Hidden Markov Models

based on chapters from the book
Durbin, Eddy, Krogh and Mitchison
Biological Sequence Analysis
Shamir’s lecture notes
and Rabiner’s tutorial on HMM

music recognition

Deal with variations in
- pitch
- timing
- timbre
- …
Stock Market Prediction

- Actual Value versus Forecasted Value for Tata Steel in Rupees over the period 5-9 2009 – 23-9 2011.
- Variations of value over time.

Activity Tracking

Activities:
- Walking
- Running
- Cycling
- stair climbing
- sleeping, etc.
application: gene finding

deal with variations in
- actual sound → actual base (match/substitutions)
- timing → insertions/deletions

Profile and multiple sequence alignment

<table>
<thead>
<tr>
<th>VGA</th>
<th>HAGEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>NVDEV</td>
</tr>
<tr>
<td>VEA</td>
<td>DVAGH</td>
</tr>
<tr>
<td>VKG</td>
<td>-----D</td>
</tr>
<tr>
<td>VYS</td>
<td>TYETS</td>
</tr>
<tr>
<td>FNA</td>
<td>NIPKH</td>
</tr>
<tr>
<td>TAGA</td>
<td>NGAGV</td>
</tr>
</tbody>
</table>

123 45678
Basic Questions

Given:
• A sequence of “observations”
• A probabilistic model of our “domain”

Questions:
• Does the given sequence belong to a certain family?
  – Markov chains
  – Hidden Markov Models (HMMs)
• Can we say something about the internal structure of the sequence? (indirect observations)
  – Hidden Markov Models (HMMs)

Introduction Markov Chain Model

Characteristics
• Discrete time
• Discrete space
• No state History
  – Present state only
• States and transitions

Notations:

- $P(X)$ probability for event $X$
- $P(X,Y)$ event $X$ and event $Y$
- $P(X|Y)$ event $X$ given event $Y$
Definition of Markov Chain Model

- A Markov chain\(^1\) model is defined by
  - a set of states
    - some states emit a symbol (unique per state)
    - other states (e.g., the begin state) are silent
  - a set of transitions with associated probabilities
    - the transitions going out of a given state define a distribution over the possible next states (i.e., all positive, and sum equals 1)

\(^1\) Марков А. А., Распространение закона больших чисел на величины, зависящие друг от друга. — Известия физико-математического общества при Казанском университете. — 2-я серия. — Том 15. (1906) — С. 135—156

Markov Model

Markov Model \( M = (Q, P, T) \), with
- \( Q \) the set of states
- \( P \) the set of initial probabilities \( p_x \) for each state \( x \) in \( Q \)
- \( T = (t_{xy}) \) the transition probabilities matrix/graph, with \( t_{xy} \) the probability of the transition from state \( x \) to state \( y \).

This is a first order Markov Model: no history is modeled

An observation \( X \) is a sequence of states:
\( X = x_1 x_2 \ldots x_n \)

The probability of an observation \( X \) given the model \( M \) is equal to:
\[
P(X|M) = p_{x_1} t_{x_1x_2} t_{x_2x_3} \ldots t_{x_{n-1}x_n} = p_{x_1} \cdot \prod_{i=2}^{n} t_{x_{i-1}x_i}
\]
A Markov Chain Model Example

- Transition probabilities
  - $\Pr(x_i=a|x_{i-1}=g)=0.16$
  - $\Pr(x_i=c|x_{i-1}=g)=0.34$
  - $\Pr(x_i=g|x_{i-1}=g)=0.38$
  - $\Pr(x_i=t|x_{i-1}=g)=0.12$

$\sum_{x_i} \Pr(x_i | x_{i-1} = g) = 1$

over all neighbors $x_i$

The Probability of a Sequence for a Markov Chain Model

$\Pr($CGGT$) = \Pr(C)\Pr(G|C)\Pr(G|G)\Pr(T|G)$
Markov Chains: Another Example

\( M_1:\)

\[ Q = \{ A, B, C \} \]

\[ P = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \]

\( \) 

unique starting state A

\[ T = \begin{pmatrix} A & B & C \\ 0.7 & 0.3 & 0 \\ 0 & 0.2 & 0.8 \\ 0.4 & 0 & 0.6 \end{pmatrix} \]

\( A \) \( B \) \( C \)

\[ \begin{array}{c}
A \\
B \\
C \\
\end{array} \rightarrow
\begin{array}{c}
A \\
B \\
C \\
\end{array} \]

\[ P( \text{AABBCCC} \mid M_1) = 1 \cdot 7 \cdot 3 \cdot 2 \cdot 8 \cdot 6 \cdot 10^{-6} = 1.2 \times 10^{-2} \]

\( M_2:\)

\[ Q = \{ A, B, C \} \]

\[ P = \begin{pmatrix} 0.6 & 0.4 & 0.3 \\ 0.4 & 0.6 & 0.5 \end{pmatrix} \]

\( \) 

\[ \begin{array}{c}
A \\
B \\
C \\
\end{array} \rightarrow
\begin{array}{c}
A \\
B \\
C \\
\end{array} \]

\[ P( \text{AABBCCC} \mid M_2) = 1 \cdot 6 \cdot 4 \cdot 3 \cdot 6 \cdot 5 \cdot 10^{-6} = 1.1 \times 10^{-2} \]

Markov Models: Properties

Given some sequence \( x \) of length \( L \), we can ask:

How probable is the sequence \( x \) given our model \( M \)?

- For any probabilistic model of sequences, we can write this probability as

\[ \Pr(x) = \Pr(x_L, x_{L-1}, \ldots, x_1) = \Pr(x_L \mid x_{L-1}, \ldots, x_1) \Pr(x_{L-1} \mid x_{L-2}, \ldots, x_1) \ldots \Pr(x_1) \]

- key property of a (1st order) Markov chain: the probability of each \( x_i \) depends only on the value of \( x_{i-1} \)

\[ \Pr(x) = \Pr(x_L \mid x_{L-1}) \Pr(x_{L-1} \mid x_{L-2}) \ldots \Pr(x_2 \mid x_1) \Pr(x_1) \]

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

- initial state $x_0$ fixed
- initial probabilities
- final state \[ \text{[not depicted]} \]

$X = x_1x_2 \ldots x_n$

$P(X|M) = \prod_{i=1}^{n} t_{x_{i-1}x_i}$

small values: underflow

$t_{0x} = p_x$

$\log P(X|M) = \sum_{i=1}^{n} \log t_{x_{i-1}x_i}$

Markov Model: Comparing Models

Given:

$X = x_1x_2 \ldots x_n$

$P(X|M) = \prod_{i=1}^{n} t_{x_{i-1}x_i}$

Question: X best explained by which model?

i.e., we would like to know: $P(M_1 | X)$ vs. $P(M_2 | X)$

But can only calculate: $P(X | M_1)$ vs. $P(X | M_2)$

Bayes Rule: $P(A|B) = P(B|A) \cdot P(A) / P(B)$

$\frac{P(M_1|X)}{P(M_2|X)} = \frac{P(X|M_1) \cdot P(M_1)}{P(X|M_2) \cdot P(M_2)}$
Motivation for Markov Models in Computational Biology

- There are many cases in which we would like to represent the statistical regularities of some class of sequences
  - genes
  - various regulatory sites in DNA (e.g., where RNA polymerase and transcription factors bind)
  - proteins in a given family
- Markov models are well suited to this type of task
Markov Chain: An Example Application

- **CpG islands**
  - *CG* di-nucleotides are *rarer* in *eukaryotic genomes* than expected given the marginal probabilities of *C* and *G*
  - but the regions upstream of genes (*reading is from 5’ to 3’*) are *richer* in *CG* di-nucleotides than elsewhere – so called *CpG islands*
  - useful evidence for finding genes

- Application: Predict *CpG islands* with Markov chains
  - a Markov chain to represent *CpG islands*
  - a Markov chain to represent the rest of the genome

Markov Chains for Discrimination

- Suppose we want to distinguish *CpG islands* from other sequence regions
- Given sequences from *CpG islands*, and sequences from other regions, we can construct
  - a model to represent *CpG islands*
  - a null model to represent the other regions

- We can then score a test sequence $X$ by:

$$score(X) = \log \frac{\Pr(X \mid \text{CpGModel})}{\Pr(X \mid \text{nullModel})}$$
Markov Chains for Discrimination

We can use the scoring function:

\[ \text{score}(X) = \log \frac{\Pr(X \mid \text{CpG Model})}{\Pr(X \mid \text{null Model})} \]

- Because according to Bayes’ rule we have:

\[
\begin{align*}
\Pr(CpG \mid X) &= \frac{\Pr(X \mid CpG) \Pr(CpG)}{\Pr(X)} \\
\Pr(null \mid X) &= \frac{\Pr(X \mid null) \Pr(null)}{\Pr(X)}
\end{align*}
\]

- If we are not taking into account prior probabilities (\( \Pr(\text{CpG}) \) and \( \Pr(\text{null}) \)) of the two classes, then from Bayes’ rule it is clear that we just need to compare \( \Pr(X|\text{CpG}) \) and \( \Pr(X|\text{null}) \) as is done in our scoring function \( \text{score()} \).

Markov Chain Application: CpG islands

<table>
<thead>
<tr>
<th>observed frequencies</th>
<th>island (+)</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.180</td>
<td>0.274</td>
<td>0.426</td>
<td>0.120</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.171</td>
<td>0.368</td>
<td>0.274</td>
<td>0.188</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0.161</td>
<td>0.339</td>
<td>0.375</td>
<td>0.125</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.079</td>
<td>0.355</td>
<td>0.384</td>
<td>0.182</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>non island (-)</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.300</td>
<td>0.205</td>
<td>0.285</td>
<td>0.210</td>
</tr>
<tr>
<td>C</td>
<td>0.322</td>
<td>0.298</td>
<td>0.078</td>
<td>0.302</td>
</tr>
<tr>
<td>G</td>
<td>0.248</td>
<td>0.246</td>
<td>0.298</td>
<td>0.208</td>
</tr>
<tr>
<td>T</td>
<td>0.177</td>
<td>0.239</td>
<td>0.292</td>
<td>0.292</td>
</tr>
</tbody>
</table>

In general consecutive CG pairs \( \text{CG} \rightarrow \text{CG} \) are rare, although [islands]
Occur in signal (e.g.) promotor regions.
basic questions

Observation: DNA sequence
Model 1: CpG islands
Model 2: non-islands

• does this sequence belong to a certain family?  
  Markov chains
  is this a CpG island (or not)?

• can we say something about the internal structure?  
  Markov Chains: windowing
  where are the CpG islands?

application: CpG islands

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>0.180</td>
<td>0.274</td>
<td>0.426</td>
<td>0.120</td>
<td>-</td>
<td>0.300</td>
<td>0.205</td>
<td>0.285</td>
<td>0.210</td>
</tr>
<tr>
<td></td>
<td>0.171</td>
<td>0.368</td>
<td>0.274</td>
<td>0.188</td>
<td></td>
<td>0.322</td>
<td>0.298</td>
<td>0.078</td>
<td>0.302</td>
</tr>
<tr>
<td>+</td>
<td>0.161</td>
<td>0.339</td>
<td>0.375</td>
<td>0.125</td>
<td>-</td>
<td>0.248</td>
<td>0.246</td>
<td>0.298</td>
<td>0.208</td>
</tr>
<tr>
<td></td>
<td>0.079</td>
<td>0.355</td>
<td>0.384</td>
<td>0.182</td>
<td></td>
<td>0.177</td>
<td>0.239</td>
<td>0.292</td>
<td>0.292</td>
</tr>
</tbody>
</table>

X = ACGT

\[
\frac{P(X | \text{ island})}{P(X | \text{ non})} = \frac{\prod_{i=1}^{n} t_{x_{i-1}x_{i}}}{\prod_{i=1}^{n} t_{-x_{i-1}x_{i}}}
\]

X = ACGT  A->C  C->G  G->T  Note: A score > 1 is an Indication of a CpG island.

0.274 · 0.274 · 0.125 = 2.82

0.205 · 0.078 · 0.208
**application: CpG islands**

LLR = Log-Likelihood Ratio

\[
\log\left(\frac{t_{xy}^+}{t_{xy}^-}\right)
\]

<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>-0.74</td>
<td>0.42</td>
<td>0.58</td>
<td>-0.80</td>
</tr>
<tr>
<td>C</td>
<td>-0.91</td>
<td>0.30</td>
<td><strong>1.81</strong></td>
<td>-0.69</td>
</tr>
<tr>
<td>G</td>
<td>-0.62</td>
<td>0.46</td>
<td>0.33</td>
<td>-0.73</td>
</tr>
<tr>
<td>T</td>
<td>-1.17</td>
<td>0.57</td>
<td>0.39</td>
<td>-0.68</td>
</tr>
</tbody>
</table>

log-score \((\log_2)\)

\[
\log \frac{P(X|\text{island})}{P(X|\text{non})} = \log \prod_{i=1}^{n} t_{x_{i-1}x_i}^+ = \sum_{i=1}^{n} \log\left(\frac{t_{x_{i-1}x_i}^+}{t_{x_{i-1}x_i}^-}\right)
\]

\[
X = ACGT
\]

\[
\log_2 \frac{0.274 \cdot 0.274 \cdot 0.125}{0.205 \cdot 0.078 \cdot 0.208} = 0.42 + 1.81 - 0.73 = 1.50
\]

**CpG Log-Likelihood Ratio**

\[
\log\left(\frac{t_{xy}^+}{t_{xy}^-}\right)
\]

<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>-0.74</td>
<td>0.42</td>
<td>0.58</td>
<td>-0.80</td>
</tr>
<tr>
<td>C</td>
<td>-0.91</td>
<td>0.30</td>
<td><strong>1.81</strong></td>
<td>-0.69</td>
</tr>
<tr>
<td>G</td>
<td>-0.62</td>
<td>0.46</td>
<td>0.33</td>
<td>-0.73</td>
</tr>
<tr>
<td>T</td>
<td>-1.17</td>
<td>0.57</td>
<td>0.39</td>
<td>-0.68</td>
</tr>
</tbody>
</table>

LLR(ACGT) = 0.42 + 1.81 – 0.73 = 1.50 (0.37 ‘bits’ per base)

\[
1.5/4 = 0.375
\]

- is a (short) sequence a CpG island?
  - compare with observed data (normalized for length)
- where (in long sequence) are CpG islands?
  - first approach: sliding window

- ! What would be the length of window?
• is a (short) sequence a CpG island?
  compare with observed data (normalized for length)

Figure 3.2. The histogram of the length-normalised scores for all the sequences. CpG islands are shown with dark grey and non-CpG with light grey.

• where (in long sequence) are CpG islands?
  first approach: sliding window
CpGplot

observed vs. expected

Window size 100
C and G contents =>
expected CG occurrences

%C + %G

putative islands

A set of 10 windows fulfilling the thresholds before island is called

Islands of unusual CG composition
EMBOSS_001 from 1 to 286
Observed/Expected ratio > 0.60
Percent C + Percent G > 50.00
Length > 50
Length 114 (51..164)

Some Notes on: Higher Order Markov Chains

- The Markov property specifies that the probability of a state depends only on the probability of the previous state
- But we can build more “memory” into our states by using a higher order Markov model
- In an n-th order Markov model

\[
\Pr(x_i \mid x_{i-1}, x_{i-2}, \ldots, x_1) = \Pr(x_i \mid x_{i-1}, \ldots, x_{i-n})
\]

The probability of the current state depends on the previous n states.
Selecting the Order of a Markov Chain Model

- But the number of parameters we need to estimate for an $n$-th order Markov model grows exponentially with the order.
  - For modeling DNA we need $O(4^{n+1})$ parameters (# of state transitions) for an $n$-th order model.

- 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 (5.44 x 10^6 bases), we would see each (length 3) word ~ 85,000 times on average (divide by 4^3).
  - Estimating the parameters of a 9th order chain, we would see each (length 10) word ~ 5 times on average (divide by 4^10 ~ 10^6).

Higher Order Markov Chains

- An $n$-th order Markov chain over some alphabet $A$ is equivalent to a first order Markov chain over the alphabet of $n$-tuples: $A^n$.

- Example: a 2nd order Markov model for DNA can be treated as a 1st order Markov model over alphabet

  AA, AC, AG, AT
  CA, CC, CG, CT
  GA, GC, GG, GT
  TA, TC, TG, TT

  Transition probabilities: $P(A|AA)$, $P(A|AC)$, etc.
A Fifth Order Markov Chain Equivalent

\[
\Pr(GCTAC) = \Pr(GCTAC) \Pr(A \mid GCTAC) \Pr(C \mid GCTAC)
\]

Hidden Markov Model

Where (in long sequence) are CpG islands?

- first approach: Markov Chains + windowing
- second approach: *Hidden Markov Model*
Given observed sequence AGGCT, which state emits which item?

Another example: Eddy (2004)

Posterior decoding $P(s_i=q \mid X)$, i.e., given sequence X what is the probability that the i-th state is equal to q.

An (toy) HMM for 5’ splice site recognition.

Figure from: What is a hidden Markov model?
Example: weather

Emission probabilities

- P(\(\text{\(\text{H}\)}\)) = 0.1
- P(\(\text{\(\text{M}\)}\)) = 0.2
- P(\(\text{\(\text{L}\)}\)) = 0.7

Transition probabilities

- \(0.6\) from H to M
- \(0.3\) from M to H
- \(0.4\) from M to L

Initial probabilities

- \(p_{\text{H}} = 0.4\)
- \(p_{\text{M}} = 0.2\)
- \(p_{\text{L}} = 0.4\)

Observed weather vs. pressure

\(\text{P(\(\text{HHHHH}\) | \(\text{HHHHH}\)}) = 1 \cdot 2 \cdot 7 \cdot 7 = 196 \cdot 10^{-5}\)}

\(\text{P(\(\text{RCCSS}\) | \(\text{MMMM}\)) = 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 432 \cdot 10^{-5}\)}

\(\text{P(\(\text{RCCSS}, \text{HHHHH}\) | \(\text{MMMM}\)) = 4 \cdot 1 \cdot 6 \cdot 2 \cdot 6 \cdot 2 \cdot 6 \cdot 6 \cdot 7 \cdot 7 = 1016 \cdot 10^{-7}\)}

\(\text{P(\(\text{RCCSS}, \text{MMMM}\)) = 2 \cdot 3 \cdot 2 \cdot 4 \cdot 2 \cdot 4 \cdot 2 \cdot 3 \cdot 3 = 14 \cdot 10^{-7}\)}
CpG islands ctd.

8 states $A^+$ vs $A^-$
unique observation each state

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+$</td>
<td>0.180</td>
<td>0.274</td>
<td>0.426</td>
<td>0.120</td>
</tr>
<tr>
<td>C</td>
<td>0.171</td>
<td>0.368</td>
<td>0.274</td>
<td>0.188</td>
</tr>
<tr>
<td>G</td>
<td>0.161</td>
<td>0.339</td>
<td>0.375</td>
<td>0.125</td>
</tr>
<tr>
<td>T</td>
<td>0.079</td>
<td>0.355</td>
<td>0.384</td>
<td>0.182</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-$</td>
<td>0.300</td>
<td>0.205</td>
<td>0.285</td>
<td>0.210</td>
</tr>
<tr>
<td>A</td>
<td>0.322</td>
<td>0.298</td>
<td>0.078</td>
<td>0.302</td>
</tr>
<tr>
<td>C</td>
<td>0.248</td>
<td>0.246</td>
<td>0.298</td>
<td>0.208</td>
</tr>
<tr>
<td>G</td>
<td>0.177</td>
<td>0.239</td>
<td>0.292</td>
<td>0.292</td>
</tr>
</tbody>
</table>

‘$+$’ denotes CpG island
‘$-$’ denotes non-CpG island

hidden Markov model

model $M = (\Sigma, Q, T)$
- states $Q$
- transition probabilities $t_{pq}$, $p, q \in Q$

observation $X = x_1 x_2 \ldots x_n \in \Sigma^*$
observe states indirectly ‘hidden’
- emission probabilities $e_{px}, p \in Q, x \in \Sigma, e_p(x)$

probability
observation given the model
? there may be many state seq’s
HMM main questions

Given HMM \( M \):
- probability of observation \( X \)?
- most probable state sequence?
- how to find the parameters of the model \( M \)? *training*

Three Important Questions
(See also L.R. Rabiner (1989))

- How likely is a given sequence?
  - The Forward algorithm (probability over all paths)
- What is the most probable “path” for generating a given sequence?
  - The Viterbi algorithm
- How can we learn the HMM parameters given a set of sequences?
  - The Forward-Backward (Baum-Welch) algorithm
Given sequence X: most probable state vs. most probable path

* most probable state (over all state sequences)
  posterior decoding
  using forward & backward probabilities
* most probable path (= single state sequence)
  Viterbi

The Forward Algorithm: probability of observation X

\[ f_q(i) \text{ probability ending in state } q \text{ emitting symbol } x_i \]

\[ f_q(i) = \sum_{p \in Q} f_p(i-1) t_{pq} e_q(x_i) \]
The Forward Algorithm: probability of observation $X$

probability observing $x_1, \ldots, x_i$ and ending in state $q$:

$$f_q(i) = P(x_1 \ldots x_i, \pi_i = q)$$

$$f_q(i) = \sum_{p \in Q} f_p(i-1) t_{pq} e_q(x_i)$$

‘forward’ probability

$$P(X) = \sum_{p \in Q} f_p(n) t_{p*}$$

* = end-state

---

Probability of observation: weather

Initial state:

- Remain in H
- Coming from M
- Coming from L

Transitions:

- Remain in H
- Coming from M
- Coming from L

Start: 0 1

<table>
<thead>
<tr>
<th>Transition</th>
<th>Probability</th>
<th>Calculation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>H 0 4 1 = 4</td>
<td>4 6 +6 4 +24 1 2 = 144 (x10⁻⁴)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M 0 2 3 = 6</td>
<td>4 3 +6 2 +24 5 4 = 576 (x10⁻⁴)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L 0 4 6 = 24</td>
<td>4 1 +6 4 +24 4 3 = 372 (x10⁻⁴)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HMM: posterior decoding

Given \( X \) the prob. that the \( i-th \) state equals \( q: P(\pi_i = q \mid X) \)

- forward:
  \[ f_q(i) = P(x_1 \ldots x_i, \pi_i = q) \]

- backward:
  \[ b_q(i) = P(x_{i+1} \ldots x_n \mid \pi_i = q) \]

\[ P(X, \pi_i = q) = f_q(i)b_q(i) \quad \Rightarrow \quad P(\pi_i = q \mid X) = \frac{f_q(i)b_q(i)}{P(X)} \]

HMM main questions

- observation \( X \in \Sigma^* \Rightarrow \) most probable state sequence

- again:
  We cannot try all possibilities
  Viterbi

- probability of this observation?
- most probable state sequence?
- how to find the model? training
Viterbi algorithm

most probable state sequence for observation $X$

1. **Dynamic Programming**: $v_q(i)$ probability ending in state $q$ and emitting $x_i$

$$v_q(i) = \max_{p \in Q} v_p(i-1) \cdot t_{pq} \cdot e_q(x_i)$$

2. **Traceback**: most probable state sequence

Decoding Problem: The Viterbi algorithm

1. **Dynamic Programming**: max probability ending in state
2. **Traceback**: most probable state sequence

$$v_q(i) = \max_{p \in Q} v_p(i-1) \cdot t_{pq} \cdot e_q(x_i)$$
Another decoding method, **Posterior Decoding**:

**Input:**
Given a Hidden Markov Model \( M = (\Sigma, Q, \Theta) \) and a sequence \( X \) for which the generating path \( P \) is unknown.

**Question:**
For each \( 1 \leq i \leq L \) (the length of the path \( P \)) and state \( q \) in \( Q \) compute the probability: \( P(\pi_i = q \mid X) \).

\[ P(\pi_i = q \mid X) \]

**Posterior Decoding Problem**

\( P(\pi_i = q \mid X) \) gives two additional decoding possibilities:
1. Alternative ‘path’ \( P^* \) that follows the max probability states:
   \[ \arg\max_{\text{state } q} \{ P(\pi_i = q \mid X) \} \].
2. Define a function \( g(q) \) on the states \( q \) in \( Q \), then
   \[ G(i \mid X) = \sum_q \{ P(\pi_i = q \mid X) \cdot g(q) \} \]

We can use 2) to calculate the posterior probability of each nucleotide of \( X \) to be in a CpG-island, using a function \( g(q) \) defined on all states \( q \) in \( Q \):

\[ g(q) = \begin{cases} 1 & \text{for all } q \text{ that are CpG-island states}, \\ 0 & \text{otherwise}. \end{cases} \]
HMM Decoding: two explanations

posterior $\Sigma$
best state every position
But: path may not be allowed by model

viterbi max
optimal global path
But: many paths with similar probability

dishonest casino dealer
Observation
36616366466232534413661661163252562462255265262266435353336
Viterbi
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL

Compare to:
Forward
FFLLLLLLLLLLLLLLFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL

Posterior (total)
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
Learning if correct path is known

- Learning is simple if we **know the correct path** for each sequence in our training set

- estimate parameters by counting the number of times each parameter is used across the training set

![Diagram](https://via.placeholder.com/150)

**Sketch: Parameter estimation**

training sequences $X^{(i)}$

optimize score $\prod_{i=1}^{n} P(X^{(i)} | \Theta)$ for model $\Theta$.

*If state sequences are known*

- count transitions $pq$ $A_{pq}$
- count emissions $b$ in $p$ $E_p(b)$

divide by

- total transitions in $p$
- emissions in $q$

Laplace correction for dealing with 'zero' probabilities.
Adding 1 to each count.
Learning With Hidden State

- If we *don’t know the correct path* for each sequence in our training set, consider all possible paths for the sequence.

- Estimate parameters through a procedure that counts the expected number of times each parameter is used across the training set.

Learning Parameters: The Baum-Welch Algorithm

- Here we use the Forward-Backward algorithm.
- An Expectation Maximization (EM) algorithm
  - EM is a family of algorithms for learning probabilistic models in problems that involve hidden states.
- In this context, the hidden state is the path that best explains each training sequence.

- Note, finding the parameters of the HMM that optimally explains the given sequences is NP-Complete!
**HMM: state sequences unknown: Baum-Welch**

**Baum-Welch training**

- Based on given HMM $\Theta$
- Given a training set of sequences $X$

- Determine:
  - expected number of transitions and
  - expected number of emissions

- Apply ML and build a new (better) model:
  - ML tries to find a model that gives the training data the highest likelihood
- Iterate until convergence.

**Note:**
- can get stuck in local maxima
- does not understand the semantics of the states

---

**Baum-Welch Re-estimation**

For the re-estimation we need the expected counts For the transitions and the emissions in the HMM:

- Apply the backward-forward algorithm.

**Probability of state $q$ when emitting $X_i$:**

$$P(\pi_i = q \mid X) = \frac{f_q(i)b_q(i)}{P(x)}$$

**Probability of transition $(p,q)$ after emitting $X_i$:**

$$P(\pi_i = p, \pi_{i+1} = q \mid X, \Theta) = \frac{f_p(i)\cdot t_{pq} \cdot e_q(x_{i+1}) \cdot b_q(i+1)}{P(X)}$$
Estimation of Transition Probability

\[ A_{pq} \text{ sum over all training sequences } X \text{ sum over all positions } i \]

Estimation of Emission Probability

\[ E_p(b) \text{ sum over all training sequences } X \text{ sum over all positions } i \text{ with } x_i=b \]

Estimate parameters by ratio of expected counts.

Baum-Welch training

concerns:
- guaranteed to converge target score, not \( \Theta \)
- unstable solutions !
- local maximum

practical
- small values \( \rightarrow \) renormalize

tips:
- repeat for several initial HMM \( \Theta \)
- start with meaningful HMM \( \Theta \)
Viterbi training (sketch):

- determine optimal paths
- re-compute as if paths are known
- score may decrease!

Computational Complexity of HMM Algorithms

- Given an HMM with $S$ states and a sequence of length $L$, the complexity of the Forward, Backward and Viterbi algorithms is $O(S^2L)$
  - This assumes that the states are densely interconnected
- Given $M$ training sequences of length $L$, the complexity of Baum Welch on each iteration is $O(MS^2L)$


Furthermore: