

# MA3259 Lecture 19

## **Hidden Markov Models and Applications III: Three Algorithms**

LX Zhang  
Department of Mathematics  
National University of Singapore  
matzlx@nus.edu.sg

# Learning Algorithm

---

**Input:** For a sequence  $O: o_1, o_2, \dots, o_t$ ,

**Output:** find the parameters  $\lambda=(P, B, \pi)$  that maximizes  $\Pr[O | \lambda]$ .

(We assume the topology of the model is fixed.)

# Two Cases

- How to generate the observed sequence is also known

**Examples:**

**GIVEN:** A sequence of 2,000 rolls. In addition, how the casino player changes dice and produces these 2,000 rolls is recorded by video camera.

- How to generate the observed sequence is unknown

**Examples:**

**GIVEN:** Gambler observed 2,000 rolls of the casino player in one night, but he could not detect when he changes dice

# Algorithm in Case 1.

Input: An observed sequence  $O: o_1, o_2, \dots, o_t$  and the corresponding parse  $\mathcal{Q}: q_1, q_2, \dots, q_t$ .

Set:

$$\begin{aligned} A_i &= \# \text{ times state } S_i \text{ in } \mathcal{Q} \\ A_{ij} &= \# \text{ times transition } S_i \rightarrow S_j \text{ occurs in } \mathcal{Q} \\ B_i(o) &= \# \text{ times state } S_i \text{ in } \mathcal{Q} \text{ emits } o \text{ in } O \end{aligned}$$

The parameter  $\lambda=(P, B, \pi)$  is estimated as

$$P=(p_{ij}): \quad p_{ij} = \frac{A_{ij}}{\sum_k A_{ik}} \qquad \pi=(\pi_i): \quad \pi_i = \frac{A_i}{\sum_j A_j}$$

---

$$B=(b_i): \quad b_i(o_j) = \frac{B_i(o_j)}{\sum_k B_i(o_k)}$$

# Casino Example: A sequence of 300 rolls and the corresponding generating parse.

Rolls 315116246446644245311321631164152133625144543631656626566666  
 Die FFLLLLLLLLLLLLLLLLL

Rolls 651166453132651245636664631636663162326455236266666625151631  
 Die LLLLLLFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFF

Rolls 222555441666566563564324364131513465146353411126414626253356  
 Die FFFFFFFLLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls 366163666466232534413661661163252562462255265252266435353336  
 Die LLLLLLFF

Rolls 233121625364414432335163243633665562466662632666612355245242  
 Die FFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLFFFFFFFFFFFF

$$A_{FF} = 209 \quad A_{FL} = 6$$

$$A_{LF} = 6 \quad A_{LL} = 78$$

$$A_F = 216 \quad A_L = 84$$

$$B_F(1)=36$$

$$B_F(2)=41$$

$$B_F(3)=36$$

$$B_F(4)=33$$

$$B_F(5)=30$$

$$B_F(6)=34$$

$$B_L(1)=6$$

$$B_L(2)=8$$

$$B_L(3)=12$$

$$B_L(4)=3$$

$$B_L(5)=11$$

$$B_L(6)=44$$

**Casino Example:** A sequence of 300 rolls and the corresponding generating parse.

$$A_{FF} = 209 \quad A_{FL} = 6$$

$$A_{LF} = 6 \quad A_{LL} = 78$$

$$A_F = 216 \quad A_L = 84$$

$$p_{FF} = \frac{A_{FF}}{A_{FF} + A_{FL}} = \frac{209}{215} = 0.972$$

$$p_{FL} = \frac{A_{FL}}{A_{FF} + A_{FL}} = \frac{6}{215} = 0.038$$

$$p_{LF} = \frac{A_{LF}}{A_{LF} + A_{LL}} = \frac{6}{84} = 0.071$$

$$p_{LL} = \frac{A_{LL}}{A_{LF} + A_{LL}} = \frac{78}{84} = 0.929$$

$$B_F(1)=36$$

$$B_F(2)=41$$

$$B_F(3)=36$$

$$B_F(4)=33$$

$$B_F(5)=30$$

$$B_F(6)=34$$

$$B_L(1)=6$$

$$B_L(2)=8$$

$$B_L(3)=12$$

$$B_L(4)=3$$

$$B_L(5)=11$$

$$B_L(6)=44$$

$$b_F(1) = \frac{B_F(1)}{\sum_j B_F(j)} = \frac{36}{210} = 0.171$$

$$b_F(2) = \frac{B_F(2)}{\sum_j B_F(j)} = \frac{41}{210} = 0.195$$

$$b_F(3) = \frac{B_F(3)}{\sum_j B_F(j)} = \frac{36}{210} = 0.171$$

$$b_F(4) = \frac{B_F(4)}{\sum_j B_F(j)} = \frac{33}{210} = 0.157$$

$$b_F(5) = \frac{B_F(5)}{\sum_j B_F(j)} = \frac{30}{210} = 0.142$$

$$b_F(6) = \frac{B_F(6)}{\sum_j B_F(j)} = \frac{34}{210} = 0.162$$

**Rationale:** When the underlying states are known, max likelihood estimate is the normalized frequency of transitions and emissions that occur in the training data.

**Drawback:**

Given little data, there may be **overfitting:**

In other words,  $P(x|\theta)$  is maximized, but  $\theta$  is unreasonable  
**0 probabilities – BAD**

**Example:**

Given 10 casino rolls, we observe

O: 2, 6, 5, 6, 1, 6, 3, 6, 2, 3  
Q: L, L, L, L, L, L, L, L, L, L

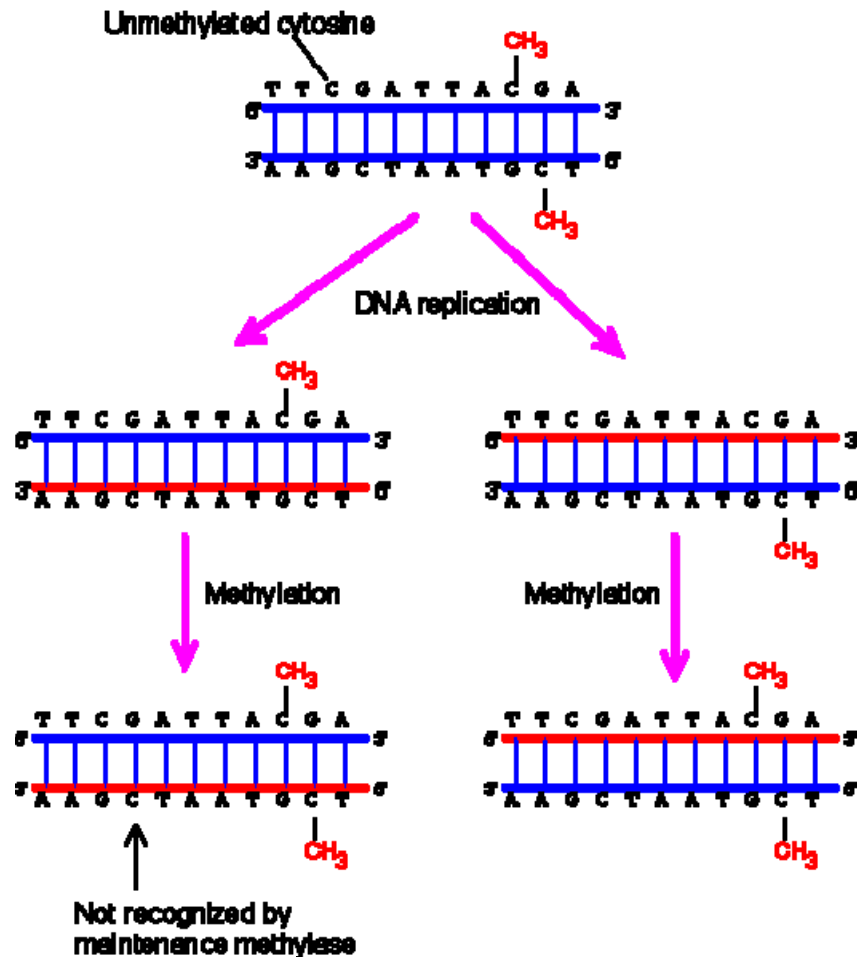
# Modified Estimation

For small training sets, we add small pseudocount terms to avoid 0 probabilities.

$$\begin{aligned} A_i &= \# \text{ times state } S_i \text{ in } \mathcal{Q} + \sum_j a_{ij} \\ A_{ij} &= \# \text{ times transition } S_i \rightarrow S_j \text{ occurs in } \mathcal{Q} + a_{ij} \\ B_i(o) &= \# \text{ times state in } \mathcal{Q} \text{ emits } o \text{ in } \mathcal{O} + b_i(o) \end{aligned}$$

Where  $a_{ij}$  and  $b_i(o)$  are pseudocounts representing our prior belief.

# Example: CpG Islands (Batzoglou's slide)



- One way cells differentiate is methylation
  - Addition of CH<sub>3</sub> in C-nucleotides
  - Silences genes in region
- CG (denoted CpG) often mutates to TG, when methylated
- In each cell, one copy of X is silenced, methylation plays role
- Methylation is inherited during cell division

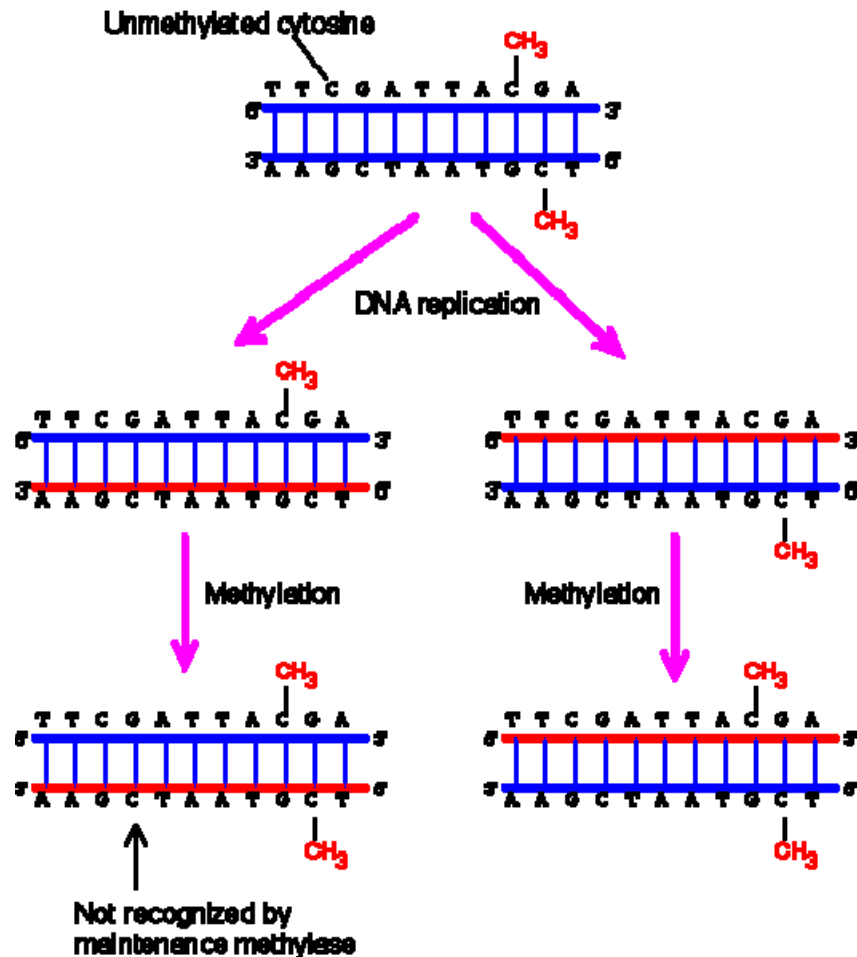
# Cellular differentiation

A process by which a less specialized cell becomes a more specialized cell type. Differentiation occurs during the development of a multicellular organism.

It also occurs in adult. Adult stem cells divide and create fully-differentiated daughter cells during tissue repair and during normal cell turnover.

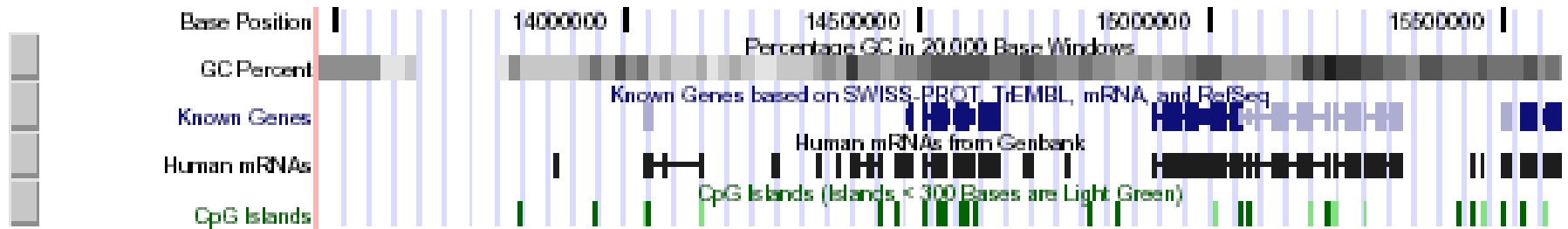
Differentiation dramatically changes a cell's size, shape, membrane potential, metabolic activity, and responsiveness to signals

# Example: CpG Islands (Batzoglou's slide)



- One way cells differentiate is methylation
  - Addition of CH<sub>3</sub> in C-nucleotides
  - Silences genes in region
- CG (denoted CpG) often mutates to TG, when methylated
- In each cell, one copy of X is silenced, methylation plays role
- Methylation is inherited during cell division

# Example: CpG Islands (Batzoglou's slide)



CpG nucleotides in the genome are frequently methylated

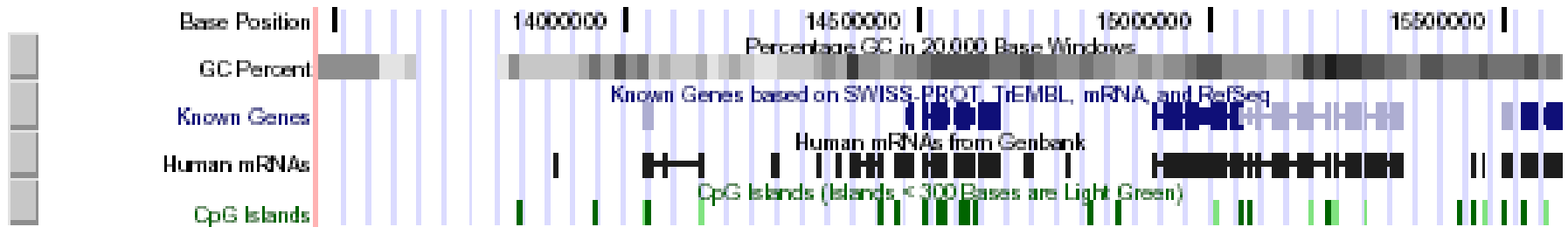
(Write CpG not to confuse with CG base pair)

$C \rightarrow \text{methyl-C} \rightarrow T$

Methylation often suppressed around genes, promoters

$\rightarrow$  CpG islands

# Example: CpG Islands (Batzoglou's slide)



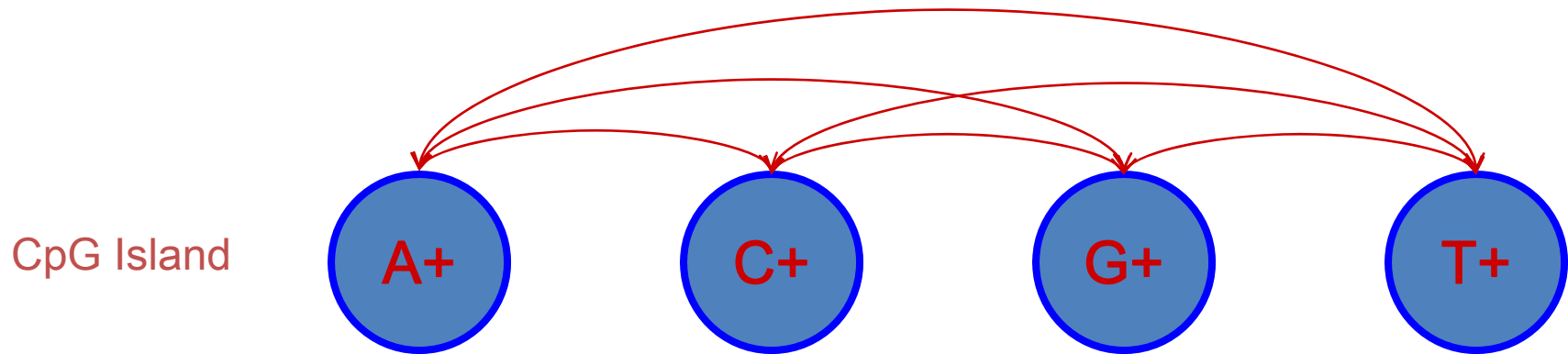
- In CpG islands,
  - CG is more frequent
  - Other pairs (AA, AG, AT...) have different frequencies

**Question:** Detect CpG islands computationally

# A model of CpG Islands – (1)

## Architecture

(Batzoglou's slide)



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

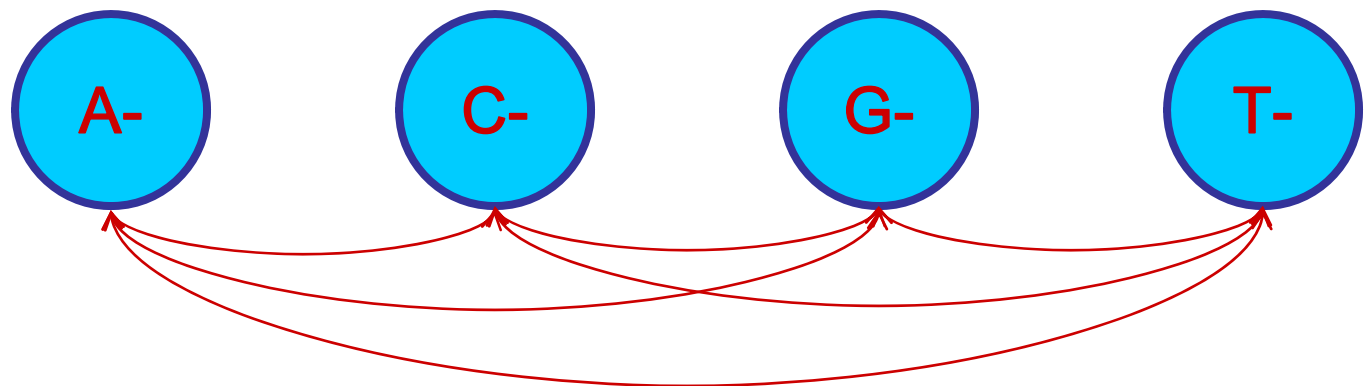
# A model of CpG Islands – (1)

## Architecture

(Batzoglou's slide)

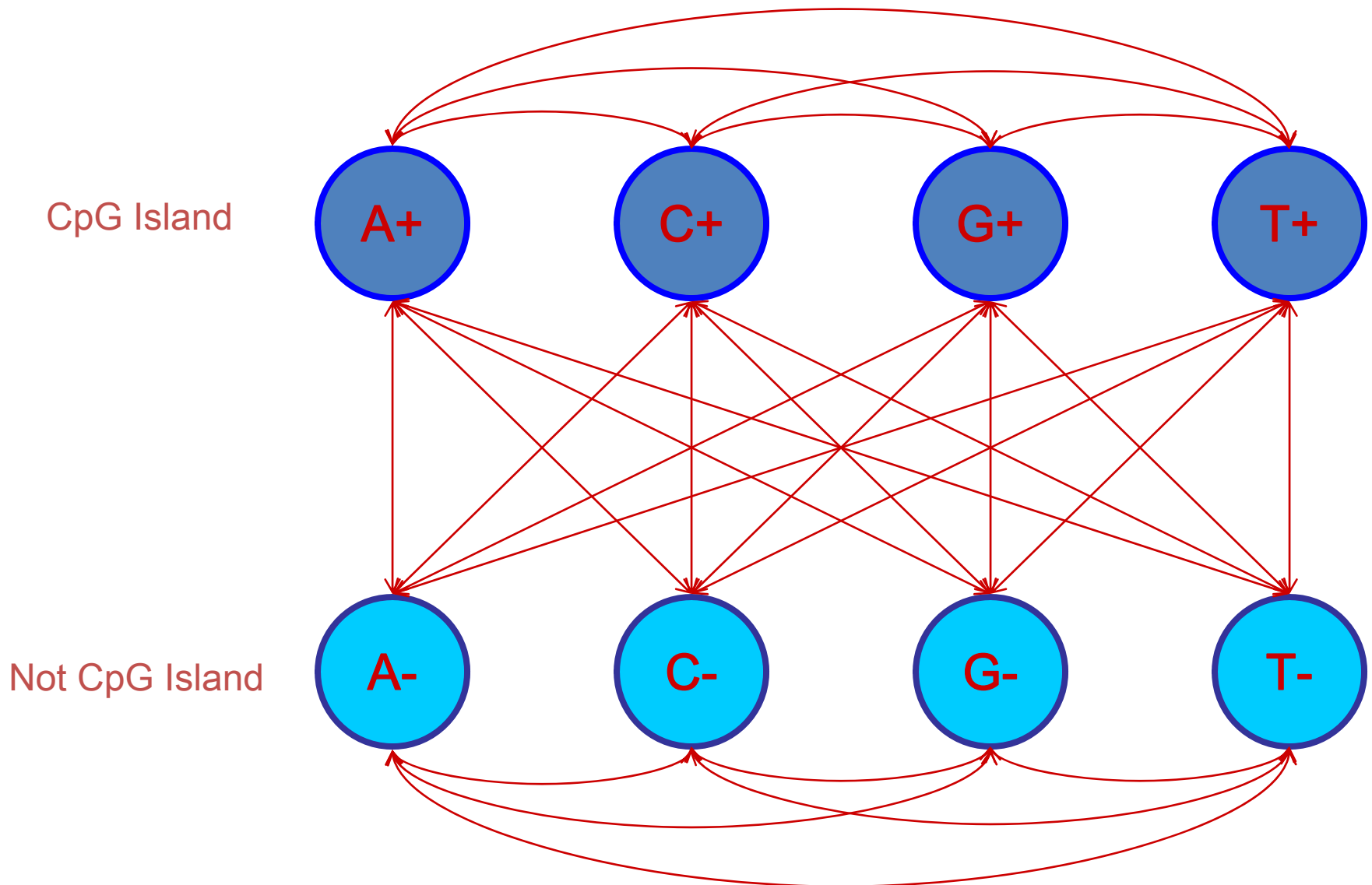
-	A	C	G	T
A	.300	.205	.285	.210
C	.233	.298	.078	.302
G	.248	.246	.298	.208
T	.177	.239	.292	.292

Not CpG Island



# A model of CpG Islands – (2)

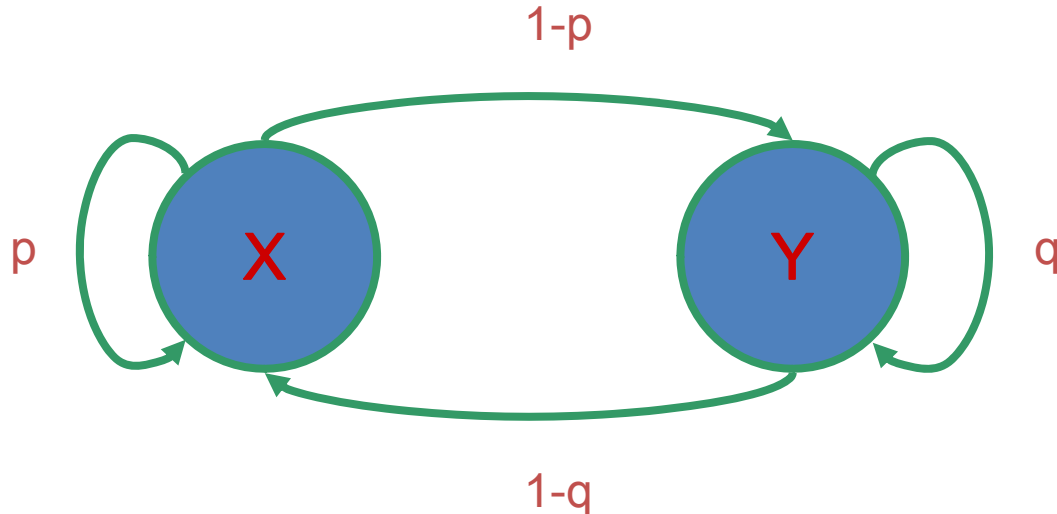
## Architecture (Batzoglou's slide)



# A model of CpG islands – (2)

## Transitions between + and – states (Batzoglou's slide)

- What about transitions between (+) and (-) states?
- They are computed from
  - Avg. length of CpG island
  - Avg. separation between two CpG islands



Length distribution of region X:

$$P[l_X = 1] = 1-p$$

$$P[l_X = 2] = p(1-p)$$

...

$$P[l_X = k] = p^{k-1}(1-p)$$

$$E[l_X] = 1/(1-p)$$

Geometric distribution, with mean  
 $1/(1-p)$

# Algorithm in Case 2: The Baum-Welch Algorithm

---

We are not able to find the true frequencies  $A_i, A_{ij}, B_i(o)$  in this case.

What we will do:

- Estimate  $A_i, A_{ij}, B_j(o)$  using our best guess of the model.
- Update the parameters of the model, based on our guess
- Repeat the above two steps until the generating probability of the observed sequence do not change much.

In a loop step,

- We have the given sequence  $O: o_1, o_2, \dots, o_t$
  - But we do not know the generating parse.
  - However, we have a model (whose parameter set is obtained in the last loop step), called **the current model**.
1. Estimate  $A_i, A_{ij}, B_k(o)$  in the training data;
  2. Update  $\theta$  according to  $A_{ij}, B_j(o)$ .

# How to estimate new parameters?

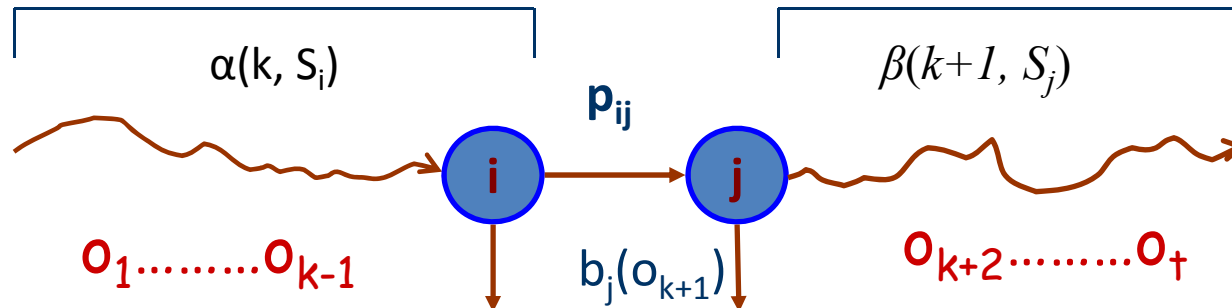
To estimate  $A_{ij}$ , **based on the current model.**

At each time point, find probability that transition  $S_i \rightarrow S_j$  is used:

$$\begin{aligned} & \Pr[Q_k = S_i, Q_{k+1} = S_j \mid O] \\ &= \Pr[Q_k = S_i, Q_{k+1} = S_j, O] / \Pr[O] = W / \Pr[O] \end{aligned}$$

$$\begin{aligned} \text{where } W &= \Pr[o_1 \dots o_k, Q_k = S_i, Q_{k+1} = S_j, o_{k+1} \dots o_t] \\ &= \Pr[o_1 \dots o_k, Q_k = S_i] \Pr[Q_{k+1} = S_j, o_{k+1} \dots o_t \mid Q_k = S_i] \\ &= \alpha(k, S_i) \Pr[Q_{k+1} = S_j, o_{k+1} \dots o_t \mid Q_k = S_i] \\ &= \alpha(k, S_i) \Pr[o_{k+1} \mid Q_{k+1} = S_j] \Pr[Q_{k+1} = j \mid Q_k = i] \Pr[o_{k+2} \dots o_t \mid Q_{k+1} = S_j] \\ &= \alpha(k, S_i) b_j(o_{k+1}) p_{ij} \beta(k+1, S_j) \end{aligned}$$

$$\text{Hence, } \Pr[Q_k = S_i, Q_{k+1} = S_j \mid O] = \alpha(k, S_i) b_j(o_{k+1}) p_{ij} \beta(k+1, S_j) / \Pr[O]$$



# How to estimate new parameters?

Based on the current model.

At each time point  $k$ , the probability that transition  $S_i \rightarrow S_j$  is used is

$$\Pr[Q_k = S_i, Q_{k+1} = S_j | O] = \alpha(k, S_i) b_j(o_{k+1}) p_{ij} \beta(k+1, S_j) / \Pr[O]$$

$N_{ij}$  = the  $E[\# \text{ times transition } S_i \rightarrow S_j, \text{ given current } \theta]$

$$= \sum_{k \leq t-1} \Pr[Q_k = S_i, Q_{k+1} = S_j | O]$$

$N_i$  = the  $E[\# \text{ times state } S_i \text{ given current } \theta]$

$$= \sum_j N_{ij}$$

Similarly,

$N_i(o)$  = the  $E[\# \text{ times state } S_i \text{ in } Q \text{ emits } o \text{ in } O]$

$$= (1 / \Pr[O]) \sum_{\{i \mid o_i=o\}} \alpha(k, S_i) \beta(k, S_i)$$

Use  $N_{ij}$ ,  $N_i$  and  $N_i(o)$  to define the new parameters like in Case 1.

# The Baum-Welch Algorithm

**Input:** An observed sequence  $O: o_1, o_2, \dots, o_t$

**Initialization:**

Pick the best-guess of the model parameters  
(or arbitrary)

**Iteration:**

1. Forward
2. Backward
3. Calculate  $N_{ij}, N_i, N_i(o)$ , given  $\theta_{\text{CURRENT}}$
4. Calculate new model parameters  $\theta_{\text{NEW}} : p_{ij}, b_k(o)$
5. Calculate new log-likelihood  $\Pr[O | \theta_{\text{NEW}}]$

Until  $\Pr[O | \theta_{\text{NEW}}]$  does not change much

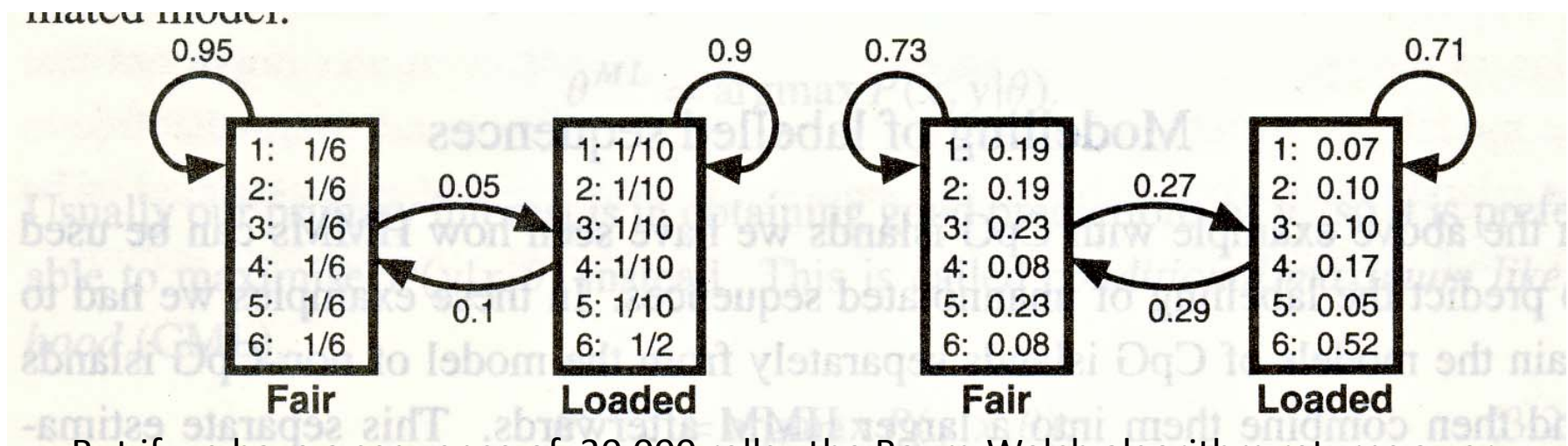
Time Complexity:

# iterations  $\times O(K^2N)$

- Guaranteed to increase the log likelihood  $\Pr[x | \theta]$
- Not guaranteed to find globally best parameters

# Dishonest casino example

Apply the Baum-Welch algorithm to the sequence of 300 rolls, we can obtain the following estimation:



But if we have a sequence of 30,000 rolls, the Baum-Welch algorithm returns a very close model:

