

## Chap. 3 Markov chains and hidden Markov models

Biointelligence Laboratory  
School of Computer Sci. & Eng.  
Seoul National University  
Seoul 151-742, Korea

This slide file is available online at  
<http://bi.snu.ac.kr/>

Copyright (c) 2002 by SNU CSE Biointelligence Lab

1

## In this Chapter

- The probabilistic model for sequence analysis
  - ◆ HMM (hidden Markov model) or, its simpler version, Markov model
  - ◆ ‘Does this sequence belong to a particular family?’
  - ◆ ‘Assuming the sequence does come from some family, what can we say about its internal structure?’
    - Identify alpha helix or beta sheet regions in a protein sequence.

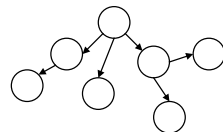
Copyright (c) 2002 by SNU CSE Biointelligence Lab

2

## Probabilistic Models

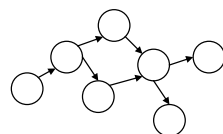


Quantized speech signal  
⊕ £ ψ † £



- Word recognition
- Language understanding

Probabilistic model



- Gene finding
- Gene/Protein families



ACCGAAG  
Biological sequences

Copyright (c) 2002 by SNU CSE Biointelligence Lab

3

## Example: CpG Islands

- In the human genome, CpG dinucleotides are rarer than would be expected from the independent probabilities of C and G because of biological reasons.
  - ◆ C → methylation → T
- In the promoters or start regions of many genes, there might exist many more CpGs than elsewhere in the genome.
  - ◆ C-methylation is usually suppressed in such areas.
  - ◆ These regions are called CpG islands (a few hundred to a few thousand bases long).

Copyright (c) 2002 by SNU CSE Biointelligence Lab

4

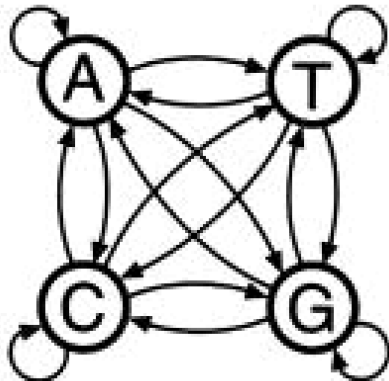
## CpG Islands (Cont'd)

- Two raised questions
  - ◆ Given a short stretch of genomic sequence, how would we decide if it comes from a CpG island or not?
  - ◆ Given a long piece of sequence, how would we find the CpG islands in it, if there are any?

## Markov Chains

- What sort of model might we use for CpG island regions?
  - ◆ Dinucleotides are important. → a model in which the probability of a symbol depends on the previous symbol.
- Markov chains (the simplest one)
  - ◆ A collection of 'states' → represented as a node or vertex
    - Each state corresponds to a particular residue.
  - ◆ Arrows between states. → represents the transition from one state to another.

## Markov Chains for DNA Sequences



Transition probabilities

$$a_{st} = P(x_i = t \mid x_{i-1} = s)$$

## Markov Properties

The probability of the sequence  $x = x_1 x_2 \dots x_L$

$$P(x) = P(x_L, \dots, x_2, x_1)$$

$$= P(x_L \mid x_{L-1}, \dots, x_2, x_1)$$

$$\cdot P(x_{L-1} \mid x_{L-2}, \dots, x_2, x_1) \dots P(x_1) \quad \text{Chain rule}$$

By the Markov properties  $P(x_i \mid x_{i-1}, \dots, x_2, x_1) = P(x_i \mid x_{i-1})$

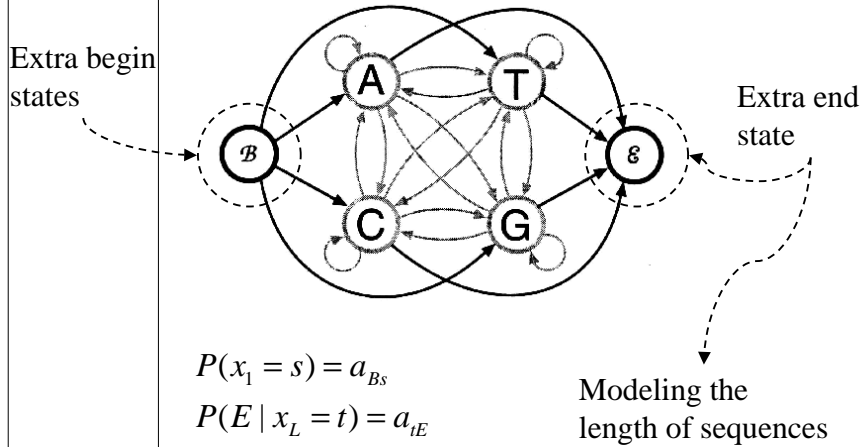
$$P(x) = P(x_L, \dots, x_2, x_1)$$

$$= P(x_L \mid x_{L-1}) P(x_{L-1} \mid x_{L-2}) \dots P(x_1)$$

$$= P(x_1) \prod_{i=2}^L a_{x_{i-1} x_i}$$

Transition probabilities

# Modeling the Beginning and End of Sequences



# Markov Chains for Discrimination

- Real data for the CpG island example
  - ◆ From a set of human DNA sequences, a total of 48 putative CpG islands were extracted.
  - ◆ Two Markov chain models for the CpG island example were derived.

- '+' model and '-' model
  - Transition probabilities are estimated from data as follows:

$$a_{st}^+ = \frac{c_{st}^+}{\sum_{t'} c_{st'}^+}$$

- Maximum likelihood estimators

# Markov Chains for Discrimination (Cont'd)

- Two resulting tables

+	A	C	G	T	-	A	C	G	T
A	0.180	0.274	0.426	0.120	A	0.300	0.205	0.285	0.210
C	0.171	0.368	0.274	0.188	C	0.322	0.298	0.078	0.302
G	0.161	0.339	0.375	0.125	G	0.248	0.246	0.298	0.208
T	0.079	0.355	0.384	0.182	T	0.177	0.239	0.292	0.292

- ◆ Each row sums to one.
- ◆ The table is asymmetric.

# Likelihood Ratio Test

- Calculate the log-odds ratio

$$S(x) = \log \frac{P(x | \text{model } +)}{P(x | \text{model } -)} = \sum_{i=1}^L \log \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-}$$

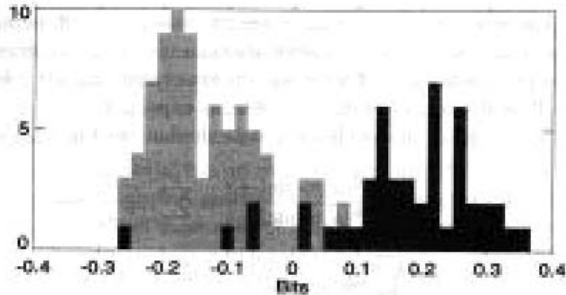
likelihood

$$= \sum_{i=1}^L \beta_{x_{i-1}x_i}$$

- A table for  $\beta$  (log likelihood ratio)

$\beta(\log_2)$	A	C	G	T
A	-0.740	0.419	0.580	-0.803
C	-0.913	0.302	1.812	-0.0685
G	-0.624	0.461	0.331	-0.730
T	-1.169	0.573	0.393	-0.679

## The Distribution of Scores, $S(x)$



- Normalized by the length of the sequence
- Dark gray: +, light gray: -
- Reasons of errors: inadequate models or mislabeling of the training sequences

Copyright (c) 2002 by SNU CSE Biointelligence Lab

13

## Extensions to the Previous Model

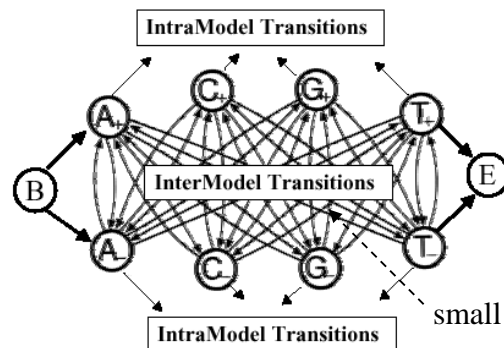
- The second question:
  - ♦ Given a long piece of sequence, how would we find the CpG islands in it, if there are any?
  - ♦ Use of Markov chains
    - Calculate the log-odds score for a window of 100 nucleotides around every nucleotide in the sequence.
    - CpG islands will stand out with positive values. → inadequate
      - What if the CpG islands have sharp boundaries.
      - Why the window size of 100?
- We need the more satisfactory model for this question.

Copyright (c) 2002 by SNU CSE Biointelligence Lab

14

## Extensions to the Previous Model (Cont'd)

- Simulate in one model the 'islands' in a 'sea' of non-island genomic sequences
  - ♦ Both the Markov chains in one model



It is no longer possible to tell what state the model was in when  $x_i$  was generated just by looking at  $x_i$ .

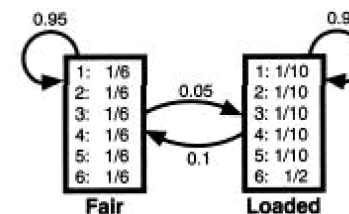
→ **hidden** Markov models

Copyright (c) 2002 by SNU CSE Biointelligence Lab

15

## Hidden Markov Models

- Transition probabilities of the states (Markov properties)
 
$$a_{kl} = P(\pi_i = l \mid \pi_{i-1} = k)$$
- Emission probabilities of the symbols (generative models)
 
$$e_k(b) = P(x_i = b \mid \pi_i = k)$$
- Example: the occasionally dishonest casino



Hidden states → which die

The joint probability of a sequence  $x$  and a path  $\pi$

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

Copyright (c) 2002 by SNU CSE Biointelligence Lab

16

## Example: the Joint Probability in CpG Islands

- The probability of sequence CGCG being emitted by the state  $(C_+, G_-, C_-, G_+)$  in the model is

$$a_{0,C_+} \times 1 \times a_{C_+G_-} \times 1 \times a_{G_-C_-} \times 1 \times a_{C_-G_+} \times 1 \times a_{G_+0}$$

- ◆ In general, we do not know the path.  $\rightarrow$  estimate the path.
- ◆ The most probable one  $\rightarrow$  the Viterbi algorithm
- ◆ Based on a *a posteriori* distribution over states

## The Underlying Path of States

- Observed sequence  $\rightarrow$  *decode* the sequence of the underlying states.
  - ◆ From the speech recognition field
- There may be many state sequences that could give rise to any particular sequence of symbols.
  - ◆  $(C_+, G_+, C_+, G_+)$ ,  $(C_-, G_-, C_-, G_-)$ , and  $(C_+, G_-, C_+, G_-) \rightarrow$  CGCG

The most probable

The least probable

## The Most Probable State Path

- The most probable path  $\pi^*$  (one solution)
  - ◆  $\pi^* = \operatorname{argmax}_{\pi} P(x, \pi)$
- Recursive construction of the most probable path
  - ◆ The Viterbi algorithm
  - ◆  $v_k(i)$ : the probability of the most probable path ending in state  $k$  with observation  $i$ .
  - ◆  $v_0(0) = 1$
  - ◆  $v_k(i+1) = e_i(x_{i+1}) \max_k (v_k(i) a_{kl})$

## Viterbi Algorithm

### Algorithm: Viterbi

Initialisation ( $i = 0$ ):  $v_0(0) = 1, v_k(0) = 0$  for  $k > 0$ .

Recursion ( $i = 1 \dots L$ ):  $v_l(i) = e_i(x_i) \max_k (v_k(i-1) a_{kl})$   
 $\text{ptr}_l(i) = \operatorname{argmax}_k (v_k(i-1) a_{kl})$ .

Termination:  $P(x, \pi^*) = \max_k (v_k(L) a_{k0})$ ;  
 $\pi_L^* = \operatorname{argmax}_k (v_k(L) a_{k0})$ .

Traceback ( $i = L \dots 1$ ):  $\pi_{i-1}^* = \text{ptr}_i(\pi_i^*)$ .

The logarithm of the probabilities is used because of computational reasons.

## Example: Viterbi

- The model of CpG islands (CGCG)

$\pi$		C	G	C	G
$\mathcal{B}$	1	0	0	0	0
$A_+$	0	0	0	0	0
$C_+$	0	0.13	0	0.012	0
$G_+$	0	0	0.034	0	0.0032
$T_+$	0	0	0	0	0
$A_-$	0	0	0	0	0
$C_-$	0	0.13	0	0.0026	0
$G_-$	0	0	0.010	0	0.00021
$T_-$	0	0	0	0	0

## Example: Casino, Part 2

- Generated rolls and Viterbi path finding

```

Rolls  315116246446644245311321631164152133625144543631656626566666
Die    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls  65116645313265124563666463163666316232645523626666625151631
Die    LLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Viterbi LLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls  222555441666566563564324364131513465146353411126414626253356
Die    FFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls  36616366646623253441366166116325256246225526525226643535336
Die    LLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Viterbi LLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls  233121625364414432335163243633665562466662632666612355245242
Die    FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
    
```

## The Probability of a Sequence

- The probability of sequences
  - Discrimination between CpG islands and non-island regions
- $P(x) = \sum_{\pi} P(x, \pi)$ 
  - Enumerating all the possible  $\pi$  is impractical.
  - One solution is to use  $P(x, \pi)$  instead of  $P(x)$ .
    - Somewhat startling but surprisingly good in many cases.
- The value of  $P(x)$  can also be calculated by dynamic programming skills.

$$f_k(i) = P(x_1, \dots, x_i, \pi_i = k)$$

$$f_l(i+1) = e_l(x_{i+1}) \sum_k f_k(i) a_{kl}$$

## The Forward Algorithm

### Algorithm: Forward algorithm

Initialisation ( $i = 0$ ):  $f_0(0) = 1, f_k(0) = 0$  for  $k > 0$ .

Recursion ( $i = 1 \dots L$ ):  $f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}$

Termination:  $P(x) = \sum_k f_k(L) a_{k0}$

- The logarithm of the probabilities can be used because of computational reasons.

- In this case, scaling of the probabilities is more appropriate.

## Another Question

- The Viterbi algorithm → find the most probable path
- Forward algorithm → calculate the probability of a sequence
- We might want to know what the most probable state is for an observation  $x_i$ .
  - ◆ The posterior probability of state  $k$  at time  $i$  when the emitted sequence is known.
  - ◆  $P(\pi_i = k | x)$

## The Backward Algorithm

- The posterior probability of state  $k$  at time  $i$  given the observed sequence  $x$ ,  $P(\pi_i = k | x)$ .

$$P(x, \pi_i = k) = P(x_1 \dots x_i, \pi_i = k) P(x_{i+1} \dots x_L | x_1 \dots x_i, \pi_i = k)$$

$$\approx \underbrace{P(x_1 \dots x_i, \pi_i = k)}_{f_k(i)} \underbrace{P(x_{i+1} \dots x_L | \pi_i = k)}_{b_k(i)}$$

Initialization ( $i = L$ ):  $b_k(L) = a_{k0}$  for all  $k$

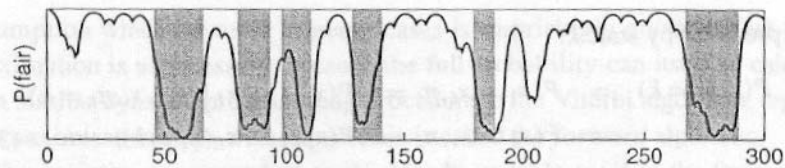
Recursion ( $i = L - 1, \dots, 1$ ):  $b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i + 1)$

Termination:  $P(x) = \sum_l a_{0l} e_l(x_1) b_l(1)$

$$P(\pi_i = k | x) = \frac{P(x, \pi_i = k)}{P(x)} = \frac{f_k(i) b_k(i)}{P(x)}$$

## Example: Casino, Part 3

- pp. 60, Figure 3.6



**Figure 3.6** The posterior probability of being in the state corresponding to the fair die in the casino example. The  $x$  axis shows the number of the roll. The shaded areas show when the roll was generated by the loaded die.

## Posterior Decoding

- When many different paths have almost the same probability as the most probable one.
- Another approximation to the optimal path

$$\hat{\pi}_i = \arg \max_k P(\pi_i = k | x)$$

- ◆ Can produce illegal paths.

- The posterior probability of the function  $g(k)$  at time  $i$  given the observed sequence  $x$ ,

$$G(i | x) = \sum_k P(\pi_i = k | x) g(k).$$

## CpG Islands Revisited

- What really concerns us is whether a base is part of an island or not.
  - ◆  $g(k) = 1$  for  $k \in \{A_+, C_+, G_+, T_+\}$
  - ◆  $g(k) = 0$  for  $k \in \{A_-, C_-, G_-, T_-\}$
  - ◆  $\rightarrow G(i|x)$  is the posterior probability according to the model that base  $i$  is in a CpG island.
    - This is not quite the most probable global labeling of a given sequence.

## Example: Prediction of CpG Islands

- By the Viterbi decoding
  - ◆ False negatives: 2
  - ◆ False positives: 121  $\rightarrow$  67 (by some post-processing)
- By posterior decoding
  - ◆ False negatives: 2
  - ◆ False positives: 236  $\rightarrow$  83 (by some post-processing)
- Some false positives are real CpG islands.
- False negatives are perhaps wrongly labeled.
- It is possible that a more sophisticated model is needed.

## Example: Casino, Part 4

- The model is changed.
  - ◆ The switching probability from fair to loaded  $\rightarrow 0.01$
  - ◆ The Viterbi decoding
    - Never visits the loaded die state.
- pp. 61, Figure 3.7

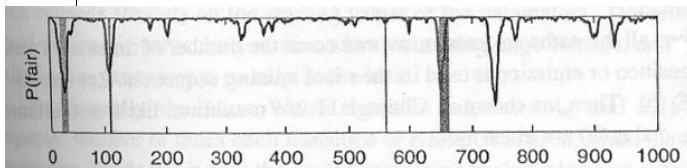


Figure 3.7 The posterior probability of the die being fair, but using probability 0.01 for switching to the loaded die (cf. Figure 3.6).