

CSC 471/871 Lecture 3: Markov Chains and Hidden Markov Models

Stephen D. Scott

1

Outline

- Markov chains
- Hidden Markov models (HMMs)
  - Formal definition
  - Finding most probable state path (Viterbi algorithm)
  - Forward and backward algorithms
- Specifying an HMM

2

Markov Chains

An Example: CpG Islands

- Focus on nucleotide sequences
- The sequence "CG" (written "CpG") tends to appear more frequently in some places than in others
- Such CpG islands are usually  $10^2$ – $10^3$  bases long
- Questions:
  1. Given a short segment, is it from a CpG island?
  2. Given a long segment, where are its islands?

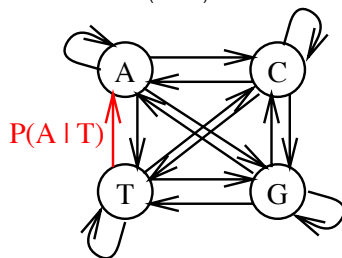
3

Modeling CpG Islands

- Model will be a CpG generator
- Want probability of next symbol to depend on current symbol
- Will use a standard (non-hidden) Markov model
  - Probabilistic state machine
  - Each state emits a symbol

4

Modeling CpG Islands (cont'd)



5

The Markov Property

- A first-order Markov model (what we study) has the property that observing symbol  $x_i$  while in state  $\pi_i$  depends only on the previous state  $\pi_{i-1}$  (which generated  $x_{i-1}$ )
- Standard model has 1-1 correspondence between symbols and states, thus

$$P(x_i | x_{i-1}, \dots, x_1) = P(x_i | x_{i-1})$$

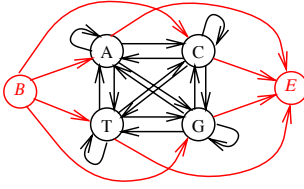
and

$$P(x_1, \dots, x_L) = P(x_1) \prod_{i=2}^L P(x_i | x_{i-1})$$

6

### Begin and End States

- For convenience, can add special "begin" ( $B$ ) and "end" ( $E$ ) states to clarify equations and define a distribution over sequence lengths
- Emit empty (null) symbols  $x_0$  and  $x_{L+1}$  to mark ends of sequence



$$P(x_1, \dots, x_L) = \prod_{i=1}^{L+1} P(x_i | x_{i-1})$$

- Will represent both with single state named 0

7

### Markov Chains for Discrimination

- How do we use this to differentiate islands from non-islands?
- Define two Markov models: islands ("+" ) and non-islands ("-")
  - Each model gets 4 states (A, C, G, T)
  - Take training set of known islands and non-islands
  - Let  $c_{st}^+$  = number of times symbol  $t$  followed symbol  $s$  in an island:

$$\bar{P}^+(t | s) = \frac{c_{st}^+}{\sum_{t'} c_{st'}^+}$$

- Example probabilities in [Durbin et al., p. 50]
- Now score a sequence  $X = \langle x_1, \dots, x_L \rangle$  by summing the log-odds ratios:

$$\log \left( \frac{\bar{P}(X | +)}{\bar{P}(X | -)} \right) = \sum_{i=1}^{L+1} \log \left( \frac{\bar{P}^+(x_i | x_{i-1})}{\bar{P}^-(x_i | x_{i-1})} \right)$$

8

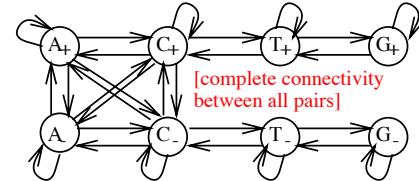
### Outline

- Markov chains
- **Hidden Markov models (HMMs)**
  - Formal definition
  - Finding most probable state path (Viterbi algorithm)
  - Forward and backward algorithms
- Specifying an HMM

9

### Hidden Markov Models

- Second CpG question: Given a long sequence, where are its islands?
  - Could use tools just presented by passing a fixed-width window over the sequence and computing scores
  - Trouble if islands' lengths vary
  - Prefer single, unified model for islands vs. non-islands



- Within the + group, transition probabilities similar to those for the separate + model, but there is a small chance of switching to a state in the - group

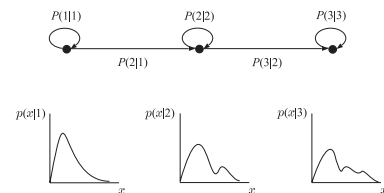
10

### What's Hidden in an HMM?

- No longer have one-to-one correspondence between states and emitted characters
  - E.g. was C emitted by  $C_+$  or  $C_-$ ?
- Must differentiate the symbol sequence  $X$  from the state sequence  $\pi = \langle \pi_1, \dots, \pi_L \rangle$ 
  - State transition probabilities same as before:  $P(\pi_i = \ell | \pi_{i-1} = j)$  (i.e.  $P(\ell | j)$ )
  - Now each state has a prob. of emitting any value:  $P(x_i = x | \pi_i = j)$  (i.e.  $P(x | j)$ )

11

### What's Hidden in an HMM? (cont'd)

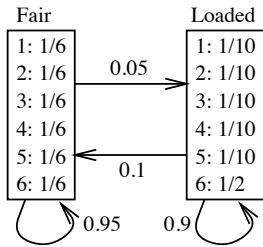


[In CpG HMM, emission probs discrete and = 0 or 1]

12

### Example: The Occasionally Dishonest Casino

- Assume that a casino is typically fair, but with probability 0.05 it switches to a loaded die, and switches back with probability 0.1



- Given a sequence of rolls, what's hidden?

13

### The Viterbi Algorithm

- Probability of seeing symbol sequence  $X$  and state sequence  $\pi$  is

$$P(X, \pi) = P(\pi_1 | 0) \prod_{i=1}^L P(x_i | \pi_i) P(\pi_{i+1} | \pi_i)$$

- Can use this to find most likely path:

$$\pi^* = \operatorname{argmax}_{\pi} P(X, \pi)$$

and trace it to identify islands (paths through "+" states)

- There are an exponential number of paths through chain, so how do we find the most likely one?

14

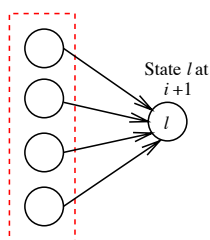
### The Viterbi Algorithm

(cont'd)

- Assume that we know (for all  $k$ )  $v_k(i)$  = probability of most likely path ending in state  $k$  with observation  $x_i$
- Then

$$v_{\ell}(i+1) = P(x_{i+1} | \ell) \max_k \{v_k(i) P(\ell | k)\}$$

All states at  $i$



15

### The Viterbi Algorithm

(cont'd)

- Given the formula, can fill in table with dynamic programming:

-  $v_0(0) = 1, v_k(0) = 0$  for  $k > 0$

- For  $i = 1$  to  $L$ ; for  $\ell = 1$  to  $M$  (# states)

$$* v_{\ell}(i) = P(x_i | \ell) \max_k \{v_k(i-1) P(\ell | k)\}$$

$$* \operatorname{ptr}_i(\ell) = \operatorname{argmax}_k \{v_k(i-1) P(\ell | k)\}$$

-  $P(X, \pi^*) = \max_k \{v_k(L) P(0 | k)\}$

-  $\pi_L^* = \operatorname{argmax}_k \{v_k(L) P(0 | k)\}$

- For  $i = L$  to 1

$$* \pi_{i-1}^* = \operatorname{ptr}_i(\pi_i^*)$$

- To avoid underflow, use  $\log(v_{\ell}(i))$  and add

16

### The Forward Algorithm

- Given a sequence  $X$ , find  $P(X) = \sum_{\pi} P(X, \pi)$

- Use dynamic programming like Viterbi, replacing max with sum, and  $v_k(i)$  with

$$f_k(i) = P(x_1, \dots, x_i, \pi_i = k) \text{ (= prob. of observed sequence through } x_i, \text{ stopping in state } k)$$

-  $f_0(0) = 1, f_k(0) = 0$  for  $k > 0$

- For  $i = 1$  to  $L$ ; for  $\ell = 1$  to  $M$  (# states)

$$* f_{\ell}(i) = P(x_i | \ell) \sum_k f_k(i-1) P(\ell | k)$$

-  $P(X) = \sum_k f_k(L) P(0 | k)$

- To avoid underflow, can again use logs, though exactness of results compromised (Section 3.6)

17

### The Backward Algorithm

- Given a sequence  $X$ , find the probability that  $x_i$  was emitted by state  $k$ , i.e.

$$P(\pi_i = k | X) = \frac{P(\pi_i = k, X)}{P(X)}$$

$$= \frac{\overbrace{P(x_1, \dots, x_i, \pi_i = k)}^{f_k(i)} \overbrace{P(x_{i+1}, \dots, x_L | \pi_i = k)}^{b_k(i)}}{P(X)}$$

computed by forward alg

- Algorithm:

-  $b_k(L) = P(0 | k)$  for all  $k$

- For  $i = L-1$  to 1; for  $k = 1$  to  $M$  (# states)

$$* b_k(i) = \sum_{\ell} P(\ell | k) P(x_{i+1} | \ell) b_{\ell}(i+1)$$

18

### Example Use of Forward/Backward Algorithm

- Define  $g(k) = 1$  if  $k \in \{A_+, C_+, G_+, T_+\}$  and 0 otherwise
- Then  $G(i | X) = \sum_k P(\pi_i = k | X) g(k) =$  probability that  $x_i$  is in an island
- For each state  $k$ , compute  $P(\pi_i = k | X)$  with forward/backward algorithm
- Technique applicable to any HMM where set of states is partitioned into classes
  - Use to label individual parts of a sequence

19

### Outline

- Markov chains
- Hidden Markov models (HMMs)
  - Formal definition
  - Finding most probable state path (Viterbi algorithm)
  - Forward and backward algorithms
- Specifying an HMM

20

### Specifying an HMM

- Two problems: defining **structure** (set of states) and **parameters** (transition and emission probabilities)
- Start with latter problem, i.e. given a training set  $X_1, \dots, X_N$  of independently generated sequences, learn a good set of parameters  $\theta$
- Goal is to maximize the (log) likelihood of seeing the training set given that  $\theta$  is the set of parameters for the HMM generating them:

$$\sum_{j=1}^N \log(P(X_j; \theta))$$

21

### When State Sequence Known

- Estimating parameters when e.g. islands already identified in training set
- Let  $A_{k\ell} =$  number of  $k \rightarrow \ell$  transitions and  $E_k(b) =$  number of emissions of  $b$  in state  $k$

$$P(\ell | k) = A_{k\ell} / \left( \sum_{\ell'} A_{k\ell'} \right)$$

$$P(b | k) = E_k(b) / \left( \sum_{b'} E_k(b') \right)$$

22

### When State Sequence Known (cont'd)

- **Be careful if little training data available**
  - E.g. an unused state  $k$  will have undefined parameters
  - Workaround: Add **pseudocounts**  $r_{k\ell}$  to  $A_{k\ell}$  and  $r_k(b)$  to  $E_k(b)$  that reflect prior biases about probabilities
  - Increased training data decreases prior's influence
  - [Sjölander et al. 96]

23

### The Baum-Welch Algorithm

- Used for estimating parameters when state sequence unknown
- Special case of the **expectation maximization** (EM) algorithm
- Start with arbitrary  $P(\ell | k)$  and  $P(b | k)$ , and use to estimate  $A_{k\ell}$  and  $E_k(b)$  as **expected** number of occurrences given the training set\*:

$$A_{k\ell} = \sum_{j=1}^N \frac{1}{P(X_j)} \sum_{i=1}^L f_k^j(i) P(\ell | k) P(x_{i+1}^j | \ell) b_\ell^j(i+1)$$

$$E_k(b) = \sum_{j=1}^N \sum_{i: x_i^j = b} P(\pi_i = k | X_j) = \sum_{j=1}^N \frac{1}{P(X_j)} \sum_{i: x_i^j = b} f_k^j(i) b_k^j(i)$$

- Use these (& pseudocounts) to recompute  $P(\ell | k)$  and  $P(b | k)$
- After each iteration, compute log likelihood and halt if no improvement

\*Superscript  $j$  corresponds to  $j$ th train example

24

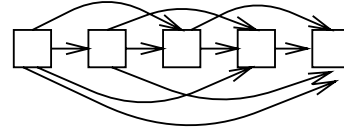
## HMM Structure

- How to specify HMM states and connections?
- States come from background knowledge on problem, e.g. size-4 alphabet, +/−, ⇒ 8 states
- Connections:
  - Tempting to specify complete connectivity and let Baum-Welch sort it out
  - **Problem:** Huge number of parameters could lead to local max
  - Better to use background knowledge to invalidate some connections by initializing  $P(\ell | k) = 0$
  - \* Baum-Welch will respect this

25

## Silent States

- May want to allow model to generate sequences with certain parts **deleted**
  - E.g. when aligning DNA or protein sequences against a fixed model or matching a sequence of spoken words against a fixed model, some parts of the input might be omitted

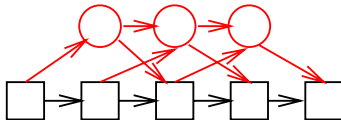


- Problem: Huge number of connections, slow training, local maxima

26

## Silent States (cont'd)

- **Silent states** (like begin and end states) don't emit symbols, so they can "bypass" a regular state



- If there are no purely silent loops, can update Viterbi, forward, and backward algorithms to work with silent states [Durbin et al., p. 71]
- Used extensively in **profile HMMs** for modeling sequences of protein families (aka **multiple alignments**)

27

Topic summary due in 1 week!

28