CS 533: Natural Language Processing

Latent-Variable Generative Models and the Expectation Maximization (EM) Algorithm

Karl Stratos

Rutgers University

Motivation: Bag-Of-Words (BOW) Document Model

- Fixed-length documents $x \in V^T$
- BOW parameters: word distribution p_W over V defining

$$p_X(x) = \prod_{t=1}^{T} p_W(x_t)$$

- Model's generative story: any word in any document is independently generated.
- What if the true generative story underlying data is different?

$$\begin{aligned} V &= \{a, b\} & \qquad x^{(1)} &= (a, a, a, a, a, a, a, a, a, a) \\ T &= 10 & \qquad x^{(2)} &= (b, b, b, b, b, b, b, b, b, b) \end{aligned}$$

- MLE: $p_X(x^{(1)}) = p_X(x^{(2)}) = (1/2)^{10}$

Latent-Variable BOW (LV-BOW) Document Model

- LV-BOW parameters
  - p_Z: "topic" distribution over $\{1 \ldots K\}$
  - $p_{W|Z}$: conditional word distribution over V

  defining

$$p_{X|Z}(x|z) = \prod_{t=1}^{T} p_{W|Z}(x_t|z) \qquad\qquad \forall z \in \{1 \ldots K\}$$

$$p_X(x) = \sum_{z=1}^{K} p_Z(z) \times p_{X|Z}(x|z)$$

- Model's generative story: for each document, a topic is generated and conditioning on that words are independently generated

Back to the Example

$$V = \{a, b\} \qquad x^{(1)} = (a, a, a, a, a, a, a, a, a, a)$$

$$T = 10 \qquad x^{(2)} = (b, b, b, b, b, b, b, b, b, b)$$

- $K = 2$ with $p_Z(1) = p_Z(2) = 1/2$
- $p_{W|Z}(a|1) = p_{W|Z}(b|2) = 1$
- $p_X(x^{(1)}) = p_X(x^{(2)}) = 1/2 \gg (1/2)^{10}$

Key idea: introduce a **latent variable** Z to model true generative process more faithfully

The Latent-Variable Generative Model Paradigm

Model. Joint distribution over X and Z

$$p_{XZ}(x, z) = p_Z(z) \times p_{X|Z}(x|z)$$

Learning. We don't observe Z!

$$\max_{p_{XZ}} \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X} \left[\log \underbrace{\sum_{z \in \mathcal{Z}} p_{XZ}(x, z)}_{p_X(x)} \right]$$

The Learning Problem

- How can we solve

$$\max_{p_{XZ}} \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X} \left[\log \sum_{z \in \mathcal{Z}} p_{XZ}(x, z) \right]$$

- Specifically for LV-BOW, given N documents $x^{(1)} \ldots x^{(N)} \in V^T$, how can we learn topic distribution p_Z and conditional word distribution $p_{W|Z}$ that maximize

$$\sum_{i=1}^{N} \log \left(\sum_{z \in \mathcal{Z}} p_Z(z) \times \prod_{t=1}^{T} p_{W|Z}(x_t^{(i)} | z) \right)$$

A Proposed Algorithm

1. Initialize p_Z and $p_{W|Z}$ as random distributions.
2. Repeat until convergence:
   2.1 For $i = 1 \ldots N$ compute conditional posterior distribution

$$p_{Z|X}(z|x^{(i)}) = \frac{p_Z(z) \times \prod_{t=1}^{T} p_{W|Z}(x_t^{(i)}|z)}{\sum_{z'=1}^{K} p_Z(z') \times \prod_{t=1}^{T} p_{W|Z}(x_t^{(i)}|z')}$$

   2.2 Update model parameters by

$$p_Z(z) = \frac{\sum_{i=1}^{N} p_{Z|X}(z|x^{(i)})}{\sum_{z'=1}^{K} \sum_{i=1}^{N} p_{Z|X}(z'|x^{(i)})}$$

$$p_{W|Z}(w|z) = \frac{\sum_{i=1}^{N} p_{Z|X}(z|x^{(i)}) \times \textbf{count}(w|x^{(i)})}{\sum_{w' \in V} \sum_{i=1}^{N} p_{Z|X}(z|x^{(i)}) \times \textbf{count}(w'|x^{(i)})}$$

   where $\textbf{count}(w|x^{(i)})$ is number of times $w \in V$ appears in $x^{(i)}$.

Code

```
def compute_posterior(data, pZ, pW_Z):
    pZ_cond = {}
    for i in range(len(data)):
        pZ_cond[i] = {}
        normalizer = 0
        for z in Z:
            pZ_cond[i][z] = pZ[z] * np.prod([pW_Z[z][w] for w in data[i]])
            normalizer += pZ_cond[i][z]

        for z in Z:
            pZ_cond[i][z] /= normalizer

    return pZ_cond
```

```
for inum in range(M):
    ll = compute_log_likelihood(data, pZ, pW_Z)
    print_stuff(inum, pZ, pW_Z, ll)

    pZ_cond = compute_posterior(data, pZ, pW_Z)

    expected_count_Z = {}
    total_expected_count_Z = 0
    for z in Z:
        expected_count_Z[z] = sum([pZ_cond[i][z] for i in range(len(data))])
        total_expected_count_Z += expected_count_Z[z]

    for z in Z:
        pZ[z] = expected_count_Z[z] / total_expected_count_Z

    expected_count_ZW = {}
    for z in Z:
        for w in V:
            expected_count_ZW[(z, w)] = sum([pZ_cond[i][z] * data[i].count(w)
                                             for i in range(len(data))])

    expected_count_Z = {}
    for z in Z:
        expected_count_Z[z] = sum([expected_count_ZW[(z, w)] for w in V])

    for z in Z:
        for w in V:
            pW_Z[z][w] = expected_count_ZW[(z, w)] / expected_count_Z[z]
```

Code in Action

```
Data
a a a a a a a a a a
b b b b b b b b b b

Iteration 1
pZ(1)=0.38  pZ(2)=0.62
pW_Z(a|1)=0.76  pW_Z(b|1)=0.24  pW_Z(a|2)=0.31  pW_Z(b|2)=0.69
Log likelihood: -7.96229

Iteration 2
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=1.00  pW_Z(b|1)=0.00  pW_Z(a|2)=0.00  pW_Z(b|2)=1.00
Log likelihood: -1.38887

Iteration 3
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=1.00  pW_Z(b|1)=0.00  pW_Z(a|2)=0.00  pW_Z(b|2)=1.00
Log likelihood: -1.38629

Iteration 4
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=1.00  pW_Z(b|1)=0.00  pW_Z(a|2)=0.00  pW_Z(b|2)=1.00
Log likelihood: -1.38629

pX(a a a a a a a a a a)=0.50
pX(b b b b b b b b b b)=0.50
```

Code in Action: Bad Initialization

```
Data
a a a a a a a a a a
b b b b b b b b b b

Iteration 1
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.50  pW_Z(b|1)=0.50  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -13.86294

Iteration 2
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.50  pW_Z(b|1)=0.50  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -13.86294

Iteration 3
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.50  pW_Z(b|1)=0.50  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -13.86294

Iteration 4
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.50  pW_Z(b|1)=0.50  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -13.86294

pX(a a a a a a a a a a)=0.000977
pX(b b b b b b b b b b)=0.000977
```

Another Example

Initial values

```
Data
a a a a a a a a a a
b b b b b a a a a a
a a a a a b b b b b

Iteration 1
pZ(1)=0.38  pZ(2)=0.62
pW_Z(a|1)=0.76  pW_Z(b|1)=0.24  pW_Z(a|2)=0.31  pW_Z(b|2)=0.69
Log likelihood: -19.53394

Iteration 2
pZ(1)=0.47  pZ(2)=0.53
pW_Z(a|1)=0.85  pW_Z(b|1)=0.15  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -17.41683

Iteration 3
pZ(1)=0.35  pZ(2)=0.65
pW_Z(a|1)=0.97  pW_Z(b|1)=0.03  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -16.03630

Iteration 4
pZ(1)=0.33  pZ(2)=0.67
pW_Z(a|1)=1.00  pW_Z(b|1)=0.00  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -15.77058

Iteration 5
pZ(1)=0.33  pZ(2)=0.67
pW_Z(a|1)=1.00  pW_Z(b|1)=0.00  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -15.77052
```

After convergence

```
Iteration 100
pZ(1)=0.33  pZ(2)=0.67
pW_Z(a|1)=1.00  pW_Z(b|1)=0.00  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -15.77052

pX(a a a a a a a a a a)=0.333333
pX(b b b b b a a a a a)=0.000652
pX(a a a a a b b b b b)=0.000652
```

Again Possible to Get Stuck in a Local Optimum

Initial values

```
Data
a a a a a a a a a a
b b b b b a a a a a
a a a a a b b b b b

Iteration 1
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.50  pW_Z(b|1)=0.50  pW_Z(a|2)=0.50  pW_Z(b|2)=0.50
Log likelihood: -20.79442

Iteration 2
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.67  pW_Z(b|1)=0.33  pW_Z(a|2)=0.67  pW_Z(b|2)=0.33
Log likelihood: -19.09543

Iteration 3
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.67  pW_Z(b|1)=0.33  pW_Z(a|2)=0.67  pW_Z(b|2)=0.33
Log likelihood: -19.09543

Iteration 4
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.67  pW_Z(b|1)=0.33  pW_Z(a|2)=0.67  pW_Z(b|2)=0.33
Log likelihood: -19.09543

Iteration 5
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.67  pW_Z(b|1)=0.33  pW_Z(a|2)=0.67  pW_Z(b|2)=0.33
Log likelihood: -19.09543
```

After convergence

```
Iteration 100
pZ(1)=0.50  pZ(2)=0.50
pW_Z(a|1)=0.67  pW_Z(b|1)=0.33  pW_Z(a|2)=0.67  pW_Z(b|2)=0.33
Log likelihood: -19.09543

pX(a a a a a a a a a a)=0.017342
pX(b b b b b a a a a a)=0.000542
pX(a a a a a b b b b b)=0.000542
```

Why Does It Work?

- A special case of the **expectation maximization (EM) algorithm** adapted for LV-BOW
- EM is an extremely important and general concept
  - Another special case: variational autoencoders (VAEs, next class)

Setting

- Original problem: difficult to optimize (nonconvex)

$$\max_{p_{XZ}} \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X} \left[\log \sum_{z \in \mathcal{Z}} p_{XZ}(x, z) \right]$$

- Alternative problem: easy to optimize (often concave)

$$\max_{p_{XZ}} \mathop{\mathbf{E}}_{\substack{x \sim \mathbf{pop}_X \\ z \sim q_{Z|X}(\cdot|x)}} \left[\log p_{XZ}(x, z) \right]$$

where $q_{Z|X}$ is some arbitrary posterior distribution that is easy to compute

Solving the Alternative Problem

- Many models we considered (LV-BOW, HMM, PCFG) can be written as

$$p_{XZ}(x,z) = \prod_{(\tau,a)\in\mathcal{E}} p_\tau(a)^{\textbf{count}^\tau(a|x,z)}$$

  - $\mathcal{E}$ is a set of possible event type-value pairs.
  - $\textbf{count}^\tau(a|x,z)$ is number of times $\tau = a$ happens in (x,z)
  - Model has a distribution p_τ over possible values of type τ

- Example

$$p_{XZ}((a,a,a,b,b),2) = p_Z(2) \times p_{W|Z}(a|2)^3 \times p_{W|Z}(b|2)^2 \quad \text{(LV-BOW)}$$

$$p_{XZ}((\texttt{La},\texttt{La},\texttt{La}),(N,N,N)) = o(\texttt{La}|N)^3 \times t(N|*) \times t(N|N)^2 \times t(\texttt{STOP}|N) \quad \text{(HMM)}$$

Closed-Form Solution

If $x^{(1)} \ldots x^{(N)} \sim \mathbf{pop}_X$ are iid samples,

$$\begin{aligned}
&\max_{p_{XZ}} \underset{\substack{x \sim \mathbf{pop}_X \\ z \sim q_{Z|X}(\cdot|x)}}{\mathbf{E}} [\log p_{XZ}(x, z)] \\
&\approx \max_{p_{XZ}} \sum_{i=1}^{N} \sum_{z} q_{Z|X}(z|x^{(i)}) \log p_{XZ}(x^{(i)}, z) \\
&= \max_{p_\tau} \sum_{i=1}^{N} \sum_{z} q_{Z|X}(z|x^{(i)}) \sum_{(\tau,a) \in \mathcal{E}} \textbf{count}^\tau(a|x^{(i)}, z) \log p_\tau(a) \\
&= \max_{p_\tau} \sum_{(\tau,a) \in \mathcal{E}} \left(\sum_{i=1}^{N} \sum_{z} q_{Z|X}(z|x^{(i)}) \textbf{count}^\tau(a|x^{(i)}, z) \right) \log p_\tau(a)
\end{aligned}$$

MLE solution!

$$p_\tau(a) = \frac{\sum_{i=1}^{N} \sum_{z} q_{Z|X}(z|x^{(i)}) \textbf{count}^\tau(a|x^{(i)}, z)}{\sum_{a'} \sum_{i=1}^{N} \sum_{z} q_{Z|X}(z|x^{(i)}) \textbf{count}^\tau(a'|x^{(i)}, z)}$$

This is How We Derived LV-BOW EM Updates

Using $q_{Z|X} = p_{Z|X}$

$$p_Z(z) = \frac{\sum_{i=1}^N \sum_{z'} p_{Z|X}(z'|x^{(i)})\textbf{count}^\tau(z' = z|x^{(i)}, z')}{\sum_{z''} \sum_{i=1}^N \sum_{z'} p_{Z|X}(z'|x^{(i)})\textbf{count}^\tau(z' = z''|x^{(i)}, z')}$$
$$= \frac{\sum_{i=1}^N p_{Z|X}(z|x^{(i)})}{\sum_{z''} \sum_{i=1}^N p_{Z|X}(z''|x^{(i)})}$$

$$p_{W|Z}(w|z) = \frac{\sum_{i=1}^N \sum_{z'} p_{Z|X}(z'|x^{(i)})\textbf{count}^\tau(z' = z, w|x^{(i)}, z')}{\sum_{w' \in V} \sum_{i=1}^N \sum_{z'} p_{Z|X}(z'|x^{(i)})\textbf{count}^\tau(z' = z, w'|x^{(i)}, z')}$$
$$= \frac{\sum_{i=1}^N p_{Z|X}(z|x^{(i)})\textbf{count}(w|x^{(i)})}{\sum_{w' \in V} \sum_{i=1}^N p_{Z|X}(z|x^{(i)})\textbf{count}(w'|x^{(i)})}$$

Game Plan

- So we have established that it is often easy to solve the alternative problem

$$\max_{p_{XZ}} \mathop{\mathbf{E}}_{\substack{x \sim \mathbf{pop}_X \\ z \sim q_{Z|X}(\cdot|x)}} [\log p_{XZ}(x, z)]$$

  where $q_{Z|X}$ is any posterior distribution easy to compute

- We will relate the original log likelihood objective with this quantity by the following slide.

ELBO: Evidence Lower Bound

For any $q_{Z|X}$ we define

$$\mathrm{ELBO}(p_{XZ}, q_{Z|X}) = \mathop{\mathbf{E}}_{\substack{x \sim \mathbf{pop}_X \\ z \sim q_{Z|X}(\cdot|x)}} [\log p_{XZ}(x, z)] + H(q_{Z|X})$$

where $H(q_{Z|X}) = \mathop{\mathbf{E}}_{\substack{x \sim \mathbf{pop}_X \\ z \sim q_{Z|X}(\cdot|x)}} \left[-\log q_{Z|X}(z|x) \right]$.

Claim. For all $q_{Z|X}$,

$$\mathrm{ELBO}(p_{XZ}, q_{Z|X}) \leq \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X} \left[\log \sum_{z \in \mathcal{Z}} p_{XZ}(x, z) \right]$$

with equality iff $q_{Z|X} = p_{Z|X}$. (Proof on board)

EM: Coordinate Ascent on ELBO

Input: sampling access to $\mathbf{pop}_X$, definition of p_{XZ}
Output: local optimum of

$$\max_{p_{XZ}} \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X} \left[\log \sum_{z \in \mathcal{Z}} p_{XZ}(x, z) \right]$$

1. Initialize p_{XZ} (e.g., random distribution).
2. Repeat until convergence:

$$q_{Z|X} \leftarrow \arg\max_{\bar{q}_{Z|X}} \; \mathrm{ELBO}(p_{XZ}, \bar{q}_{Z|X})$$

$$p_{XZ} \leftarrow \arg\max_{\bar{p}_{XZ}} \; \mathrm{ELBO}(\bar{p}_{XZ}, q_{Z|X})$$

3. Return p_{XZ}

Equivalently

Input: sampling access to $\mathbf{pop}_X$, definition of p_{XZ}
Output: local optimum of

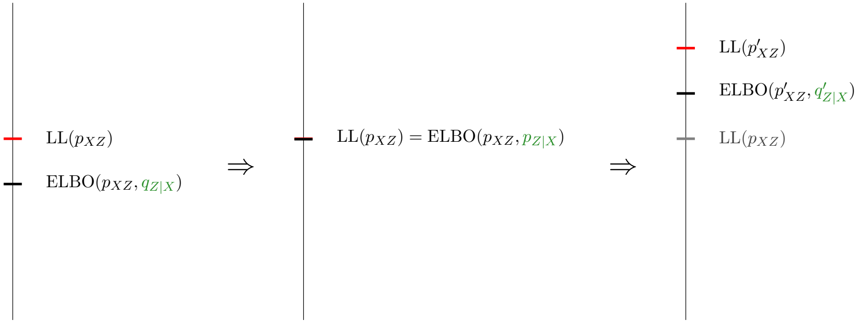
$$\max_{p_{XZ}} \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X} \left[\log \sum_{z \in \mathcal{Z}} p_{XZ}(x, z) \right]$$

1. Initialize p_{XZ} (e.g., random distribution).
2. Repeat until convergence:

$$p_{XZ} \leftarrow \arg\max_{p_{XZ}} \mathop{\mathbf{E}}_{\substack{x \sim \mathbf{pop}_X \\ z \sim p_{Z|X}(\cdot|x)}} [\log p_{XZ}(x, z)]$$

3. Return p_{XZ}

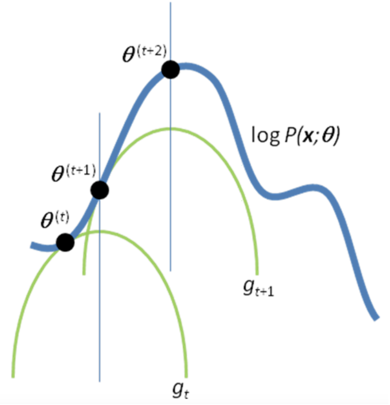
EM Can Only Increase the Objective (Or Leave It Unchanged)



$$\mathrm{LL}(p_{XZ}) = \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X}\left[\log \sum_{z \in \mathcal{Z}} p_{XZ}(x, z)\right]$$

$$\mathrm{ELBO}(p_{XZ}, q_{Z|X}) = \mathrm{LL}(p_{XZ}) - D_{\mathrm{KL}}(q_{Z|X} || p_{Z|X}) = \mathop{\mathbf{E}}_{\substack{x \sim \mathbf{pop}_X \\ z \sim q_{Z|X}(\cdot|x)}}[\log p_{XZ}(x, z)] + H(q_{Z|X})$$

EM Can Only Increase the Objective (Or Leave It Unchanged)



From https://media.nature.com/full/nature-assets/nbt/journal/v26/n8/extref/nbt1406-S1.pdf

Sample Version

Input: N iid samples from $\mathbf{pop}_X$, definition of p_{XZ}
Output: local optimum of

$$\max_{p_{XZ}} \frac{1}{N} \sum_{i=1}^{N} \log \sum_{z \in \mathcal{Z}} p_{XZ}(x^{(i)}, z)$$

1. Initialize p_{XZ} (e.g., random distribution).
2. Repeat until convergence:

$$p_{XZ} \leftarrow \arg\max_{\bar{p}_{XZ}} \sum_{i=1}^{N} \sum_{z \in \mathcal{Z}} p_{Z|X}(z|x^{(i)}) \log \bar{p}_{XZ}(x^{(i)}, z)$$

3. Return p_{XZ}

EM for HMM (Baum-Welch)

Input: sequences $x^{(1)} \ldots x^{(N)} \in V^T$

1. Initialize emission $o(w|y)$ and transition $t(y'|y)$ probabilities.
2. Repeat until convergence:

$$o, t \leftarrow \arg\max_{\bar{o},\bar{t}} \sum_{i=1}^{N} \sum_{z \in \mathcal{Y}^T} p_{Z|X}(z|x^{(i)}) \log p_{XZ}^{\bar{o},\bar{t}}(x^{(i)}, z)$$

where

$$p_{XZ}^{o,t}(x, z) = \prod_{y,w} o(w|y)^{\textbf{count}((y,w)|x,z)} \times \prod_{y,y'} t(y'|y)^{\textbf{count}((y,y')|x,z)}$$

Baum-Welch Updates: Emission Probabilities

$$o(w|y) = \frac{\sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)}) \textbf{count}((y,w)|x^{(i)}, z)}{\sum_{w' \in V} \sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)}) \textbf{count}((y,w')|x^{(i)}, z)}$$

$$= \frac{\sum_{i=1}^{N} \sum_{t=1}^{T} \mu(y|x^{(i)}, t) \left[\left[x_t^{(i)} = w\right]\right]}{\sum_{w' \in V} \sum_{i=1}^{N} \sum_{t=1}^{T} \mu(y|x^{(i)}, t) \left[\left[x_t^{(i)} = w'\right]\right]}$$

where $\mu(y|x^{(i)}, t)$ is the *conditional* probability that t-th label is equal to y in $x^{(i)}$ which can be calculated from the forward/backward probabilities:

$$\mu(y|x^{(i)}, t) = \frac{\alpha(t,y) \times \beta(t,y)}{p_X(x^{(i)})}$$

Baum-Welch Updates: Transition Probabilities

$$t(y'|y) = \frac{\sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)}) \textbf{count}((y,y')|x^{(i)}, z)}{\sum_{y' \in \mathcal{Y}} \sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)}) \textbf{count}((y,y')|x^{(i)}, z)}$$

$$= \frac{\sum_{i=1}^{N} \sum_{t=1}^{T} \mu(y, y'|x^{(i)}, t)}{\sum_{w' \in V} \sum_{i=1}^{N} \sum_{t=1}^{T} \mu(y, y'|x^{(i)}, t)}$$

where $\mu(y, y'|x^{(i)}, t)$ is the *conditional* probability that t-th label pair is equal to (y, y') in $x^{(i)}$ which can be calculated from the forward/backward probabilities:

$$\mu(y, y'|x^{(i)}, t) = \frac{\alpha(t, y) \times t(y'|y) \times o(x_t|y') \times \beta(t+1, y')}{p_X(x^{(i)})}$$

Summary of Baum-Welch

- Given N unlabeled sequences, find a local optimum of

$$\arg\max_{o,t} \frac{1}{N} \sum_{i=1}^{N} \log \sum_{z \in \mathcal{Y}^T} p_{XZ}^{o,t}(x^{(i)}, z)$$

  where o and t are emission/transition probabilities of HMM

- Initialize o, t and repeat until convergence:
  - Run forward-backward algorithm on $x^{(1)} \ldots x^{(N)}$ using the current o, t values
  - Use the probabilities to compute marginals.
  - Use the marginals to compute "expected counts" of word-tag pairs (w, y) and tag pairs (y, y') across all data.
  - Get new o, t by the previous updates.

EM for PCFG

Input: sequences $x^{(1)} \dots x^{(N)} \in V^T$

1. Initialize rule probabilities $q(\alpha \to \beta)$.
2. Repeat until convergence:

$$q \leftarrow \arg\max_{\bar{q}} \sum_{i=1}^{N} \sum_{z \in \textbf{GEN}(x^{(i)})} p_{Z|X}(z|x^{(i)}) \log p_{XZ}^{\bar{q}}(x^{(i)}, z)$$

where

$$p_{XZ}^{q}(x, z) = \prod_{\alpha \to \beta} q(\alpha \to \beta)^{\textbf{count}(\alpha \to \beta | x, z)}$$

Unary Rule Probability Updates

$$q(a \to w) = \frac{\sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)})\textbf{count}(a \to w|x^{(i)}, z)}{\sum_{w'} \sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)})\textbf{count}(a \to w'|x^{(i)}, z)}$$

$$= \frac{\sum_{i=1}^{N} \sum_{t=1}^{T} \mu(a|x^{(i)}, t) \left[\left[x_t^{(i)} = w\right]\right]}{\sum_{w'} \sum_{i=1}^{N} \sum_{t=1}^{T} \mu(a|x^{(i)}, t) \left[\left[x_t^{(i)} = w'\right]\right]}$$

where $\mu(a|x^{(i)}, t)$ is the *conditional* probability that a spans $x_t^{(i)}$ which can be calculated from the inside/outside probabilities:

$$\mu(a|x^{(i)}, t) = \frac{\alpha(a, t, t) \times \beta(a, t, t)}{p_X(x^{(i)})}$$

Binary Rule Probability Updates

$$q(a \to b\ c) = \frac{\sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)})\textbf{count}(a \to b\ c|x^{(i)}, z)}{\sum_{(b',c')} \sum_{i=1}^{N} \sum_{z} p_{Z|X}(z|x^{(i)})\textbf{count}(a \to b'\ c'|x^{(i)}, z)}$$

$$= \frac{\sum_{i=1}^{N} \sum_{1 \leq t \leq k < s \leq T} \mu(a \to b\ c|x^{(i)}, t, k, s)}{\sum_{(b',c')} \sum_{i=1}^{N} \sum_{1 \leq t \leq k < s \leq T} \mu(a \to b\ c|x^{(i)}, t, k, s)}$$

where $\mu(a \to b\ c|x^{(i)}, t, k, s)$ is the *conditional* probability that rule $a \to b\ c$ spans $x_t^{(i)} \dots x_s^{(i)}$ with a split point k which can be calculated from the inside/outside probabilities:

$$\mu(a \to b\ c|x^{(i)}, t, k, s) = \frac{\beta(a, t, s) \times q(a \to b\ c) \times \alpha(b, t, k) \times \alpha(c, k+1, j)}{p_X(x^{(i)})}$$

Summary Points

- Latent-variable generative models

$$p_{XZ}(x, z) = p_Z(z) \times p_{X|Z}(x|z)$$

- Learning objective

$$\mathrm{LL}(p_{XZ}) = \mathop{\mathbf{E}}_{x \sim \mathbf{pop}_X} [\log \sum_{z \in \mathcal{Z}} p_{XZ}(x, z)]$$

- ELBO is a "variational" lower bound on the objective

$$\mathrm{ELBO}(p_{XZ}, q_{Z|X}) \leq \mathrm{LL}(p_{XZ}) \qquad\qquad \forall q_{Z|X}$$

tight when $q_{Z|X} = p_{Z|X}$

- EM is an alternating maximization of ELBO

$$q_{Z|X} \leftarrow \mathop{\arg\max}_{\bar{q}_{Z|X}} \ \mathrm{ELBO}(p_{XZ}, \bar{q}_{Z|X}) = p_{Z|X}$$

$$p_{XZ} \leftarrow \mathop{\arg\max}_{\bar{p}_{XZ}} \ \mathrm{ELBO}(\bar{p}_{XZ}, q_{Z|X}) = \mathop{\arg\max}_{p_{XZ}} \mathop{\mathbf{E}}_{\substack{x \sim \mathbf{pop}_X \\ z \sim q_{Z|X}(\cdot|x)}} [\log p_{XZ}(x, z)]$$