_i = \text{apple} | x_1 \ldots x_n)$$

  - **M-Step:**
    - Compute new transition and emission parameters (using the expected counts computed above)

$$q_{ML}(y_i | y_{i-1}) = \frac{c(y_{i-1}, y_i)}{c(y_{i-1})} \quad e_{ML}(x|y) = \frac{c(y,x)}{c(y)}$$

- How does this relate to the general form of EM?

Expectation Maximization (General Form)

Input: model $p(x, y|\theta)$ and unlabeled data $D = \{x^1, x^2, ...x^N\}$

Initialize parameters θ

Until convergence

- **E-step** (expectation)
  - compute the posteriors (while fixing the model parameters)

$$p(y|x, \theta_t) = \frac{p(x, y|\theta^t)}{\sum_{y'} p(x, y'|\theta^t)}$$

- **M-step** (maximization)
  - compute parameters that maximize the *expected* log likelihood

$$\theta^{t+1} \leftarrow \operatorname*{argmax}_{\theta} \sum_i \sum_y \underbrace{p(y|x^i \ \theta^t)}_{\text{computed from E-step}} \log p(x^i, y|\theta)$$

Result: learn θ that maximizes:

$$L(\theta) = \sum_i \log p(x^i|\theta) = \sum_i \log \sum_y p(x^i, y|\theta)$$

Expectation Maximization

- **E-step** (expectation) $$p(y|x,\theta_t) = \frac{p(x,y|\theta^t)}{\sum_{y'} p(x,y'|\theta^t)}$$
  - compute the posteriors (while fixing the model parameters)
  - we don't actually need to compute the full posteriors, instead, we only need to compute "**sufficient statistics**" that matter for M-step, which boil down to "**expected counts**" of things
  - computationally expensive when y is structured multivariate
- **M-step** (maximization) $$\theta^{t+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_i \sum_y p(y|x^i \ \theta^t) \log p(x^i, y|\theta)$$
  - compute parameters that maximizes the expected log likelihood
  - For models that are a **product of multinomials** (e.g., naiveBayes, HMM, PCFG), closed forms exist ➔ "**maximum likelihood estimates (MLE)**"

Some Questions about EM

1. EM always converges?
2. EM converges even with approx E-step?

   -- hard EM / soft EM
3. EM converges to a global or local optimum? (or saddle point?)
4. EM improve "likelihood". How?

$$L(\theta) = \sum_i \log p(x^i|\theta) = \sum_i \log \sum_y p(x^i, y|\theta)$$

   -- while what M-step maximizes is "*expected* likelihood"
5. Maximum Likelihood Estimates (MLEs) for M-step?
6. When to use EM (or not)?

EM improves $L(\theta)$

- Theorem:

$$\textit{For any } \underline{\theta}, \underline{\theta}^{t-1} \in \Omega,\ L(\underline{\theta}) - L(\underline{\theta}^{t-1}) \geq Q(\underline{\theta}, \underline{\theta}^{t-1}) - Q(\underline{\theta}^{t-1}, \underline{\theta}^{t-1})$$

$$L(\underline{\theta}) = \sum_{i=1}^{n} \log p(x^{(i)}; \underline{\theta}) = \sum_{i=1}^{n} \log \sum_{y \in \mathcal{Y}} p(x^{(i)}, y; \underline{\theta})$$

$$Q(\underline{\theta}, \underline{\theta}^{t-1}) = \sum_{i=1}^{n} \sum_{y \in \mathcal{Y}} p(y|x^{(i)}; \underline{\theta}^{t-1}) \log p(x^{(i)}, y; \underline{\theta})$$

➔ Improvement on expected log likelihood is lower bound for improvement on log likelihood

- Concavity of Log (Jensen's inequality):

$$\log\left(\sum_i \alpha_i x_i\right) \geq \sum_i \alpha_i \log x_i$$

For any $\underline{\theta}, \underline{\theta}^{t-1} \in \Omega$, $L(\underline{\theta}) - L(\underline{\theta}^{t-1}) \geq Q(\underline{\theta}, \underline{\theta}^{t-1}) - Q(\underline{\theta}^{t-1}, \underline{\theta}^{t-1})$

$$L(\underline{\theta}) - L(\underline{\theta}^{t-1}) = \sum_{i=1}^{n} \log \frac{\sum_y p(x^{(i)}, y; \underline{\theta})}{\sum_y p(x^{(i)}, y; \underline{\theta}^{t-1})}$$

$$= \sum_{i=1}^{n} \log \sum_y \left(\frac{p(x^{(i)}, y; \underline{\theta})}{p(x^{(i)}; \underline{\theta}^{t-1})} \right)$$

$$= \sum_{i=1}^{n} \sum_y p(y|x^{(i)}; \underline{\theta}^{t-1}) \log p(x^{(i)}, y; \underline{\theta}) - \sum_{i=1}^{n} \sum_y p(y|x^{(i)}; \underline{\theta}^{t-1}) \log p(x^{(i)}, y; \underline{\theta}^{t-1})$$

$$= Q(\underline{\theta}, \underline{\theta}^{t-1}) - Q(\underline{\theta}^{t-1}, \underline{\theta}^{t-1})$$

Convergence of EM

- Theorem:

$$\textit{For any } \underline{\theta}, \underline{\theta}^{t-1} \in \Omega,\ L(\underline{\theta}) - L(\underline{\theta}^{t-1}) \geq Q(\underline{\theta}, \underline{\theta}^{t-1}) - Q(\underline{\theta}^{t-1}, \underline{\theta}^{t-1})$$

- Above only tells us that EM is “non-decreasing” $L(\theta)$
- Under relatively mild conditions, it can be shown that EM converges to a local optimum of $L(\theta)$
- *“On the Convergence Properties of the EM Algorithm” Wu, 1983*

➔ As long as M-step improves expected log likelihood (at all), EM improves log likelihood. (Even if we don’t find argmax in M-step!)

Maximum Likelihood Estimates

Supervised Learning for

1. Language Models:

$$q_{ML}(w) = \frac{c(w)}{c()}, \quad q_{ML}(w|v) = \frac{c(v,w)}{c(v)}, \quad q_{ML}(w|u,v) = \frac{c(u,v,w)}{c(u,v)},$$

2. HMM:

$$q_{ML}(y_i|y_{i-1}) = \frac{c(y_{i-1}, y_i)}{c(y_{i-1})} \qquad e_{ML}(x|y) = \frac{c(y,x)}{c(y)}$$

3. PCFG:

$$q_{ML}(\alpha \rightarrow \beta) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)}$$

Maximum Likelihood Estimates

Models:

1. Language Models: $$p(x_1 \ldots x_n) = \prod_{i=1}^{n} p(x_i | x_{i-1})$$

2. HMM:

$$p(x_1 \ldots x_n, y_1 \ldots y_n) = q(STOP | y_n) \prod_{i=1}^{n} q(y_i | y_{i-1}) e(x_i | y_i)$$

3. PCFG: $$p(t) = \prod_{i=1}^{n} q(\alpha_i \rightarrow \beta_i)$$

What's common?

➔ product of multinomials*!

*multinomials is a conflated term. "categorical distribution" is more correct

MLEs maximize Likelihood

Supervised Learning for

1. Language Models: $q_{ML}(w) = \frac{c(w)}{c()}, \quad q_{ML}(w|v) = \frac{c(v,w)}{c(v)}, \quad q_{ML}(w|u,v) = \frac{c(u,v,w)}{c(u,v)},$

2. HMM: $q_{ML}(y_i|y_{i-1}) = \frac{c(y_{i-1}, y_i)}{c(y_{i-1})} \qquad e_{ML}(x|y) = \frac{c(y,x)}{c(y)}$

3. PCFG: $q_{ML}(\alpha \rightarrow \beta) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)}$

➔ Happens to be intuitive, we can also prove that

- MLE with actual counts maximize **log likelihood**

$$L(\theta) = \sum_i \log\, p(x^i|\theta) = \sum_i \log\, \sum_y p(x^i, y|\theta)$$

- MLE with *expected* counts maximize ***expected* log likelihood**

$$E_{p(y|x)}[l(\theta)] = \sum_i \sum_y p(y|x^i\ \theta^t)\log\, p(x^i, y|\theta)$$

MLE for multinomial distributions

- Let's first consider a simpler case.
- We want to learn parameters that maximize the (log) likelihood of the training data:

$$l(\theta) = \sum_i \log p(x^i) = \boxed{\sum_k c_k \log \theta_k}$$

- Since it's multinomial, it must be that $\sum_k \theta_k = 1$

- c_k := count of θ_k used in the likelihood of training data
- For example, for Unigram LM, $p(x^i = \text{apple}) = \theta_{\text{apple}}$ and c_k := count (apple) in the training corpus

MLE for multinomial distributions

- Learning parameters for

$$\theta = \underset{\theta}{\mathrm{argmax}} \sum_k c_k \log\theta_k \quad \text{such that} \quad \sum_k \theta_k = 1$$

- equivalent to learning parameters for

$$\underset{\theta}{\mathrm{argmax}} \sum_k c_k \log\theta_k - \min_\lambda \lambda\Big(\sum_k \theta_k - 1\Big)$$

- lambda is called **Lagrangian multiplier**

$$g(\lambda, \theta) := \sum_k c_k \log\theta_k - \underbrace{\lambda\Big(\sum_k \theta_k - 1\Big)}_{\text{encode constraint}}$$

- You can add additional lambda terms: one for each equality constraint

$$g(\lambda, \theta) := \sum_k c_k \log\theta_k - \lambda_1(f_1(\theta) - C_1) - \lambda_2(f_2(\theta) - C_2) - \ldots$$

MLE for multinomial distributions

- Learning parameters for

$$\theta = \underset{\theta}{\operatorname{argmax}} \sum_k c_k \log \theta_k \quad \text{such that} \quad \sum_k \theta_k = 1$$

- equivalent to learning parameters for

$$\min_{\lambda} \max_{\theta} \left[g(\lambda, \theta) := \sum_k c_k \log \theta_k - \lambda \left(\sum_k \theta_k - 1 \right) \right]$$

- Find optimal parameters by setting partial derivatives = 0

$$\theta_k = \frac{c_k}{\lambda} \text{ and } \lambda = \sum_k c_k$$

- We have MLE! -- can be generalized to a product of multinomials, e.g., HMM, PCFG. For each prob distribution that needs to sum to 1, create a different lambda term.
- *"Lagrange Multipliers without Permanent Scarring"*, Dan Klein (http://www.cs.berkeley.edu/~klein/papers/lagrange-multipliers.pdf)

When to use EM (or not)

- The ultimate goal of (unsupervised) learning is to find the parameters θ that maximizes the likelihood over the training data:

$$L(\theta) = \sum_i \log p(x^i|\theta) = \sum_i \log \sum_y p(x^i, y|\theta)$$

- For some models, it is difficult to find the parameters that maximize the log likelihood directly.
- For such models, it is sometimes very easy to find the parameters that maximizes the ***expected*** log likelihood. (Use EM!)

$$E_{p(y|x)}[l(\theta)] = \sum_i \sum_y p(y|x^i\ \theta^t)\log p(x^i, y|\theta)$$

- For example, there are closed form solutions (MLE) for models that are in the form of product of multinomials (i.e., categorical distributions).
- If optimizing for expected log likelihood is not any easier than optimizing for log likelihood --- no need to use EM.

Other EM Variants

- Generalized EM (GEM)
  - When exact M-step is difficult: finds θ that improves, but not necessarily maximizes. Converges to a local optimum.
- Stochastic EM
  - When exact E-step is difficult: Monte Carlo sampling. Will asymptotically converge to a local optimum
- Hard EM
  - When exact E-step is difficult: find the best prediction of the hidden variable 'y' and put all the prob mass (= 1) to that best prediction.
  - K-means is Hard EM.
  - Will converge if improving the expected log likelihood of M-step.