CMSC35000-1 Introduction to Artificial Intelligence **Winter 2005**

Lecture 1: Date of lecture

Lecturer: Partha Niyogi *Scribe: Your name*

1.0.0.1 1. Context Free Grammar

• Now we introduce probalistic models of grammar. We will introduce HMM (Hidden Markov Model) then. HMM was first proposed in speech recognition.

• Given $w_1, w_2, ..., w_n \in L$. Recall the method to parse them using rules of the form:

$A \to BC$

$A \to w$

• The rules $1, 2, ..., N$ of these form are called CFG. We build matrices

$Ch(A, j, j) = \{ \begin{array}{c} 1 \text{ if } A \to w_j \text{ is a rule} \\ 0 \text{ otherwise} \end{array}$ (Ch is $N \times n \times n$)

$Ch(A, j, k) = 1$ if there exists a rule of the form $A \to BC$ such that $Ch(B, j, l) = 1$ and $Ch(C, l+1, k) = 1$.

$Ch(A, j, j+1) = 1$ and $A \to B(w_j)C(w_{j+1})$ if $Ch(B, j, j) = 1$ and $Ch(C, j+1, j+1) = 1$

• We fill out diagonal entries first, then other entries.

• Then we look:

$Ch(s, 1, n) = 1 \Rightarrow$ the string is acceptable

$Ch(s, 1, n) = 0 \Rightarrow$ the string is not acceptable

1.0.0.2 2. An Example of a Language

• Here we give an example of a language:

$S \to a$ (probability p)

$S \to SS$ (probability $1 - p$)

Corresponding language is a^n .

The probability of a : p

The probability of a^2 : $(1-p)p^2$

The probability of a^3 : $2(1-p)^2p^2$

......

Exercise: If $p \geq \frac{1}{2}$, sum of above will converge to 1 . If $p < \frac{1}{2}$, sum of above will not converge to 1 .

• At each stage, we start with an S and make a choice of picking a rule. Process terminates when all S are disappeared. This is a random walk on the chains. Whether it terminates depends on p only.

1.0.0.3 3. Introduction to Hidden Markov Models

• Hidden Markov Model: Another way to generate probability on strings.

• Idea: There is a state sequence, an underlying Markov chain, with a given number of states. We define states $\{s_i|i = 1, 2, 3, ..., n\}$, $A_{ij} = P[S_i \rightarrow S_j]$, and $b_j(O_k)$, the probability that at the j^{th} state O_k is observed.

• Example: We have two urns, A and B . The probabilities for choosing urns are as follows:

next $\rightarrow$	A	B
A	$\frac{1}{3}$	$\frac{2}{3}$
B	$\frac{3}{5}$	$\frac{2}{5}$

A has 9 red balls an 1 blue ball. B has 3 red balls an 7 blue balls. Observation is the sequence of balls. Hidden sequence is the sequence of states. The HMM is used to determine the most likely hidden sequence. This is the method widely used in speech recognition.

1.0.0.4 4. Experiment of Hearing

• Here is a mysterious phenomenon. We let someone see a person speaking in TV and listen to a sound. At some point, the person in TV changed his mouth shape, the listener also heard a different voice. However, the actual background sound didn't change at all. It reveals that we use visual information, in addition to acoustic information, to help us to hear and the visual information can overrule the acoustic information during the process of the sound classification in our brain.

• As another example. We let a person hear some audio tape. Originally the message was sounded like noise. Then we turned on TV and it shows a woman speaking. The experimenter understood the information and he could still understand even the TV was turned of. What we did is that we combined a loud voice of a man and a weak voice of a woman. The visual signals can be a cue to help people distinguish the voice.

• The above are examples which show our speech recognition system is complex. It involves complex techniques. We can do source separation problems easily. Assume that there are two sound waves $s_1(t)$ and $s_2(t)$. They reach a ear at different delaying T_1 and T_2 , what we receive is $s_1(t-T_1)+s_2(t-T_2)$. How do we find two or more functions from the sum of them? We have two ears, so actually we receive two different functions. Maybe its a clue to this problem. Generally, this is an open problem.

1.0.0.5 5. Computation for Hidden Markov Chain

• For example, given sequence $RRBBR$, the actual observation, we want to choose an λ such that λ maximize $P(\lambda_i|RRBBR)$. If all λ_i are likely equal, by Bayes' rule, it is equivalent to find the maximum of $P(RRBBR|\lambda_i)$.

• 3-state chain example:

State 1	$\circ$	$\circ$	$\circ$	... $\circ$
State 2	$\circ$	$\circ$	$\circ$	... $\circ$
State 3	$\circ$	$\circ$	$\circ$	... $\circ$
Observation	O_1	O_2	O_3	... O_T

Any path through this is a valid state sequence.

$P(O_1O_2O_3...O_T, q_1q_2q_3...q_t) = \pi(q_1)b_{q_1}(O_1)A_{q_1q_2}b_{q_2}(O_2)...A_{q_{T-1}q_T}b_{q_T}(O_T)$

$= \pi(q_1)b_{q_1}(O_1)\prod_{i=1}^{T-1} A_{q_iq_{i+1}}b_{q_{i+1}}(O_{i+1})$

The possibility of the observation sequence $O_1O_2O_3...O_T$ is:

$P(O_1O_2O_3...O_T) = \sum_{q_1q_2q_3\cdots q_t} P(O_1O_2O_3...O_T, q_1q_2q_3...q_t)$

• However, this computation is inefficient. We have 3^T computation on possible state sequences in this example. The longer string is, the more difficult to calculate the problem.

1.0.0.6 6. Forward Algorithm

• To compute HMM efficiently, we use forward algorithm. We define $\alpha_t(i)$ as:

$\alpha_t(i) = P(O_1 O_2 ... O_t, q_t = i)$

Observations

(1): $P(O_1 O_2 ... O_T) = \sum_{i=1}^{n} P(O_1 O_2 ... O_t, q_t = i) = \sum_{i=1}^{n} \alpha_T(i)$

(2): $\alpha_1(i) = \pi(i) b_i(O_1)$ because $P(O_1, q_1 = i) = P(q_i = i) P(O_1 | q_1 = i)$

(3): $\alpha_{t+1}(j) = P(O_1 O_2 ... O_{t+1}, q_{t+1} = j) = \sum_{i=1}^{n} P(O_1 O_2 ... O_t, q_t = i) A_{ij} b_j(O_{t+1})$

$= \sum_{i=1}^{n} \alpha_t(i) A_{ij} b_j(O_{t+1})$

This is based on Markov property $P(q_{t+1} = j | a_1 a_2 ... a_{t-1}, q_t = i) = P(q_{t+1} = j | q_t = i)$.

It gives us a method to compute $P(O_1 O_2 ... O_T)$ using $O(n^2 T)$, which is better than $O(n^T)$.

1.0.0.7 7. Backward Algorithm

• Similarly, we have backward algorithm. We define $\beta_t(i) = P(O_t O_{t+1} ... O_T, q_t = i)$.

We have formulas:

$\beta_T(i) = b_i(O_T)$

$\beta_{t-1}(j) = \sum_{i=1}^{n} \beta_t(i) A_{ji} b_i(O_{t-1})$

$P(O_1 O_2 ... O_T) = \sum_{i=1}^{n} \beta_1(i) \pi(i)$