

# Markov Chains

Bioinformatics Example: CpG Islands

- Focus on nucleotide sequences: Sequences of symbols from alphabet {A, C, G, T}
- 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?

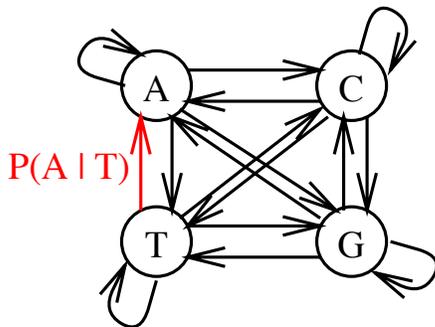
# Markov Chains

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

# Markov Chains

Modeling CpG Islands (cont'd)



# Markov Chains

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

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

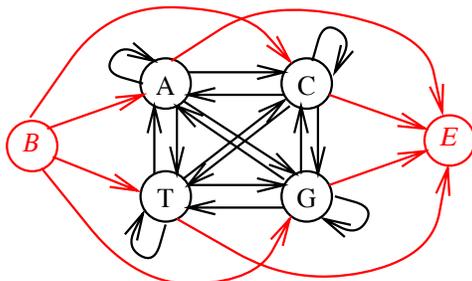
and

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

# Markov Chains

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



# Markov Chains

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:

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

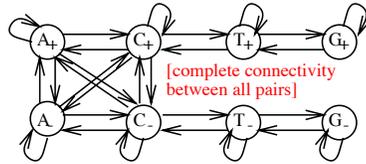
- Now score a sequence  $X = \langle x_1, \dots, x_L \rangle$  by summing the log-odds ratios:

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

# 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

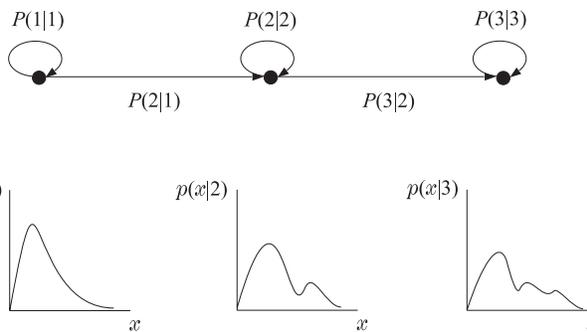
# Hidden Markov Models

What's Hidden?

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

# Hidden Markov Models

What's Hidden? (cont'd)

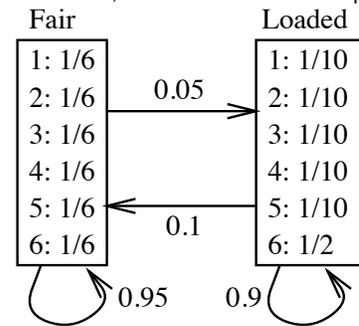


[In CpG HMM, emission probs discrete and = 0 or 1]

# Hidden Markov Models

Example: The Occasionally Dishonest Casino

Assume casino is typically fair, but with prob. 0.05 it switches to loaded die, and switches back with prob. 0.1



Given a sequence of rolls, what's hidden?

# Hidden Markov Models

The Viterbi Algorithm

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

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

- Can use this to find most likely path:

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

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

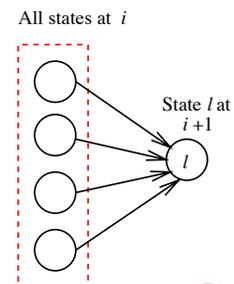
- There are an exponential number of paths through chain, so how do we find the most likely one?

# Hidden Markov Models

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

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



# Hidden Markov Models

## The Viterbi Algorithm (cont'd)

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

16 / 27

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)
  - $v_\ell(i) = P(\mathbf{x}_i | \ell) \max_k \{v_k(i-1)P(\ell | k)\}$
  - $\text{ptr}_i(\ell) = \text{argmax}_k \{v_k(i-1)P(\ell | k)\}$
- $P(X, \pi^*) = \max_k \{v_k(L)P(0 | k)\}$
- $\pi_L^* = \text{argmax}_k \{v_k(L)P(0 | k)\}$
- For  $i = L$  to 1
  - $\pi_{i-1}^* = \text{ptr}_i(\pi_i^*)$

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

# Hidden Markov Models

## The Forward Algorithm

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

17 / 27

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  $f_k(i) = P(\mathbf{x}_1, \dots, \mathbf{x}_i, \pi_i = k)$  (= prob. of observed sequence through  $\mathbf{x}_i$ , 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)
  - $f_\ell(i) = P(\mathbf{x}_i | \ell) \sum_k f_k(i-1)P(\ell | k)$
- $P(X) = \sum_k f_k(L)P(0 | k)$

To avoid underflow, can again use logs, though exactness of results compromised

# Hidden Markov Models

## The Backward Algorithm

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

18 / 27

Given a sequence  $X$ , find the probability that  $\mathbf{x}_i$  was emitted by state  $k$ , i.e.,

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

$$= \frac{\overbrace{P(\mathbf{x}_1, \dots, \mathbf{x}_i, \pi_i = k)}^{f_k(i)} \overbrace{P(\mathbf{x}_{i+1}, \dots, \mathbf{x}_L | \pi_i = k)}^{b_k(i)}}{P(X)}$$

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)
  - $b_k(i) = \sum_{\ell} P(\ell | k) P(\mathbf{x}_{i+1} | \ell) b_\ell(i+1)$

# Hidden Markov Models

## Example Use of Forward/Backward Algorithm

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

19 / 27

- 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  $\mathbf{x}_i$  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

# Hidden Markov Models

## Specifying an HMM

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

20 / 27

- 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:

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

# Hidden Markov Models

## Specifying an HMM: State Sequence Known

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

21 / 27

- 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  $k$

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

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

# Hidden Markov Models

## Specifying an HMM: State Sequence Known (cont'd)

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

22 / 27

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 parabilities
- Increased training data decreases prior's influence

# Hidden Markov Models

## Specifying an HMM: The Baum-Welch Algorithm

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

23 / 27

- Used for estimating params when state seq unknown
- Special case of *expectation maximization* (EM)
- 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<sup>1</sup>:

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

(Prob. of transition from  $k$  to  $\ell$  at position  $i$  of sequence  $j$ , summed over all positions of all sequences)

<sup>1</sup>Superscript  $j$  corresponds to  $j$ th train example

# Hidden Markov Models

## Specifying an HMM: The Baum-Welch Algorithm (cont'd)

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

24 / 27

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

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

# Hidden Markov Models

## Specifying an HMM: Structure

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

HMM Learning

Baum-Welch

Structure

25 / 27

- How to specify HMM states and connections?
- States come from background knowledge on problem, e.g., size-4 alphabet,  $+/-, \Rightarrow$  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

# Hidden Markov Models

## Specifying an HMM: Silent States

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

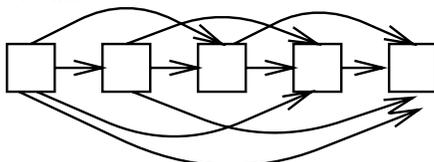
HMM Learning

Baum-Welch

Structure

26 / 27

- 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

# Hidden Markov Models

## Specifying an HMM: Silent States (cont'd)

CSOE 478/878  
Lecture 9: Hidden Markov Models  
Stephen Scott

Introduction

Outline

Markov Chains

Hidden Markov Models

Example

Viterbi

Forward Algorithm

Backward Algorithm

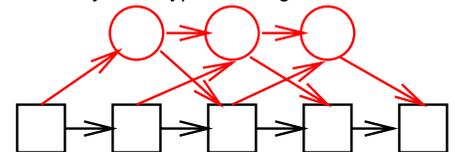
HMM Learning

Baum-Welch

Structure

27 / 27

- *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
- Used extensively in *profile HMMs* for modeling sequences of protein families (aka *multiple alignments*)