10.555 Bioinformatics
Spring 2003

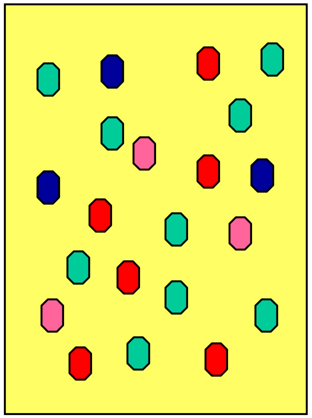
Lecture 2

Rudiments on: Dynamic programming (sequence alignment), probability and estimation (Bayes theorem) and Markov chains

Gregory Stephanopoulos
MIT

Bayes theorem

Problem: A box, containing 4 types of spheres, marked as A,T,C,G, is being sampled, yielding:



TGACGTTAAGGCTATCTCCGTAATGC

Before sampling we have no basis for any prediction, other than some model

After seeing some of the box contents we can make some predictions on:

1. How spheres are distributed in the box (model)
2. The likelihood that an A appears on the next trial
3. The probability that a different pattern has emerged

These points are intuitive. What is a *formal framework* to describe them?

Bayes theorem

$$\underbrace{P(X/Y,I)}_{\text{Posterior probability}} = \underbrace{P(X,I)}_{\text{Prior probability}} P(Y/X,I) \ / \ P(Y/I)$$

Fundamental theorem: Interchanges conditioning and non-conditioning propositions. It embodies *inference,* describes how to update our degree of belief in light of new information

Important problem: Derive a model (parametrized), M=M(w) from a body of data D:

$$P(M/D) = P(M)\ P(D/M) \ / \ P(D)$$

$$\log P(M/D) = \log \underbrace{P(D/M)}_{\textit{Data likelihood}} + \log \underbrace{P(M)}_{\textit{Prior (probability)}} - \log P(D)$$

Parameter estimation, model selection

Problem: Two models, M_1 and M_2 can be compared by comparing their probabilities $P(M_1/D)$ and $P(M_2/D)$. The *best* model *in its class* is found by determining the set of parameters w maximizing the posterior probability p(M/D), or

Min(-log P(M/D) = -log P(D/M) – log P(M) + log P(D)

This is called ***MAP estimation*** **(Maximum *a posteriori*)**

P(D) is a normalizing constant independent of optimization. If the prior P(M) is uniform over all models then the above problem is reduced to the following *Maximum Likelihood (ML)* maximization *(ML estimation):*

Min (-log P(D/M)

Parameter estimation, model selection

Problem solution:

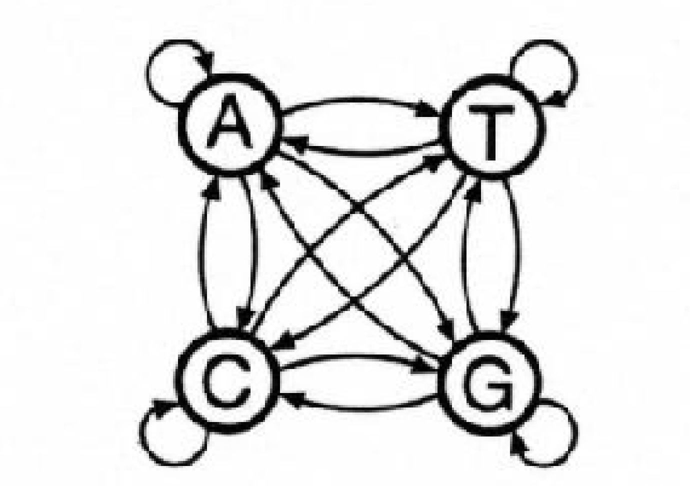
See notes

Markov Chains

See notes

First Question: *What Did We Do? The Markov Chain*

The Markov Chain for DNA



The Transition Probabilities

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

The joint probability for a sequence $\{x : x_L, x_{L-1}, x_{L-2}, \ldots, x_1\}$ is

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

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

First -Order Markov Chain for DNA Sequences

Consider a sequence of nucleotides in the following state:

$x = \{ \mathrm{A,C,G,G,C,C,A,G,T,A,C,C,G,G} \}$

Then,

$$\begin{aligned} P(x) &= P(x_L, x_{L-1}, \ldots, x_1) \\ &= P(x_L \mid x_{L-1}, \ldots, x_1) P(x_{L-1}, x_{L-2} \ldots, x_1) \end{aligned}$$

Assume now that

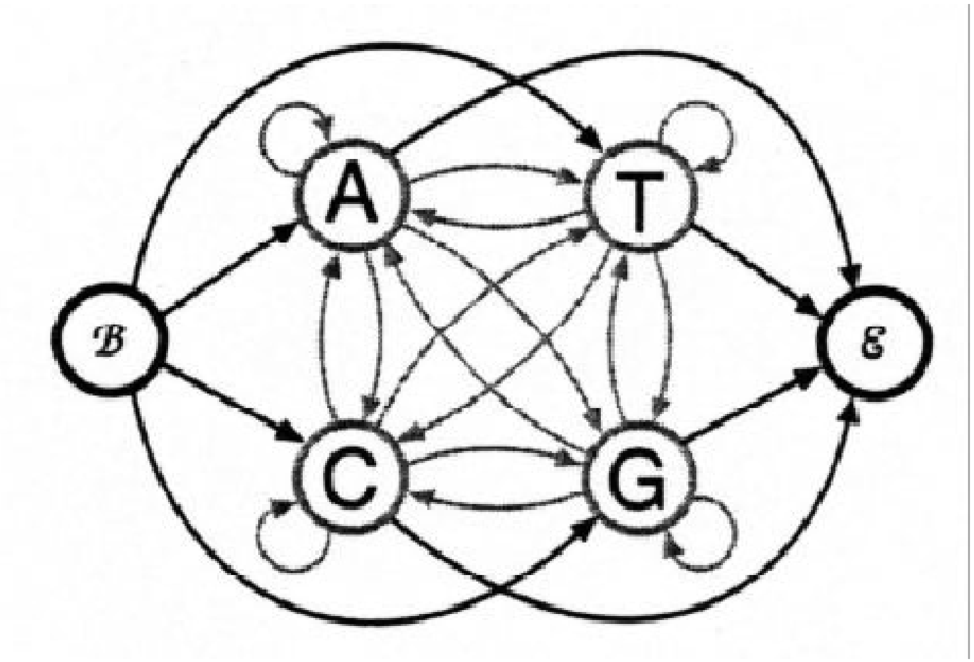
$$P(x_L \mid x_{L-1}, \ldots, x_1) = P(x_L \mid x_{L-1})$$

Then,

$$P(x) = P(x_L \mid x_{L-1}) P(x_{L-1} \mid x_{L-2}) \ldots P(x_l) = P(x_l) \Pi_i a_{x(i-1)x(i)}$$

First -Order Markov Chain for DNA

Modeling the Beginning and End of Sequences



$P(x_1=s) = a_{Bs}$

$P(E \mid x_L = t) = a_{tE}$

Note: *Usually the end of a sequence is not modelled in Markov chains. A sequence can end anywhere*

Second -Order Markov Chain for DNA Sequences

Assume a Second-Order Markov Chain

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

and note that

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

Then, instead of working with single-position states, i.e.

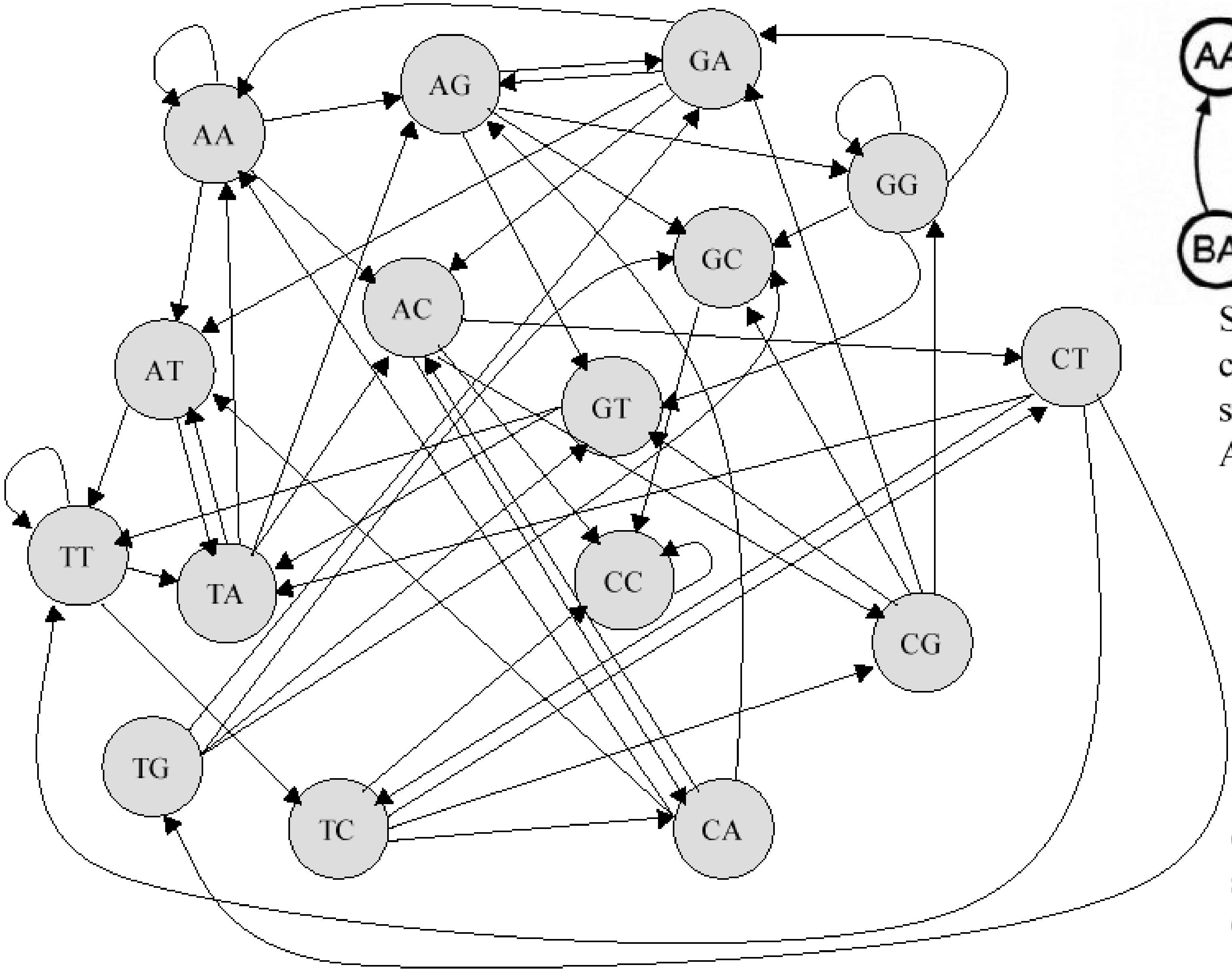
$$x = \{ \mathrm{A,C,G,G,C,C,A,G,T,A,C,C,G,G} \}$$

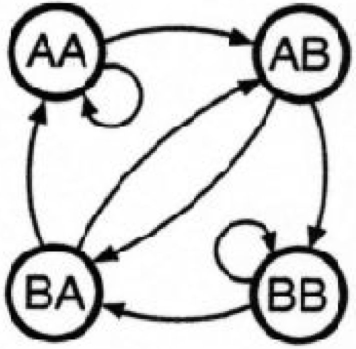
we will work with 2-position states, i.e.

$$x = \{\mathrm{(A,C),(C,G),(G,G),(G,C),(C,C),(C,A),(A,G),} \\ \mathrm{(G,T),(T,A),(A,C),(C,C),(C,G),(G,G)}\}$$

The Second-Order Markov Chain over the 4 elements {A,C,G,T} is equivalent to a First-Order Markov Chain over the 16 two-position states (AA),(AG),(AC),(AT),(GA),(GG),(GC),(GT), etc.

Second -Order Markov Chain for DNA





Second-order chain with two states only, i.e. A and B

Second-order chain with four states , i.e. A, G, C and T

Hidden Markov Models

See notes

Reading Material

1. "Biological Sequence Analysis" by R. Durbin, S.R. Eddy, A. Krogh and G. Mitchison,
Cambridge University Press (1998)
– Chapter 3 : Markov Chains and Hidden Markov Models
– Chapters 4, 5, 8, 10: Applications of Markov Chains and HMMs

2. " Bioinformatics: The Machine Learning Approach" by P. Baldi and S. Brunak, MIT Press (1999)
– Chapters 5 and 6: Theory and Applications of Neural Networks

Questions About a Single Sequence

- Does this sequence belong to a particular family?
  – A family of proteins
  – A branch of a phylogenetic tree

- Assuming that the sequence does come from a particular family, what can we say about its internal structure?
  – Identify the alpha helix or beta sheet regions in a protein
  – Identify regions with promoters
  – Internal structure of the coding (exons) and non-coding regions (introns)
  – Transition from exons to introns and back to exons (splicing sites)

Example: *The CpG Islands (Human Genome)*

- Cytocine is typically methylated in a dinucleotide, C**p**G

- High chance that the methylated-C mutates into a T:

  C**p**G dinucleotides are *rearer* in the genome than the independent probabilities of C and G would imply

- Methylation is suppressed around "start" or"promoters"

– Many more CpG dinucleotides in such regions

– CpG Islands. A few hundred to a few thousand bases long

- Two Questions (with generic value):

– Given a short stretch of genomic sequence, how could we decide if it comes from a CpG Island or not?

– Given a long stretch of DNA how can we find the CpG Islands in it, if there are any?

First Question:
Does a Short DNA Stretch Come from a CpG Island?

Approach: Construct a Model of CpG Islands

- Collect a database of 60,000 nucleotides
- Extract 48 putative CpG Islands
- For the putative CpG Regions compute the transition probabilities from nucleotide *s* to nucleotide *t*

$$a^{+}_{st} = c^{+}_{st} / \Sigma_{t'} c^{+}_{st'}$$

c^{+}_{st} *is the number of times that s is followed by t*

- Similarly for the regions withpout CpG Islands

$$a^{-}_{st} = c^{-}_{st} / \Sigma_{t'} c^{-}_{st'}$$

- Construct table of transition probabilities

First Question: ***Does a Short DNA Stretch Come from a CpG Island?***

Table of Transition Probabilities for CpG Islands

Model	A	C	G	T	
+					
A	.180	.274	.426	.120	= 1
C	.171	.368	.274	.188	= 1
G	.161	.339	.375	.125	= 1
T	.079	.355	.384	.182	= 1

Table of Transition Probabilities for Regions with no CpG Islands

Model	A	C	G	T
-				
A	.300	.205	.285	.210
C	.322	.298	.078	.302
G	.248	.246	.298	.208
T	.177	.239	.292	.292

Calculate the Log-Odds ratio for a chain *x:*

$$S(x)=\log_2 \{[P(x/model+]/[P(x/model-]\} = \Sigma_i \log_2 \{a^{+}_{x(i-1)x(i)}/ a^{-}_{x(i-1)x(i)}\} =$$

$$= \Sigma_i \log_2 \beta_{x(i-1)x(i)}$$

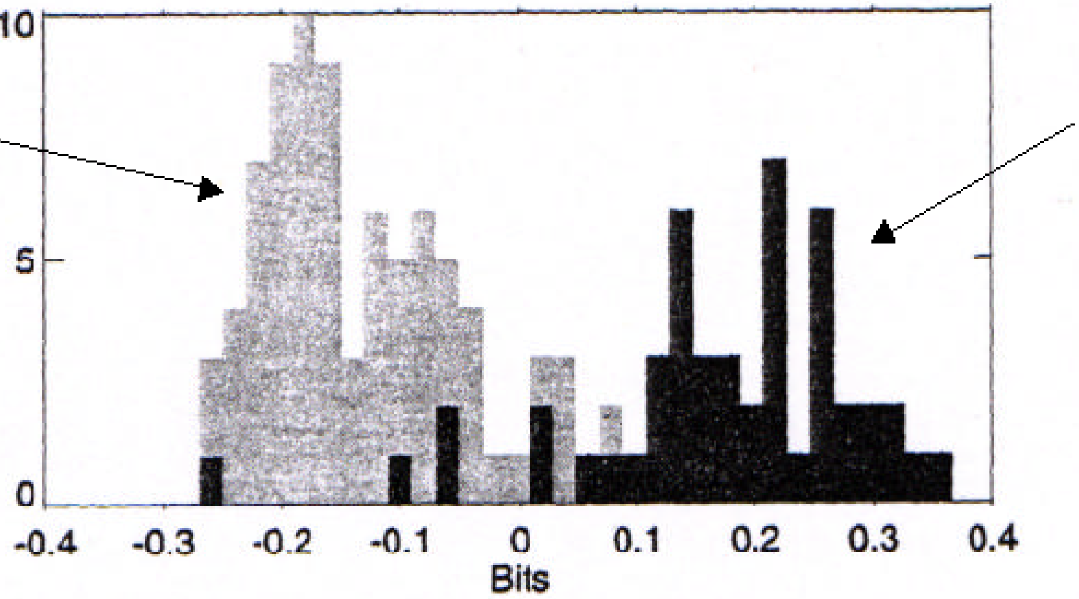
Scores S(x) allow discrimination of a model (+) against another (-)

First Question: *Does a Short DNA Stretch Come from a CpG Island?*

Likelihood Ratios

β	A	C	G	T
A	-0.740	0.419	0.580	-0.803
C	-0.913	0.302	1.812	-0.685
G	-0.624	0.461	0.331	-0.730
T	-1.169	0.573	0.393	-0.679

Test a Given Stretch of DNA



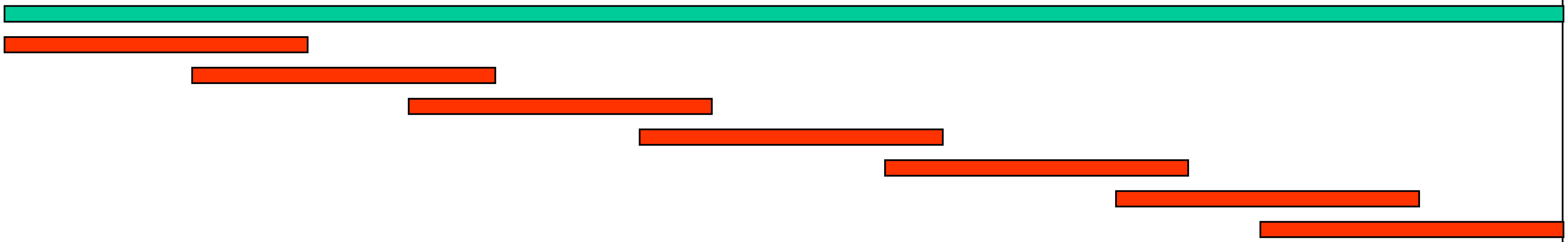
Regions free of CpG Islands

CpG Islands

Second Question: *Given a Long Stretch of DNA Find the CpG Islands in It*

A. First Approach

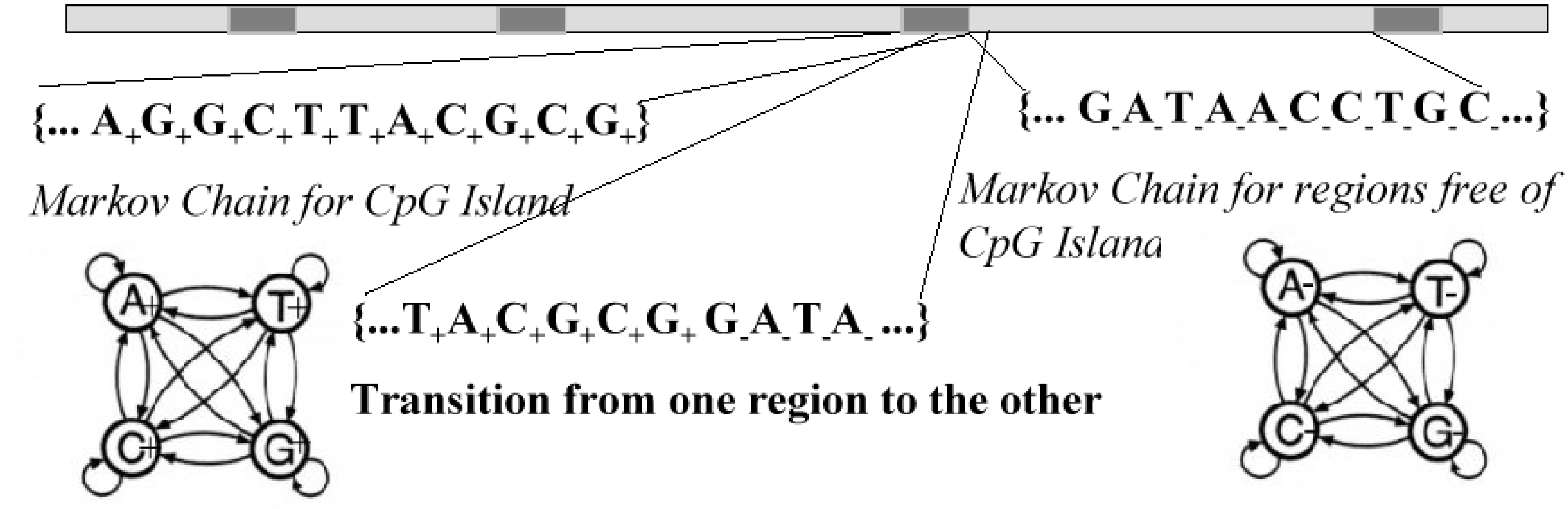
- Build the two First-Order Markov chains for the two regions, as before.
- Take windows of the DNA segment, e.g. 100 nucleotides long



- Compute the log-odds for a window and check against the two Markov models. May need to change the length of the window
- Determine the regions with CpG Islands

Second Question: *Given a Long Stretch of DNA Find the CpG Islands in It.*

B. **Second Approach**: Integrate the Two Markov Models into One



$\{\dots A_+G_+G_+C_+T_+T_+A_+C_+G_+C_+G_+\}$

$\{\dots G_-A_-T_-A_-A_-C_-C_-T_-G_-C_- \dots\}$

Markov Chain for CpG Island

Markov Chain for regions free of CpG Islana

$\{\dots T_+A_+C_+G_+C_+G_+\ G_-A_-T_-A_- \dots\}$

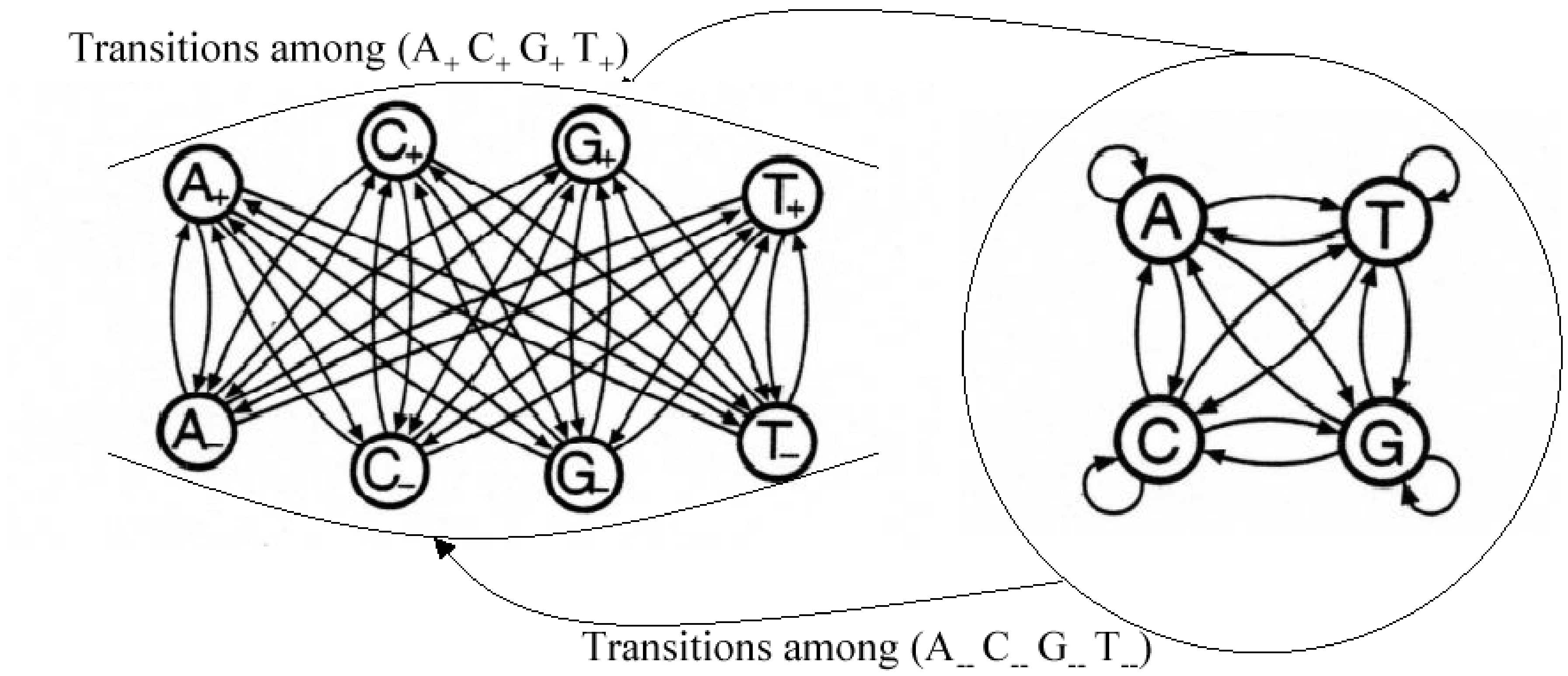
Transition from one region to the other

- *Need probabilities of transition from a CpG-Island Region to a non-CpG Islands region and vice versa.*

- *Each nucleotide can represent two different states*

Second Question: *Given a Long Stretch of DNA Find the CpG Islands in It*

B. Second Approach: Integrate the Two Markov Models into One(2)



Transitions among $(A_+ C_+ G_+ T_+)$

Transitions among $(A_{--} C_{--} G_{--} T_{--})$

- Resulting Model is called..........*Hidden Markov Model (HMM)*
- *No longer possible to tell if a symbol C was emitted by state C_+ or C_-*

The Hidden Markov Model

Distinguish the sequence of states from the sequence of symbols

Path π **:** The state sequence (specified, + or -, state of every nucleotide). $\boldsymbol{\pi}_i$ is the ith state in the path.

$$\mathbf{\{\ldots\ A_+\ G_+\ G_+\ C_+\ A_-\ T_-\ C_-\ C_-\ T\text{-}\ C_-\ A_-\ A_-\ G_-\ T_-\ C_-}$$
$$\mathbf{T_+\ G_+\ A_+\ C_+\ G_+\ C_+\ G_+\ A_-\ G_-\ G_-\ C_-\ T_-\ T_-\ A_-\ C_-\ \ldots\}}$$

- *The states in the path follow a simple Markov Chain*
- ***Transition Probabilities:*** $a_{kl} = P\,(\boldsymbol{\pi}_i = l \mid \boldsymbol{\pi}_{i-1} = k\,)$

Emissions **:** The sequence of symbols (nucleotides of unspecified state, + or -):

{... AGGCATCCTA AGTCTGACGCGAGGCTTAC ...}

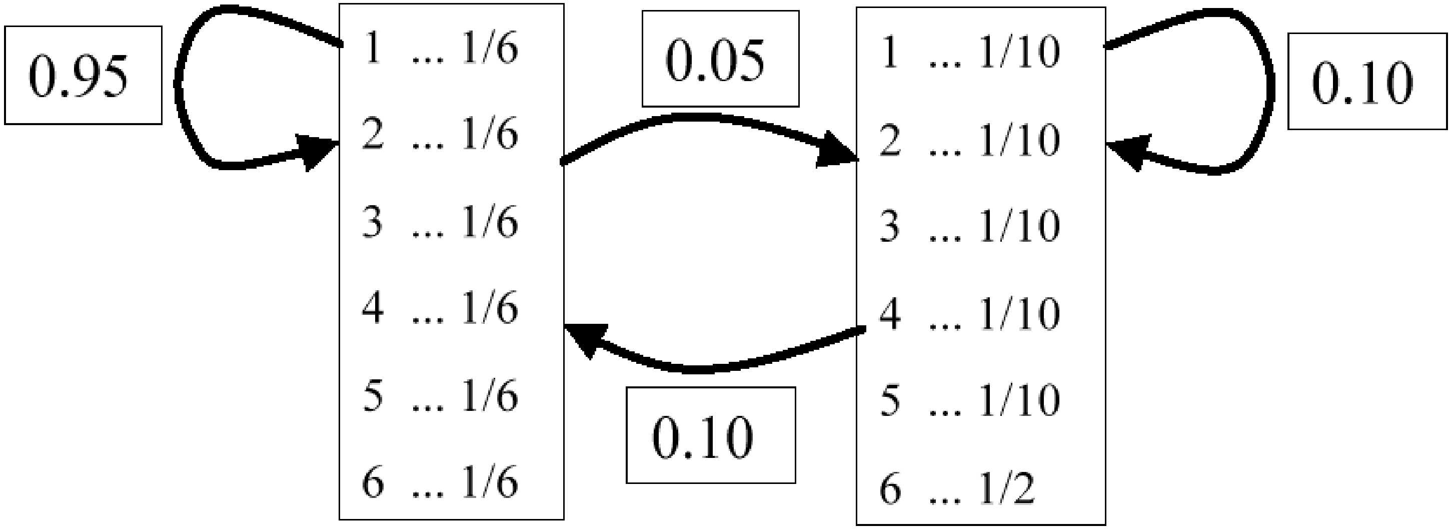
- *States and Symbols are decoupled*
- ***Emission Probability:*** *Probability of emitted symbol, b*

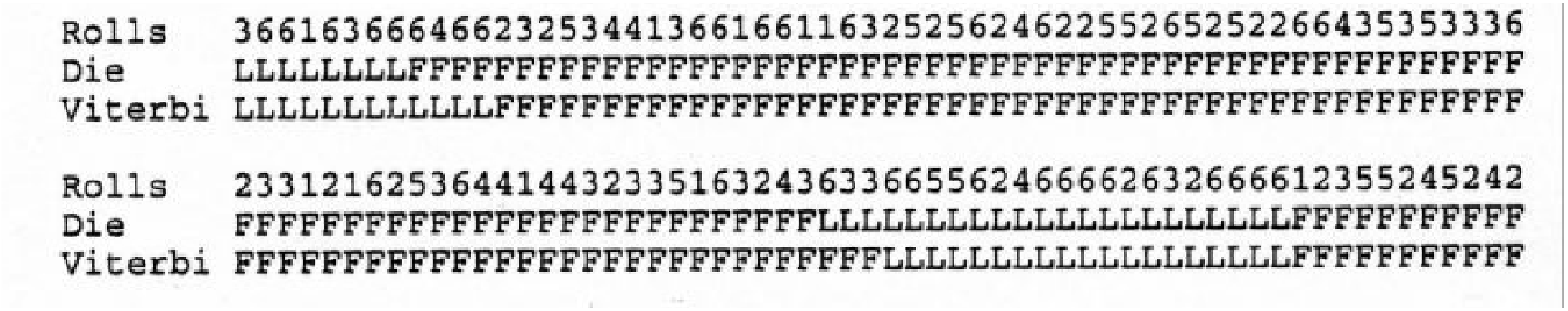
  $e_k\,(b) = P\,(x_i = b \mid \boldsymbol{\pi}_i = k\,)$ *(=0 or 1 for the CpG island problem)*

The Hidden Markov Model

Example: The Occasionally Dishonest Casino.

*The Casino uses two dice, a well-balanced, **fair**, and an unbalanced, **loaded**, one, with the following probabilities:*





```
Rolls   366163666466232534413661661163252562462255265252266435353336
Die     LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   233121625364414432335163243633665562466662632666612355245242
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
```

Think of HMM as *generative* models that generate or emit sequences:

Example, Casino:

Generate random sequences of rolls by

- Simulating the successive choices of die (hidden Markov decision)
- Rolling the chosen die (known probability)

More generally:

- Choose an initial state $\boldsymbol{\pi_1}$ according to probability $a_{0\pi(1)}$
- Emit observation according to distribution $e_k\ (b)=P(x_1=b|\boldsymbol{\pi_1} = k\ )$ for that state
- Then a new state $\boldsymbol{\pi_2}$ is chosen according to the transition probability $a_{\pi(1)i}$ *(+ to +, + to -, - to +, - to -)*

The above processes generate sequences of random observations in which an overt process ($a_{\pi(1)i}$) is combined with a hidden one (+ or -)

The Hidden Markov Model

Path π **: {... A_+ G_+ G_+ C_+ A_- T_- C_- C_- T- C_- A_- A_- G_- T_- C_-**
T_+ G_+ A_+ C_+ G_+ C_+ G_+ A_- G_- G_- C_- T_- T_- A_- C_- ...}

- ***Transition Probabilities:*** $a_{kl} = P(\boldsymbol{\pi}_i = l \mid \boldsymbol{\pi}_{i-1} = k)$

Emissions **: {... AGGCATCCTA AGTCTGACGCGAGGCTTAC..}**

- ***Emission Probability:*** *Probability of emitted symbol, b*

$$e_k(b) = P(x_i = b \mid \boldsymbol{\pi}_i = k)$$

Joint Probability of an observed sequence of symbols, x, and a state sequence, π: $\mathbf{P(x, \pi)} = a_{0\pi(1)} \Pi_i \, e_{\pi(i)}(x_i) \, a_{\pi(i)\pi(i+1)}$

Example: Sequence of Emissions (Symbols).... **CGCG**
State Sequence (Path)......... **C_+ G_- C_- G_+**

Joint Probability $= (a_{0,C+})*1*(a_{C+,G-})*1*(a_{G-,C-})*1*(a_{C-,G+})*1*(a_{C+,0})$

Problem: *Given a Long Stretch of DNA Find the CpG Islands in It*

Given : A sequence of nucleotides, e.g. CGCG

The sequence of symbols {CGCG } can be generated

from any of the following paths: $\{C_+G_+C_+G_+\}$ $\{C_-G_-C_-G_-\}$ $\{C_+G_-C_+G_-\}$

$\{C_-G_+C_-G_+\}$ $\{C_-G_+C_+G_-\}$ $\{C_+G_-C_-G_+\}$

with very different probabilities.

Find : The sequence of the underlying states, i.e. The Path

Solution : From the set of all possible state sequences, which can produce the sequence of the observed symbols, select the one which

"Maximizes the joint probability of the given sequence of symbols, x, and associated sequence of states (Path), $\boldsymbol{\pi}$, i.e.

The Most Probable Path = $\boldsymbol{\pi}$ * = argMax $P\ (x,\ \boldsymbol{\pi})$

The Viterbi Algorithm

Point-1: Let the probability, $v_k(i)$, of the *most probable path* ending in state *k* with observation *i* be known, for all states, *k*.

Point-2: The probability, $v_l\,(i+1)$, of state *l* , after the observation, *i+1*, is made, can be calculated by the equation,

$$v_l(i+1) = e_1(x_{i+1})\ Max_k\{v_k(i)a_{kl}\}$$

Step-1: Initialize (*i=0*): $\boldsymbol{v_0\,(0) = 1,\quad v_k(0) = 0\ \ for\ k>0}$

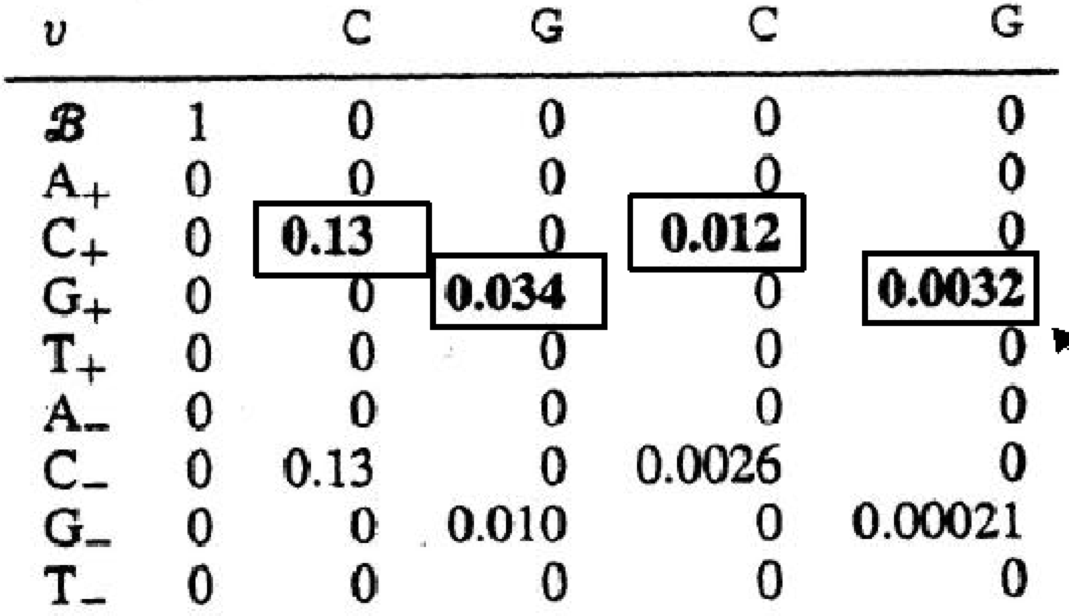
Steps-1-L: $\boldsymbol{v_l(i)=e_l(x_i)\ max_k\,\{v_k(i\text{-}1)\ a_{kl}\,\}}$

Notes:

- The Viterbi algorithm employs the strategy of Dynamic Programming
- Probabilities should be expressed in a log space to avoid underflow errors

Examples of the Viterbi Algorithm

A. The CGCG Region and the CpG Islands:

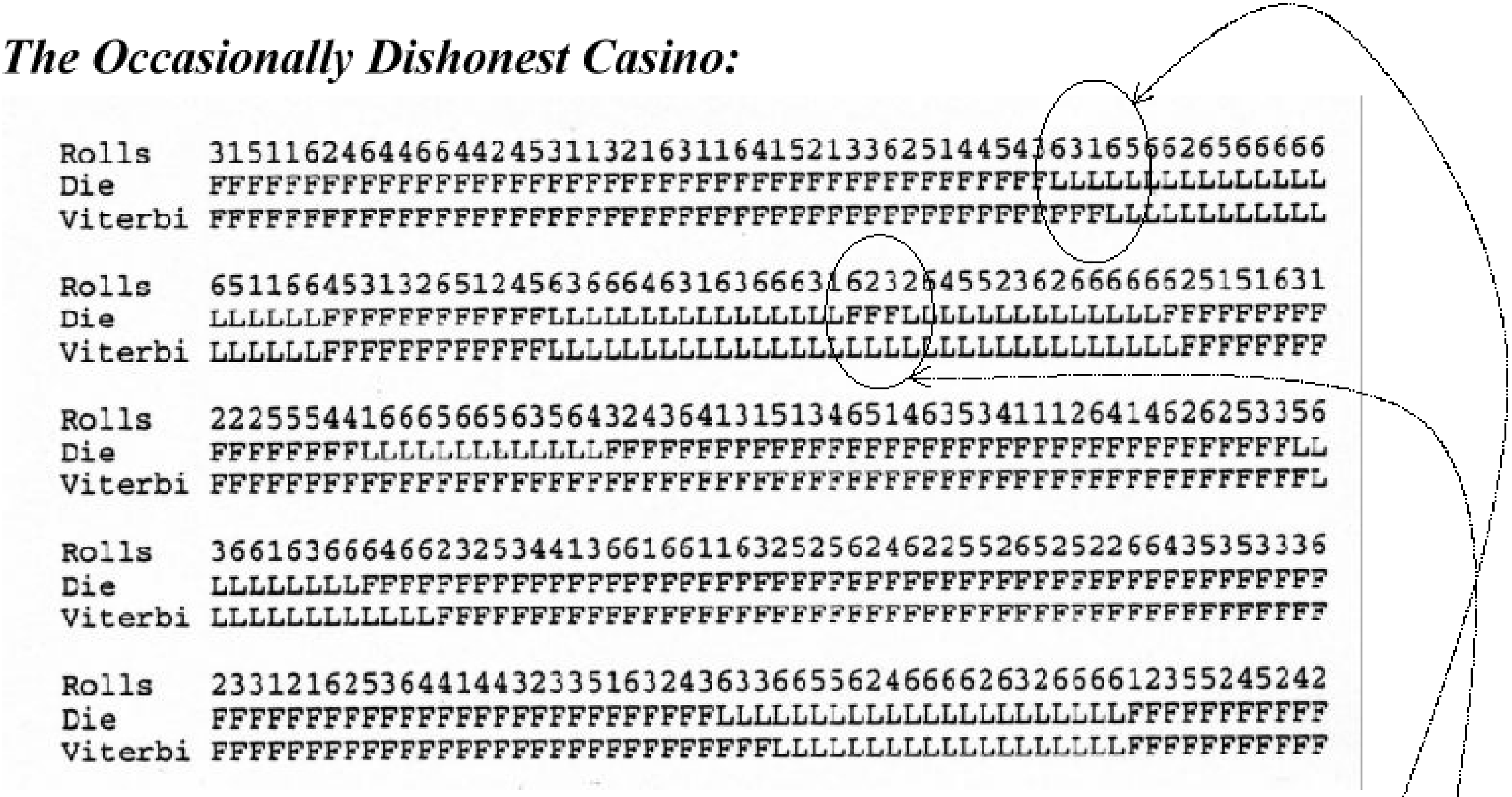


v		C	G	C	G
$\mathcal{B}$	1	0	0	0	0
A_+	0	0	0	0	0
C_+	0	**0.13**	0	**0.012**	0
G_+	0	0	**0.034**	0	**0.0032**
T_+	0	0	0	0	0
A_-	0	0	0	0	0
C_-	0	0.13	0	0.0026	0
G_-	0	0	0.010	0	0.00021
T_-	0	0	0	0	0

The Most Probable Path: $\{C_+G_+C_+G_+\}$

Examples of the Viterbi Algorithm

B. The Occasionally Dishonest Casino:



```
Rolls   315116246446644245311321631164152133625144543631656626566666
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLL

Rolls   651166453132651245636664631636663162326455236266666625151631
Die     LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFFFF
Viterbi LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF

Rolls   222555441666566563564324364131513465146353411126414626253356
Die     FFFFFFFFLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls   366163666466232534413661661163252562462255265252266435353336
Die     LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   233121625364414432335163243633665562466662632666612355245242
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
```

Notes:

- *Delayed transitions*
- *Failure when occasional violation of underlying statistics occurs*

Notes on HMMs and the Viterbi Algorithm

- **Probability of a sequence of symbols, *x*:** $P(x) = \Sigma P(x, \pi)$
- What is the probability that observation, x_i , came from state, *k*, given the observed sequence (Posterior State Probability), i.e.

$$P\ (\boldsymbol{\pi}_i = k \mid x\ )$$

- **Estimation of parameters for HMMs**

– When the State Sequences (Paths) are known. Count the number of transitions and emissions in a given set of known sequences, i.e.

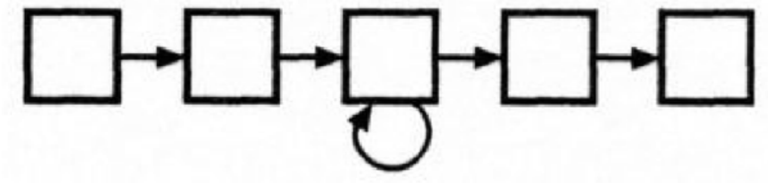
- ***Transition Probabilities*** $a_{kl} = A_{kl} / \Sigma_{l'} A_{kl'}$
- ***Emission Probabilities*** $e_k\ (b) = E_k\ (b) / \Sigma_{\ b'} E_k\ (b')$

– When the state sequences (Paths) are unknown: Baum-Welch and Viterbi Training

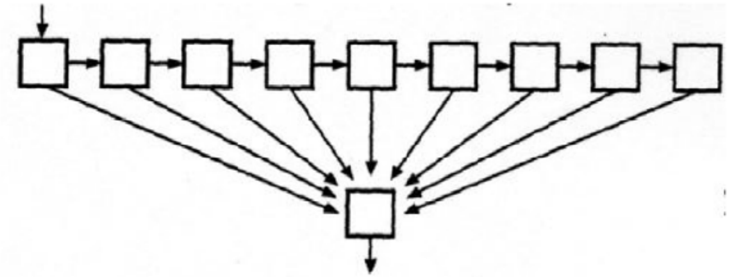
Structure of HMMs

- **Length of extent of model**

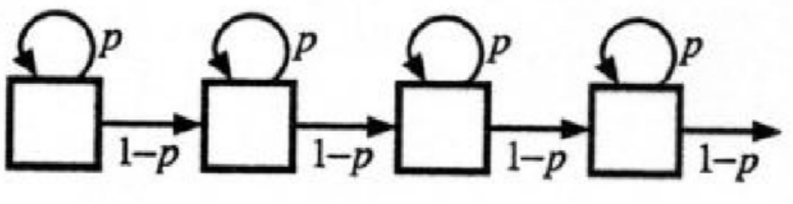
  - Exponentially decaying: $P\ (k\ residues) = (1\text{-}p)p^{k\text{-}1}$



  - Defined range of length, e.g., Model with distribution of lengths between 2 and 10



  - Non-geometric length distribution, e.g., array of n states


p
p
p
p
1−p
1−p
1−p
1−p