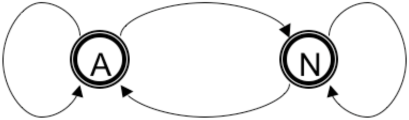
CMPT-413: Computational Linguistics
HMM6: Unsupervised learning of Hidden Markov Models

Anoop Sarkar
http://www.cs.sfu.ca/∼anoop

February 28, 2013

Hidden Markov Model

$$\text{Model } \theta = \begin{cases} \pi_i & \text{probability of starting at state } i \\ a_{i,j} & \text{probability of transition from state } i \text{ to state } j \\ b_i(o) & \text{probability of output } o \text{ at state } i \end{cases}$$



killer crazy clown problem

killer crazy clown problem

Hidden Markov Model Algorithms

- HMM as parser: compute the best sequence of states for a given observation sequence.
- HMM as language model: compute probability of given observation sequence.
- HMM as learner: given a corpus of observation sequences, learn its distribution, i.e. learn the parameters of the HMM from the corpus.
  - Learning from a set of observations with the sequence of states provided (states are not hidden) [Supervised Learning]
  - Learning from a set of observations without any state information. [Unsupervised Learning]

Learning from Unlabeled Data

- Unlabeled Data $U = x_1, \ldots, x_m$:

```
x1: killer clown
x2: killer problem
x3: crazy problem
x4: crazy clown
```

- `y1, y2, y3, y4` are unknown.
- But we can enumerate all possible values for `y1, y2, y3, y4`
- For example, for `x1: killer clown`

`x1,y1,1:`	`killer/A clown/A`	$p_1 = \pi_A \cdot b_A(killer) \cdot a_{A,A} \cdot b_A(clown)$
`x1,y1,2:`	`killer/A clown/N`	$p_2 = \pi_A \cdot b_A(killer) \cdot a_{A,N} \cdot b_N(clown)$
`x1,y1,3:`	`killer/N clown/N`	$p_3 = \pi_N \cdot b_N(killer) \cdot a_{N,N} \cdot b_N(clown)$
`x1,y1,4:`	`killer/N clown/A`	$p_4 = \pi_N \cdot b_N(killer) \cdot a_{N,A} \cdot b_A(clown)$

Learning from Unlabeled Data

- Assume some values for $\theta = \pi, a, b$
- We can compute $P(y \mid x_\ell, \theta)$ for any y for a given x_ℓ

$$P(y \mid x_\ell, \theta) = \frac{P(x, y \mid \theta)}{\sum_{y'} P(x, y' \mid \theta)}$$

- For example, we can compute $P(\mathtt{NN} \mid \mathtt{killer\ clown}, \theta)$ as follows:

$$\frac{\pi_N \cdot b_N(killer) \cdot a_{N,N} \cdot b_N(clown)}{\sum_{i,j} \pi_i \cdot b_i(killer) \cdot a_{i,j} \cdot b_j(clown)}$$

- $P(y \mid x_\ell, \theta)$ is called the *posterior probability*

Learning from Unlabeled Data

- Compute the posterior for all possible outputs for each example in training:
- For `x1: killer clown`
  ```
  x1,y1,1:  killer/A clown/A
  x1,y1,2:  killer/A clown/N
  x1,y1,3:  killer/N clown/N
  x1,y1,4:  killer/N clown/A
  ```
  $P(\texttt{AA} \mid \texttt{killer clown}, \theta)$
  $P(\texttt{AN} \mid \texttt{killer clown}, \theta)$
  $P(\texttt{NN} \mid \texttt{killer clown}, \theta)$
  $P(\texttt{NA} \mid \texttt{killer clown}, \theta)$
- For `x2: killer problem`
  ```
  x2,y2,1:  killer/A problem/A
  x2,y2,2:  killer/A problem/N
  x2,y2,3:  killer/N problem/N
  x2,y2,4:  killer/N problem/A
  ```
  $P(\texttt{AA} \mid \texttt{killer problem}, \theta)$
  $P(\texttt{AN} \mid \texttt{killer problem}, \theta)$
  $P(\texttt{NN} \mid \texttt{killer problem}, \theta)$
  $P(\texttt{NA} \mid \texttt{killer problem}, \theta)$
- Similarly for `x3: crazy problem`
- And `x4: crazy clown`

Learning from Unlabeled Data

- For unlabeled data, the log probability of the data given θ is:

$$
\begin{aligned}
L(\theta) &= \sum_{\ell=1}^{m} \log \sum_{y} P(x_\ell, y \mid \theta) \\
&= \sum_{\ell=1}^{m} \log \sum_{y} P(y \mid x_\ell, \theta) \cdot P(x_\ell \mid \theta)
\end{aligned}
$$

- Unlike the fully observed case there is no simple solution to finding θ to maximize $L(\theta)$
- We instead initialize θ to some values, and then iteratively find better values of θ: $\theta^0, \theta^1, \ldots$ using the following formula:

$$
\begin{aligned}
\theta^t &= \operatorname*{argmax}_{\theta} Q(\theta, \theta^{t-1}) \\
&= \sum_{\ell=1}^{m} \sum_{y} P(y \mid x_\ell, \theta^{t-1}) \cdot \log P(x_\ell, y \mid \theta)
\end{aligned}
$$

Learning from Unlabeled Data

$$
\begin{aligned}
\theta^t &= \underset{\theta}{\operatorname{argmax}}\, Q(\theta, \theta^{t-1}) \\
Q(\theta, \theta^{t-1}) &= \sum_{\ell=1}^{m} \sum_{y} P(y \mid x_\ell, \theta^{t-1}) \cdot \log P(x_\ell, y \mid \theta) \\
&= \sum_{\ell=1}^{m} \sum_{y} P(y \mid x_\ell, \theta^{t-1}) \cdot \\
&\quad \left(\sum_{i} f(i, x_\ell, y) \cdot \log \pi_i \right. \\
&\quad + \sum_{i,j} f(i, j, x_\ell, y) \cdot \log a_{i,j} \\
&\quad \left. + \sum_{i,o} f(i, o, x_\ell, y) \cdot \log b_i(o) \right)
\end{aligned}
$$

Learning from Unlabeled Data

$$
\begin{aligned}
g(i, x_\ell) &= \sum_y P(y \mid x_\ell, \theta^{t-1}) \cdot f(i, x_\ell, y) \\
g(i, j, x_\ell) &= \sum_y P(y \mid x_\ell, \theta^{t-1}) \cdot f(i, j, x_\ell, y) \\
g(i, o, x_\ell) &= \sum_y P(y \mid x_\ell, \theta^{t-1}) \cdot f(i, o, x_\ell, y)
\end{aligned}
$$

$$
\begin{aligned}
\theta^t = \operatorname*{argmax}_{\pi, a, b} & \sum_{\ell=1}^{m} \sum_i g(i, x_\ell) \cdot \log \pi_i \\
& + \sum_{i,j} g(i, j, x_\ell) \cdot \log a_{i,j} \\
& + \sum_{i,o} g(i, o, x_\ell) \cdot \log b_j(o)
\end{aligned}
$$

Learning from Unlabeled Data

$$Q(\theta, \theta^{t-1}) = \sum_{\ell=1}^{m}$$

$$\sum_{i} g(i, x_\ell) \log \pi_i + \sum_{i,j} g(i, j, x_\ell) \log a_{i,j} + \sum_{i,o} g(i, o, x_\ell) \log b_i(o)$$

- The values of $\pi_i, a_{i,j}, b_i(o)$ that maximize $L(\theta)$ are:

$$\begin{aligned}
\pi_i &= \frac{\sum_\ell g(i, x_\ell)}{\sum_\ell \sum_k g(k, x_\ell)} \\
a_{i,j} &= \frac{\sum_\ell g(i, j, x_\ell)}{\sum_\ell \sum_k g(i, k, x_\ell)} \\
b_i(o) &= \frac{\sum_\ell g(i, o, x_\ell)}{\sum_\ell \sum_{o' \in V} g(i, o', x_\ell)}
\end{aligned}$$

EM Algorithm for Learning HMMs

- Initialize θ^0 at random. Let $t = 0$.
- The EM Algorithm:
  - E-step: compute expected values of y, $P(y \mid x, \theta)$ and calculate $g(i, x), g(i, j, x), g(i, o, x)$
  - M-step: compute $\theta^t = \text{argmax}_\theta\, Q(\theta, \theta^{t-1})$
  - Stop if $L(\theta^t)$ did not change much since last iteration. Else continue.
- The above algorithm is guaranteed to improve likelihood of the unlabeled data.
- In other words, $L(\theta^t) \geq L(\theta^{t-1})$
- *But* it all depends on θ^0!