Introduction to Hidden Markov Models for Gene Prediction

ECE-S690
Outline

✓ Markov Models
✓ The Hidden Part
✓ How can we use this for gene prediction?
Learning Models

✓ Want to recognize patterns (e.g. sequence motifs), we have to learn from the data
• Stochastic process with the Markov Property
  – Stochastic processes are generally looked at as collections of random variables
  – Markov Property is simply that given the present state, future states are independent of the past.
Think of a Markov Chain as a system we can use to predict the future given the present.

Additionally in these systems, the present state only depends on two things:
- Previous state
- Probability of moving from previous state to present state
Markov Chains

Current State only depends on previous state and transition probability

\[ a_{BA} = \Pr(x_i = B | x_{i-1} = A) \]
Example: Estimating Mood State from Grad Student Observations
Grad Student come in two flavors:
  - Happy
  - Depressed about research
Each type of grad student has it’s own Markov chain associated with it.
Finally, there are three locations we can observe the grad students at:
  - Lab
  - Coffee Shop
  - Bar
Example: “Happy” Grad Student Markov Chain

Observations:
Lab, Coffee, Lab, Coffee, Lab, Lab, Bar, Lab, Coffee,…
Depressed about research
Evaluating Observations

✓ The probability of observing a given sequence is equal to the product of all observed transition probabilities.

\[
\Pr(x_1) \prod_{i=2}^{L} \Pr(x_i \mid x_{i-1})
\]

✓ \(P(\text{Coffee} \rightarrow \text{Bar} \rightarrow \text{Lab}) = \)

\[
P(\text{Coffee}) P(\text{Bar} \mid \text{Coffee}) P(\text{Lab} \mid \text{Bar})
\]

\[
P(\text{CBL}) = P(\text{L} \mid \text{B}) P(\text{B} \mid \text{C})P(\text{C})
\]
1st order model

✓ Probability of Next State | Previous State
  ✓ Calculate all probabilities

• Note that there are a number of model orders for Markov Chains. For the purposes of this lecture we will stick with 1st order models
  – Simply calculate Probability of next state given current state
  – Calculate all such probabilities to form a matrix of possible transitions
Convert “Depressed” Observations to Matrix
Scoring Observations: Depressed Grad Student

<table>
<thead>
<tr>
<th>From</th>
<th>From Lab</th>
<th>From Coffee Shop</th>
<th>From Bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Lab</td>
<td>0.1</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>To Coffee Shop</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>To Bar</td>
<td>0.8</td>
<td>0.75</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Pr from each state add to 1

Student 1: LLLLBCCLLBBLLL
Student 2: LCBLBBBCBBBBL
Student 3: CCLLLLBCCLLL
# Scoring Observations: Depressed Grad Student

<table>
<thead>
<tr>
<th></th>
<th>From Lab</th>
<th>From Coffee Shop</th>
<th>From Bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Lab</td>
<td>0.1</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>To Coffee Shop</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>To Bar</td>
<td>0.8</td>
<td>0.75</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Pr from each state add to 1

Student 1: LLLCBLCLBBLL

p's
## Scoring Observations: Depressed Grad Student

<table>
<thead>
<tr>
<th></th>
<th>From Lab</th>
<th>From Coffee Shop</th>
<th>From Bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Lab</td>
<td>0.1</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>To Coffee Shop</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>To Bar</td>
<td>0.8</td>
<td>0.75</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Pr from each state add to 1

Student 1: LLLCBCLLLBBLL

\[
\text{Student 1: LLLCBCLLLBBLL } = (0.1)(0.1)(0.1)(0.75)(0.1)(0.05)(0.1)(0.8)(0.7)(0.2)(0.1) = 4.2 \times 10^{-9}
\]
### Scoring Observations: Depressed Grad Student

<table>
<thead>
<tr>
<th></th>
<th>From Lab</th>
<th>From Coffee Shop</th>
<th>From Bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Lab</td>
<td>0.1</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>To Coffee Shop</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>To Bar</td>
<td>0.8</td>
<td>0.75</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Pr from each state add to 1

Student 1: `LLLLCBCLLLBBLL` = $4.2 \times 10^{-9}$
Student 2: `LCBLBBBCBBBBBL` = $4.3 \times 10^{-5}$
Student 3: `CCLLCLLCBCLLLL` = $3.8 \times 10^{-11}$
## Equilibrium State

<table>
<thead>
<tr>
<th></th>
<th>From Lab</th>
<th>From Coffee Shop</th>
<th>From Bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Lab</td>
<td>0.333</td>
<td>0.333</td>
<td>0.333</td>
</tr>
<tr>
<td>To Coffee Shop</td>
<td>0.333</td>
<td>0.333</td>
<td>0.333</td>
</tr>
<tr>
<td>To Bar</td>
<td>0.333</td>
<td>0.333</td>
<td>0.333</td>
</tr>
</tbody>
</table>

Student 1: LLLCBCLLLBBLL = 5.6x10^-6
Student 2: LCBLBBCBBBBL = 5.6x10^-6
Student 3: CCCLCCCCBBCCL = 5.6x10^-6
Comparing to Equilibrium States

\[
\frac{\prod_i p_{x_iy_i}}{\prod_i q_{x_i}q_{y_i}}
\]

Likelihood Ratios:

- Simply the ratio of the computed probability of the string of observations given the original chain, divided by the equilibrium.
Evaluation Observations

✓ Likelihood ratios:
  ✓ Student 1 = \( \frac{4.2 \times 10^{-9}}{5.6 \times 10^{-6}} = 7.5 \times 10^{-4} \)
  ✓ Student 2 = \( \frac{4.3 \times 10^{-5}}{5.6 \times 10^{-6}} = 7.7 \)
  ✓ Student 3 = \( \frac{3.8 \times 10^{-11}}{5.6 \times 10^{-6}} = 6.8 \times 10^{-6} \)

✓ Log likelihood ratios
  ✓ Student 1 = -3.2
  ✓ Student 2 = 0.9 (Most likely sad)
  ✓ Student 3 = -5.2
The model could represent Research Breakthrough (Happy) Student!: Transition Probabilities

<table>
<thead>
<tr>
<th></th>
<th>From Lab</th>
<th>From Coffee Shop</th>
<th>From Bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Lab</td>
<td>0.6</td>
<td>0.75</td>
<td>0.5</td>
</tr>
<tr>
<td>To Coffee Shop</td>
<td>0.25</td>
<td>0.2</td>
<td>0.45</td>
</tr>
<tr>
<td>To Bar</td>
<td>0.15</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Combined Model

Happy Student

Depressed Student
Generalized HMM - Combined Model
Simplifying the Markov Chains to 0th order to model hidden states

- Describe the probability of being in a particular state overall instead of having all the transition probabilities

- **Happy Student:**
  - Lab 75%
  - Coffee 20%
  - Bar 5%

- **Sad Student:**
  - Lab 40%
  - Coffee 20%
  - Bar 40%
HMM - Combined Model

Start - Happy - Depressed - End

L: 0.75  C: 0.2  B: 0.05
L: 0.4    C: 0.2  B: 0.4
Hiddenness

- Now we have general information about the relationship between state and location
- If we simply observe the locations of the student can we tell what mood they are in?
  - Mood is Hidden
  - Observations are the locations of the students
  - Parameters of the model are the probabilities of a student being in a particular location
Evaluating Hidden State

✓ Evaluating Hidden State

✓ Observations:

LLLCBCLLBBLLCBLBBCBBBBLCLLLCCL

Hidden state:

HHHHHHHHHHHHHDDDDDDDDHHHHHHH
Applications

• Cryptanalysis
  – The study of obtaining encrypted information without access to the secret information which is required to decode it.

• Speech Recognition
  – Identify the person who is speaking knowing only what is being said and a model for probable speakers.

• Machine Translation
  – Use computers to translate from one language to another.

• Gene Prediction
  – Predicting when a gene is present based on nucleotide observations.
Particulars about HMMs

- HMMs ultimately need to be trained to be truly effective
- Give the system a series of observations and allow the model to adjust its parameters accordingly
- In the gene finding example we feed the system a series of nucleotide sequences that are known to be genes and non-genes.
Gene Prediction

• What we want:
  – Find coding and noncoding regions of an unlabeled string of DNA nucleotides
• What’s the motivation:
  – Annotate genomic data which is becoming abundant due to next generation sequencing methods
  – Gain insights into the mechanisms involved in transcription, splicing and other processes
Why are HMMs a good fit for DNA and Amino Acids?

- DNA sequences are in a particular order which is necessary for HMMs (can’t have unordered data)
- Lots of training data is available for us to train the system on what is a gene and what is not a gene
HMM Caveats

- States are supposed to be independent of each other and this isn’t always true
- Need to be mindful of overfitting
  - Need a good training set
  - More training data does not always mean a better model
- HMMs can be slow (if proper Decoding not implemented)
  - Some decoding maps out all paths through the model
  - DNA sequences can be very long so processing/annotating them can be very time consuming
Genomic Applications

✓ Finding Genes
✓ Finding Pathogenicity Islands
Example Bio App: Pathogenicity Islands

- Clusters of genes acquired by horizontal transfer
  - Present in pathogenic species but not others
- Frequently encode virulence factors
  - Toxins, secondary metabolites, adhesins
- (Flanked by repeats, regulation and have different codon usage)
- Different GC content than rest of genome

*Neisseria meningitidis, 52% G+C*

(from Tettelin et al. 2000. Science)
Modeling Sequence Composition (Simple Probability of Sequence)

- Calculate sequence distribution from known islands
- Count occurrences of A,T,G,C
- Model islands as nucleotides drawn independently from this distribution

... C C T A A G T T A G A G G A T T G A G A A ...

A: 0.15
T: 0.13
G: 0.30
C: 0.42

A: 0.15
T: 0.13
G: 0.30
C: 0.42

A: 0.15
T: 0.13
G: 0.30
C: 0.42

P(S_iIMP)
The Probability of a Sequence (Simplistic)

✓ Can calculate the probability of a particular sequence (S) according to the pathogenicity island model (MP)

\[
P(S | MP) = P(S_1, S_2, ..., S_N | MP) = \prod_{j=1}^{N} P(S_j | MP)
\]

Example

\[
S = AAATGCGCATTTTCGAA
\]

\[
P(S | MP) = P(A)^6 \times P(T)^4 \times P(G)^3 \times P(C)^2
\]

\[
= (0.15)^6 \times (0.13)^4 \times (0.30)^3 \times (0.42)^2
\]

\[
= 1.55 \times 10^{-11}
\]
A More Complex Model

Background

0.15

0.25

Island

0.85

0.75

A: 0.25
T: 0.25
G: 0.25
C: 0.25

A: 0.15
T: 0.13
G: 0.30
C: 0.42

TAAGAATTGTGTGTCACACACACATAAAAAACCCTAAGTTAGAGGATTGAGATTGGCA
GACGATTGTTCGTGATAATAAACAAGGGGGGCATAGATCAGGCTCATTGGGC
A Generative Model

\[
\begin{array}{c}
P \quad B \\ P \quad B \\ P \quad B \\ P \quad B \\ P \quad B \\ P \quad B \\ P \quad B \\ P \\ \end{array}
\]

\[S: \quad G \quad C \quad A \quad A \quad A \quad T \quad G \quad C\]

\[
P(L_{i+1}|L_i) & B_{i+1} & P_{i+1} \\ B_i & 0.85 & 0.15 \\ P_i & 0.25 & 0.75 \\
\]

\[
P(SIB) & A: 0.25 \\ T: 0.25 \\ G: 0.25 \\ C: 0.25 \\
\]

\[
P(SIP) & A: 0.42 \\ T: 0.30 \\ G: 0.13 \\ C: 0.15 \\
\]
The Hidden in HMM

- DNA does not come conveniently labeled (i.e. Island, Gene, Promoter)
- We observe nucleotide sequences

- The hidden in HMM refers to the fact that state labels, L, are not observed
  - Only observe emissions (e.g. nucleotide sequence in our example)
A Hidden Markov Model

Hidden States
L = { 1, ..., K }

Transition probabilities
\( a_{kl} \) = Transition probability from state k to state l

Emission probabilities
\( e_k(b) = P( \text{emitting } b \mid \text{state}=k) \)

Initial state probability
\( \pi(b) = P(\text{first state}=b) \)
HMM with Emission Parameters

- $a_{13}$: Probability of a transition from State 1 to State 3
- $e_2(A)$: Probability of emitting character A in state 2
Hidden Markov Models (HMM)

✓ Allows you to find sub-sequence that fit your model
✓ Hidden states are disconnected from observed states
✓ Emission/Transition probabilities
✓ Must search for optimal paths
Three Basic Problems of HMMs

✓ The Evaluation Problem
  ✓ Given an HMM and a sequence of observations, what is the probability that the observations are generated by the model?

✓ The Decoding Problem
  ✓ Given a model and a sequence of observations, what is the most likely state sequence in the model that produced the observations?

✓ The Learning Problem
  ✓ Given a model and a sequence of observations, how should we adjust the model parameters in order to maximize evaluation/decoding
Fundamental HMM Operations

**Computation**

**Decoding**
- ✓ Given an HMM and sequence $S$
- ✓ Find a corresponding sequence of labels, $L$

**Evaluation**
- ✓ Given an HMM and sequence $S$
- ✓ Find $P(S | \text{HMM})$

**Training**
- ✓ Given an HMM w/o parameters and set of sequences $S$
- ✓ Find transition and emission probabilities that maximize $P(S | \text{params, HMM})$

**Biology**
- Annotate pathogenicity islands on a new sequence
- Score a particular sequence
- Learn a model for sequence composed of background DNA and pathogenicity islands
Markov chains and processes

1st order Markov chain

2nd order Markov chain

1st order with stochastic observations -- HMM
Order & Conditional Probabilities

Order

0th  \[ P(\text{ACTGTC}) = p(A) \times p(C) \times p(T) \times p(G) \times p(T) \ldots \]

1st  \[ P(\text{ACTGTC}) = p(A) \times p(C|A) \times p(T|C) \times p(G|T) \ldots \]

2nd  \[ P(\text{ACTGCG}) = p(A) \times p(C|A) \times p(T|AC) \times p(G|CT)\ldots \]

\[ P(T|AC) \]

*Probability of T given AC*
HMM - Combined Model for Gene Detection
1st-order transition matrix (4x4)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.2</td>
<td>0.15</td>
<td>0.25</td>
<td>0.2</td>
</tr>
<tr>
<td>C</td>
<td>0.3</td>
<td>0.35</td>
<td>0.25</td>
<td>0.2</td>
</tr>
<tr>
<td>G</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>T</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>
2nd Order Model (16x4)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>0.1</td>
<td>0.3</td>
<td>0.25</td>
<td>0.05</td>
</tr>
<tr>
<td>AC</td>
<td>0.05</td>
<td>0.25</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>AG</td>
<td>0.3</td>
<td>0.05</td>
<td>0.1</td>
<td>0.25</td>
</tr>
<tr>
<td>AT</td>
<td>0.25</td>
<td>0.1</td>
<td>0.05</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Three Basic Problems of HMMs

✓ The Evaluation Problem
  ✓ Given an HMM and a sequence of observations, what is the probability that the observations are generated by the model?

✓ The Decoding Problem
  ✓ Given a model and a sequence of observations, what is the most likely state sequence in the model that produced the observations?

✓ The Learning Problem
  ✓ Given a model and a sequence of observations, how should we adjust the model parameters in order to maximize
What Questions can an HMM Answer?

Viterbi Algorithm:
What is the most probable path that generated sequence $X$?

Forward Algorithm:
What is the likelihood of sequence $X$ given HMM $M$ – $Pr(X|M)$?

Forward-Backward (Baum-Welch) Algorithm:
What is the probability of a particular state $k$ having generated symbol $X_i$?
“Decoding” With HMM

Given observations, we would like to predict a sequence of hidden states that is most likely to have generated that sequence.

Pathogenicity Island Example

Given a nucleotide sequence, we want a labeling of each nucleotide as either “pathogenicity island” or “background DNA”
The Most Likely Path

✓ Given observations, one reasonable choice for labeling the hidden states is:

\[ L^* = \arg \max_{\text{Labels}} P(\text{Labels, Sequence} | \text{Model}) \]

The sequence of hidden state labels, L*, (or path) that makes the labels and sequence most likely given the model.
Probability of a Path, Seq

\[ P = P(G \mid B)P(B_1 \mid B_0)P(C \mid B)P(B_2 \mid B_1)P(A \mid B)P(B_3 \mid B_2)\ldots P(C \mid B_7) \]

\[ = (0.85)^7 \times (0.25)^8 \]

\[ = 4.9 \times 10^{-6} \]
Probability of a Path, Seq

We could try to calculate the probability of every path, but....

\[
P = P(G|B)P(B_1|B_0)P(C|B)P(B_2|B_1)P(A|B)P(P_3|B_2)...P(C|B_7)
\]

\[
= (0.85)^3 \times (0.25)^6 \times (0.75)^2 \times (0.42)^2 \times 0.30 \times 0.15
\]

\[
= 6.7 \times 10^{-7}
\]
Decoding

✓ Viterbi Algorithm
  ✓ Finds most likely sequence of hidden states or labels, L* or P* or π*, given sequence and model

\[ L^* = \arg \max_{\text{labels}} P(Labels, Sequence | Model) \]

✓ Uses *dynamic programming* (same technique used in sequence alignment)
✓ Much more efficient than searching every path
Finding Best Path

- Viterbi
- Dynamic programming
- Maximize Probability Emission of observations on trace-back
Viterbi Algorithm

Most probable state path given sequence (observations)?
Viterbi (in pseudocode)

✓ $l$ is previous state and $k$ is next state
✓ $v_l(i) = e_l(x_i) \max_k(v_k(i-1)a_{kl})$
✓ $\pi^*$ are the paths that maximizes the probability of the previous path times new transition in $\max_k(v_k(i-1)a_{kl})$

Each node picks one max
\[
\max P \left( TG \right) = \max \left\{ \begin{array}{c}
\max P \left( TG \right) \cdot P \left( \begin{array}{c}
\text{match 2} \\
A: 0.04 \\
T: 0.01 \\
\end{array} \right)
\end{array} \right.
\]
Forward Alg: Probability of a Single Label (Hidden State)

- Calculate most probable label, $L^*_i$, at each position $i$
- Do this for all $N$ positions gives us $\{L^*_1, L^*_2, L^*_3, \ldots, L^*_N\}$

Forward algorithm (dynamic programming) $P(\text{Label}_5=\text{BIS})$

Sum over all paths

$\sum_k f_k(i-1) a_{kl}$
Forward Algorithm

\[ f_i(i) = e_i(x_i) \sum_k f_k(i-1) a_{kl} \]

Start

Add probs of all
Different paths to get
Probability of sequence

\[ P(x) = \sum_k f_k(N) a_{k0} \]
Two Decoding Options

✓ Viterbi Algorithm
   ✓ Finds most likely sequence of hidden states, $L^*$ or $P^*$ or $\pi^*$, given sequence and model

$$L^* = \arg \max_{\text{labels}} P(\text{Labels} \mid \text{Sequence}, \text{Model})$$

✓ Posterior Decoding
   ✓ Finds most likely label at each position for all positions, given sequence and model

$$\{L^*_1, L^*_2, L^*_3, \ldots, L^*_N\}$$

✓ Forward and Backward equations
Relation between Viterbi and Forward

**VITERBI**

\[ V_j(i) = P(\text{most probable path ending in state } j \text{ with observation } i) \]

**Initialization:**

\[ V_0(0) = 1 \]
\[ V_k(0) = 0, \text{ for all } k > 0 \]

**Iteration:**

\[ V_l(i) = e_l(x_i) \max_k V_k(i-1) a_{kl} \]

**Termination:**

\[ P(x, \pi^*) = \max_k V_k(N) \]

**FORWARD**

\[ f_l(i) = P(x_1 \ldots x_i, \text{state}_i=l) \]

**Initialization:**

\[ f_0(0) = 1 \]
\[ f_k(0) = 0, \text{ for all } k > 0 \]

**Iteration:**

\[ f_l(i) = e_l(x_i) \Sigma_k f_k(i-1)a_{kl} \]

**Termination:**

\[ P(x) = \Sigma_k f_k(N)a_{k0} \]
Forward/Backward Algorithms

✓ Way to compute probability of most probable path
✓ Forward and Backward can be combined to find Probability of emission, $x_i$ from state $k$ given sequence $x$. $P(\pi_i=k \mid x)$
✓ $P(\pi_i=k \mid x)$ is called posterior decoding
✓ $P(\pi_i=k \mid x) = f_k(l)b_k(l)/P(x)$
Example Application: *Bacillus subtilis*

Mining *Bacillus subtilis* chromosome heterogeneities using hidden Markov models

Pierre Nicolas¹,²,*, Laurent Bize³, Florence Muri², Mark Hoebeke¹, François Rodolphe¹, S. Dusko Ehrlich³, Bernard Prum² and Philippe Bessières¹

¹Laboratoire de Mathématique, Informatique et Génome, INRA, Route de Saint-Cyr, F-78026 Versailles cedex, France, ²Laboratoire de Statistique et Génome, CNRS, Tour Évry2, 523 place des terrasses de l’Agora, F-91034 Évry, France and ³Laboratoire de Génétique Microbienne, INRA, F-78352 Jouy-en-Josas cedex, France
Method

Three State Model

Gene+

Gene-

AT Rich

Second Order Emissions

$P(S_i) = P(S_i | \text{State}, S_{i-1}, S_{i-2})$
(capturing trinucleotide Frequencies)

Train using EM

Predict w/Posterior Decoding

Nicolas et al (2002) NAR
Results

Gene on positive strand

Gene on negative strand

A/T Rich
- Intergenic regions
- Islands

Each line is $P(labellS, model)$
color coded by label

Nicolas et al (2002) NAR
Training an HMM

**Transition probabilities**
e.g. $P(P_{i+1|B_i})$ – the probability of entering a pathogenicity island from background DNA

**Emission probabilities**
i.e. the nucleotide frequencies for background DNA and pathogenicity islands

![Diagram of HMM model with states B and P, emission probabilities P(SIB) and P(SIP), and transition probability P(L_{i+1}|L_i).]
Learning From Labelled Data

If we have a sequence that has islands marked, we can simply count

If we have a sequence that has islands marked, we can simply count

<table>
<thead>
<tr>
<th></th>
<th>B_{i+1}</th>
<th>P_{i+1}</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_i</td>
<td>3/5</td>
<td>1/5</td>
<td>1/5</td>
</tr>
<tr>
<td>P_i</td>
<td>1/3</td>
<td>2/3</td>
<td>0</td>
</tr>
<tr>
<td>Start</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

P(L_{i+1}|L_i)

P(SIB)

<table>
<thead>
<tr>
<th></th>
<th>A:</th>
<th>T:</th>
<th>G:</th>
<th>C:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A:</td>
<td>1/5</td>
<td></td>
<td>0</td>
<td>2/5</td>
</tr>
<tr>
<td>T:</td>
<td></td>
<td></td>
<td>2/5</td>
<td></td>
</tr>
<tr>
<td>G:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

P(SIP)

ETC..
Unlabelled Data

How do we know how to count?

L: start → B → B → B → B → B → B → B → B → B → B → B → End

S: G C A A A T G C

P(L_{i+1}|L_i)

<table>
<thead>
<tr>
<th></th>
<th>B_{i+1}</th>
<th>P_{i+1}</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_i</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_i</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

P(SIB)

<table>
<thead>
<tr>
<th></th>
<th>A:</th>
<th>T:</th>
<th>G:</th>
<th>C:</th>
</tr>
</thead>
</table>

P(SIP)

|   | A: | T: | G: | C: |
Unlabeled Data

An idea:

1. Imagine we start with some parameters (e.g. initial or bad model)
2. We *could* calculate the most likely path, \( P^* \), given those parameters and \( S \)
3. We *could* then use \( P^* \) to recalculate our parameters by maximum likelihood
4. And iterate (to convergence)
Training Models for Classification

✓ Correct Order for the model
✓ Higher order models remember more “history”
✓ Additional history can have predictive value
  ✓ Example:
    ✓ predict the next word in this sentence fragment
    ✓ “…finish ___” (up, it, first, last, …?)
    ✓ now predict it given more history
    ✓ “Fast guys finish ___”
Model Order

✓ However, the number of parameters to estimate grows exponentially with the order for modeling DNA we need parameters for an nth order model, with $n \geq 5$ normally

✓ The higher the order, the less reliable we can expect our parameter estimates to be
  ✓ estimating the parameters of a 2nd order Markov chain from the complete genome of E. Coli, each word > 72,000 times on average
  ✓ estimating the parameters of an 8th order chain, word 5 times on average
HMMs in Context

✓ HMMs
  ✓ Sequence alignment
  ✓ Gene Prediction

✓ Generalized HMMs
  ✓ Variable length states
  ✓ Complex emissions models
  ✓ e.g. Genscan

✓ Bayesian Networks
  ✓ General graphical model
  ✓ Arbitrary graph structure
  ✓ e.g. Regulatory network analysis
HMMs can model different regions

Figure 4.8: The structure of a gene with some of the important signals shown.
Example Model for Gene Recognition

Start → Promoter → Transcription Factor → Exon → Splice → Intron

Repeat

End
Another Example

[Diagram showing a network of exons and intergenic DNA with annotations:
- Gene on forward strand
- Gene on reverse strand]
CpG Islands: Another Application

- CG dinucleotides are rarer in eukaryotic genomes than expected given the independent probabilities of C, G

- Particularly, the regions upstream of genes are richer in CG dinucleotides than elsewhere - *CpG islands*
CpG Islands

CpG island DNA states:
- large C, G transition probabilities

“Normal DNA” states:
- small C, G transition probabilities

Most transitions omitted for clarity
CpG Islands

- In human genome, CG dinucleotides are relatively rare
  - CG pairs undergo a process called methylation that modifies the C nucleotide
  - A methylated C mutate (with relatively high chance) to a T
- Promotor regions are CG rich
  - These regions are not methylated, and thus mutate less often
  - These are called CG (aka CpG) islands
CpG Island Prediction

✓ In a CpG island, the probability of a “C” following a “G” is much higher than in “normal” intragenic DNA sequence.

✓ We can construct an HMM to model this by combining two HMMs: one for normal sequence and one for CpG island sequence.

✓ Transitions between the two sub-models allow the model to switch between CpG island and normal DNA.

✓ Because there is more than one state that can generate a given character, the states are “hidden” when you just see the sequence.

✓ For example, a “C” can be generated by either the $C^+$ or $C^-$ states in the following model.
Inhomogenous Markov Chains

Borodovsky’s Lab: http://exon.gatech.edu/GeneMark/
Variable-length

Full

Variable Length
Interpolated HMMs

✓ Manage Model Trade-off by interpolating between various HMM Model orders
✓ GlimmerHMM
The Three Basic HMM Problems

✓ Problem 1 (Evaluation):
  Given the observation sequence $O = o_1, \ldots, o_T$ and an HMM model, how do we compute the probability of $O$ given the model?

✓ Problem 2 (Decoding):
  Given the observation sequence $O = o_1, \ldots, o_T$ and an HMM model, how do we find the state sequence that best explains the observations?
The Three Basic HMM Problems

✓ Problem 3 (Learning): How do we adjust the model parameters to maximize the probability of observations given the model?
Conclusions

✓ Markov Models
✓ HMMs
✓ Issues
✓ Applications
Example of Viterbi, Forward, Backward, and Posterior Algorithms

Real DNA sequences are inhomogeneous and can be described by a hidden Markov model with hidden states representing different types of nucleotide composition. Consider an HMM that includes two hidden states H and L for high and lower C+G content, respectively. Initial probabilities for both H and L are equal to 0.5, while transition probabilities are as follows: $a_{HH}=0.5$, $a_{HL}=0.5$, $a_{LL}=0.6$, $a_{LH}=0.4$. Nucleotides T, C, A, G are emitted from states H and L with probabilities 0.2, 0.3, 0.2, 0.3, and 0.3, 0.2, 0.3, 0.2, respectively. Use the Viterbi algorithm to define the most likely sequence of hidden states for the sequence, $X=\text{TGC}$. 