Hidden Markov Models

David Bernick

Why a shirtless dealer?

Two Coins Up the Dealer Sleeve

- Now the dealer has *both* fair and biased coins and can change between them at any time (without you noticing) with probability 0.1.

After watching a sequence of flips, can you tell when the dealer was using fair coin and when he was using biased coin?

"Madam…. How can I be expected to read your mind when you keep changing it?"

Reading the Dealer's Mind

Casino Problem: *Given a sequence of coin flips, determine when the dealer used a fair coin and a biased coin.*

- **Input:** A sequence $x = x_1\, x_2\, \ldots\, x_n$ of flips made by coins *F* (fair) and *B* (biased).
- **Output:** A sequence $\pi = \pi_1\, \pi_2 \cdots \pi_n$, with each π_i being equal to either *F* or *B* and indicating that x_i is the result of flipping the fair or biased coin, respectively.

Can we "determine" the coin in use?

The Problem with the Casino Problem

- Any outcome of coin tosses could have been generated by any combination of fair and biased coins
  - HHHHH could be generated by *BBBBB*, *FFFFF*, *FBFBF*, etc.

We need to ***grade*** different scenarios:

BBBBB, *FFFFF*, *FBFBF*, etc.

differently, depending on how likely they are.

How can we explore and grade 2^n possible scenarios?

Reading the Dealer's Mind (one window at a time)

HHHTHT**HHH**T

BBBBB

$\Pr(x|F) < \Pr(x|B)$

Reading the Dealer's Mind (one window at a time)

HHHTHTHHHT

BBBBB $\Pr(x|F) < \Pr(x|B)$

FFFFF $\Pr(x|F) > \Pr(x|B)$

Reading the Dealer's Mind (one window at a time)

HHHTHTHHHT

BBBBB $\Pr(x|F)/\Pr(x|B) < 1$

FFFFF $\Pr(x|F)/\Pr(x|B) > 1$

Reading the Dealer's Mind (one window at a time)

HHHTHTHHHT

BBBBB $\Pr(x|F)/\Pr(x|B) < 1$

FFFFF $\Pr(x|F)/\Pr(x|B) > 1$

Log-odds ratio of sequence $x = \log_2 \Pr(x|F) / \Pr(x|B)$

$\log_2 (1/2)^n / (3/4)^k \bullet (1/4)^{n-k} = \#\text{Tosses} - \log_2 3 * \#\text{Heads}$

Reading the Dealer's Mind (one window at a time)

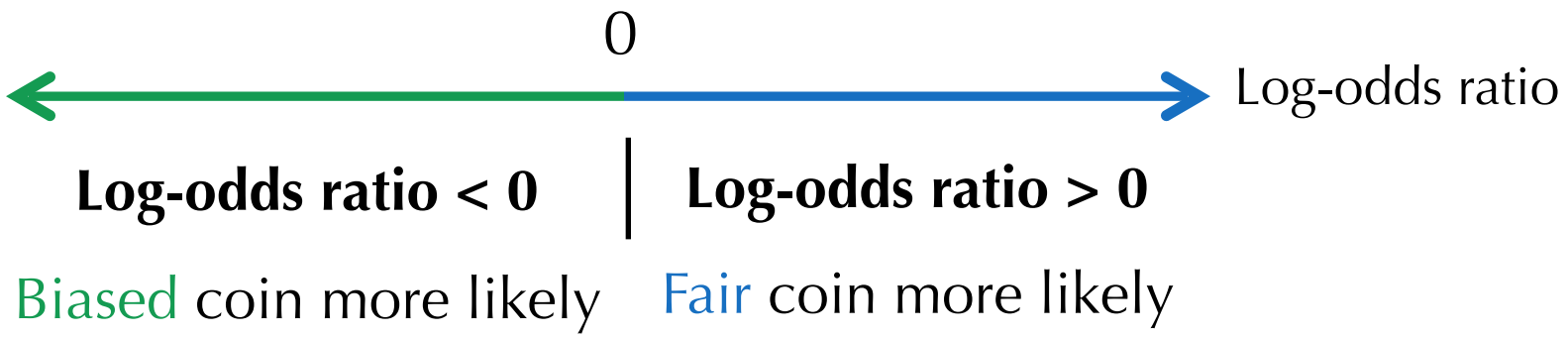
HHHTHTHHHT

BBBBB **Log-odds** < 0

FFFFF **Log-odds** > 0

Log-odds ratio of sequence $x = \log_2 \Pr(x|F) / \Pr(x|B)$

$= \#\text{Tosses} - \log_2 3 * \#\text{Heads}$



Reading the Dealer's Mind

Reading the Dealer's Mind

Reading the Dealer's Mind

Reading the Dealer's Mind

HHHTHTHHHT

BBBBB

FFFFF

FFFFF

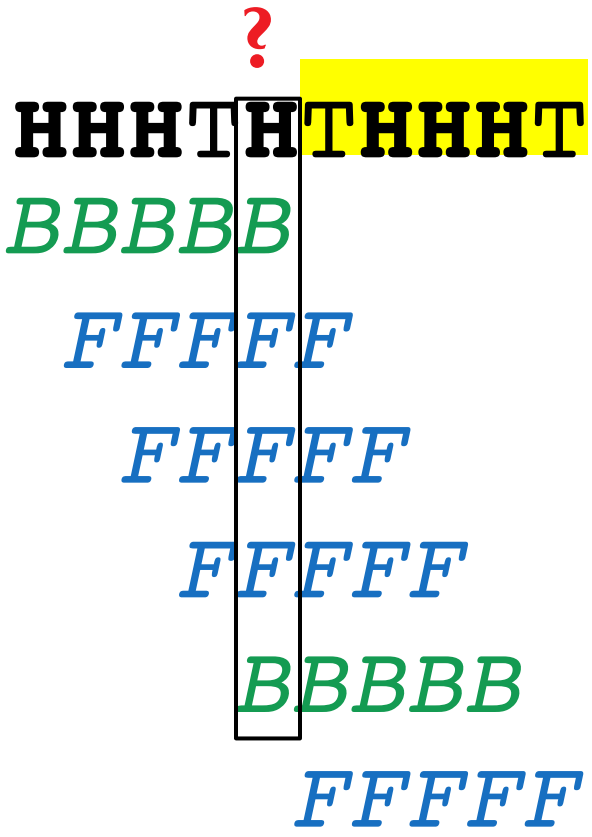
FFFFF

BBBBB

FFFFF

What are the disadvantages of this approach?

Disadvantages of the Sliding Window Approach



Different windows may classify the same coin flip differently!

The results depend on the window length. How to choose it?

Why Are CG Dinucleotides More Rare than GC Dinucleotides in Genomic Sequences?

- Different species have widely varying **GC-content** (percentages of G+C nucleotides in the genome).

46% for gorilla and human

58% for platypus

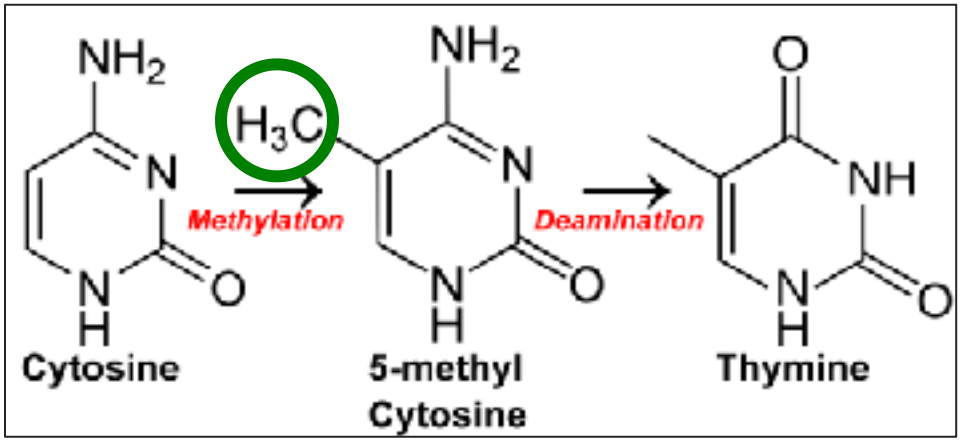
- Each of the dinucleotides CC, CG, GC, and GG is expected to occur in the human genome with frequency 0.23 * 0.23 = 5.29%.

But the frequency of CG in the human genome is only 1%!

Methylation

Methylation: adds a methyl (CH_3) group to the cytosine nucleotide (often within a CG dinucleotide).

- The resulting **methylated cytosine** has the tendency to *deaminate* into thymine.



- As a result of methylation, CG is the least frequent dinucleotide in many genomes.

Looking for CG-islands

Methylation is often suppressed around genes in areas called **CG-islands** (CG appears frequently).

ATTTCTTCTCGTCGACGCTAATTTCTTGGAAATATCATTAT

In a first attempt to find genes, how would you search for CG-islands?

Looking for CG-islands

- Different windows may classify the same position in the genome differently.
- It is not clear how to choose the length of the window for detecting CG-islands.
- Does it make sense to choose the same window length for all regions in the genome?

Hidden Markov Models

- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model

- Decoding Problem
- The Viterbi Algorithm
- Profile HMMs for Sequence Alignment
- Classifying proteins with profile HMMs
- Viterbi Learning
- Soft Decoding Problem
- Baum-Welch Learning

Turning the Dealer into a Machine

- Think of the dealer as a machine with k **hidden states** (F and B) that proceeds in a sequence of steps.
- In each step, it emits a symbol (H or T) while being in one of its hidden states.
- While in a certain state, the machine makes two decisions:
  - Which *symbol* will I emit?
  - Which *hidden state* will I move to next?

Why “Hidden”?

- An observer can see the emitted symbols of an HMM but *does not* know which state the HMM is currently in.

- The goal is to infer the most likely sequence of hidden states of an HMM based on the sequence of emitted symbols.

Hidden Markov Model (HMM)

Σ: an **alphabet** of emitted symbols

H and T

States : a set of **hidden states**

F and *B*

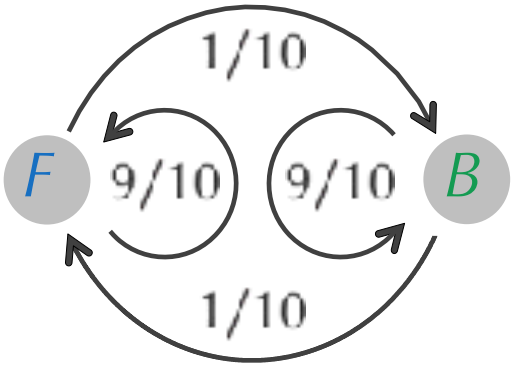
Transition = ($transition_{l,k}$): a |*States*| ×|*States*|
matrix of **transition probabilities**
(of changing from state *l* to state *k*)

	F	*B*
F	0.9	0.1
B	0.1	0.9

Emission= ($emission_k(b)$): a |*States*| × |∑|
matrix of **emission probabilities**
(of emitting symbol *b* when the HMM is in state *k*)

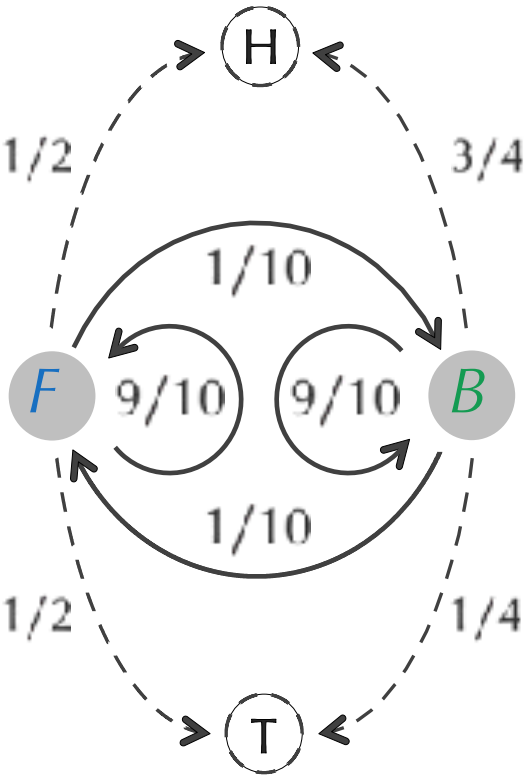
	H	T
F	0.50	0.50
B	0.75	0.25

HMM Diagram



	Transition	
	F	*B*
F	0.9	0.1
B	0.1	0.9

HMM Diagram



Transition

	F	*B*
F	0.9	0.1
B	0.1	0.9

Emission

	H	T
F	0.50	0.50
B	0.75	0.25

Hidden Path

Hidden path: the sequence $\pi = \pi_1 \ldots \pi_n$ of states that the HMM passes through.

- $\Pr(x, \pi)$: the probability that an HMM follows the hidden path π *and* emits the string $x = x_1 x_2 \ldots x_n$.

x:	T	H	T	H	H	H	T	H	T	T	H
π:	F	F	F	B	B	B	B	B	F	F	F

$$\sum_{\text{all possible emitted strings } x} \sum_{\text{all possible hidden paths } \pi} \Pr(x, \pi) = 1$$

- $\Pr(x|\pi)$: the **conditional probability** that an HMM emits the string x *after* following the hidden path π.

$$\sum_{\text{all possible emitted strings } x} \Pr(x|\pi) = 1$$

$\Pr(x, \pi) = \Pr(x|\pi) * \Pr(\pi)$

- $\Pr(x, \pi)$: the probability that an HMM follows the hidden path π and emits the string x.
- $\Pr(x_i|\pi_i)$ – probability that x_i was emitted from the state π_i (equal to $emission_{\pi i}(x_i)$).
- $\Pr(\pi_{i-1} \rightarrow \pi_i)$ – probability that the HMM moved from $\pi_{i-1} \rightarrow \pi_i$ (equal to $transition_{\pi i, \pi i+1}$).

x	T	H	T	H	H	H	T	H	T	T	H
π	F	F	F	B	B	B	B	B	F	F	F
$\Pr(\pi_{i-1} \rightarrow \pi_i)$	.5	.9	.9	.1	.9	.9	.9	.9	.1	.9	.9
$\Pr(x_i|\pi_i)$	½	½	½	¾	¾	¾	¼	¾	½	½	½

$$\Pr(\pi) = \prod_{i=1,n} \Pr(\pi_{i-1} \rightarrow \pi_i) = \prod_{i=1,n} transition_{\pi i-1, \pi i}$$

$$\Pr(x|\pi) = \prod_{i=1,n} \Pr(x_i|\pi_i) = \prod_{i=1,n} emission_{\pi i}(x_i)$$

Computing Probability of a Hidden Path Pr(π)

Probability of a Hidden Path Problem. *Compute the probability of an HMM's hidden path.*

- **Input:** A hidden path π in an HMM (∑, *States*, *Transition*, *Emission*).
- **Output:** The probability of this path, Pr(π).

Computing Probability of a Hidden Path Pr(π) and Conditional Probability of an Outcome Pr(*x*|π)

Probability of a Hidden Path Problem. *Compute the probability of an HMM's hidden path.*

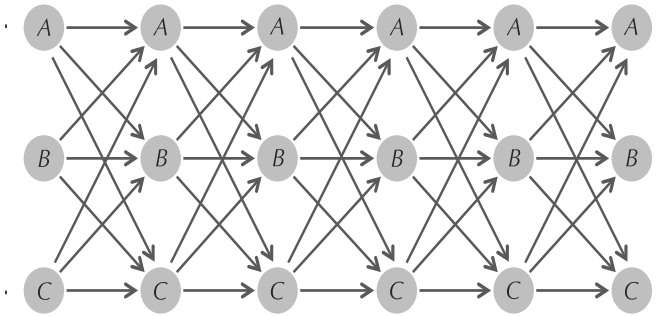
- **Input:** A hidden path π in an HMM (∑,*States*,*Transition*,*Emission*).
- **Output:** The probability of this path, Pr(π).

Probability of an Outcome Given a Hidden Path Problem. *Compute the probability that an HMM will emit a given string given its hidden path.*

- **Input:** A string $x=x_1,\ldots x_n$ emitted by an HMM (∑, *States*, *Transition*, *Emission*) and a hidden path $\pi= \pi_1,\ldots, \pi_n$.
- **Output**: The conditional probability Pr(*x*|π) that *x* will be emitted given that the HMM follows the hidden path π.

Hidden Markov Models

- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model
- Decoding Problem



- The Viterbi Algorithm
- Profile HMMs for Sequence Alignment
- Classifying proteins with profile HMMs
- Viterbi Learning
- Soft Decoding Problem
- Baum-Welch Learning

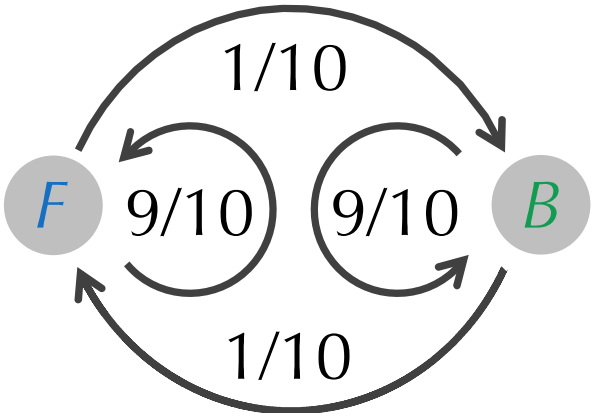
Decoding Problem

Decoding Problem: *Find an optimal hidden path in an HMM given its emitted string.*

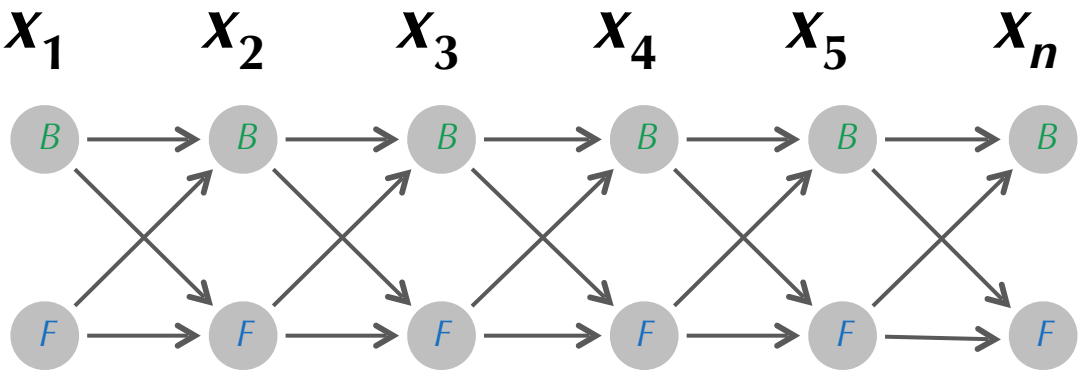
- **Input:** A string $x = x_1 \ldots x_n$ emitted by an HMM $(\sum, States, Transition, Emission)$.
- **Output:** A path π that maximizes the probability $Pr(x,\pi)$ over all possible paths through this HMM.

$$Pr(x, \pi) = Pr(x|\pi) * Pr(\pi)$$
$$= \prod_{i=1,n} Pr(x_i|\pi_i) * Pr(\pi_{i-1} \rightarrow \pi_i)$$
$$= \prod_{i=1,n} emission_{\pi i}(x_i) * transition_{\pi i-1,\pi i}$$

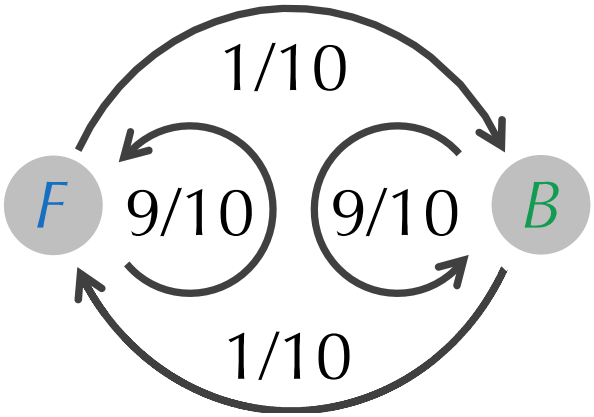
Building Manhattan for the Crooked Casino



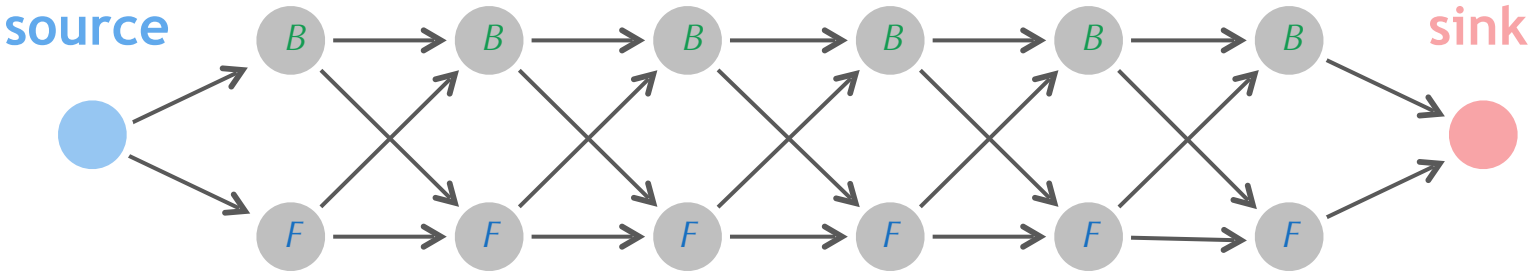
HMM diagram



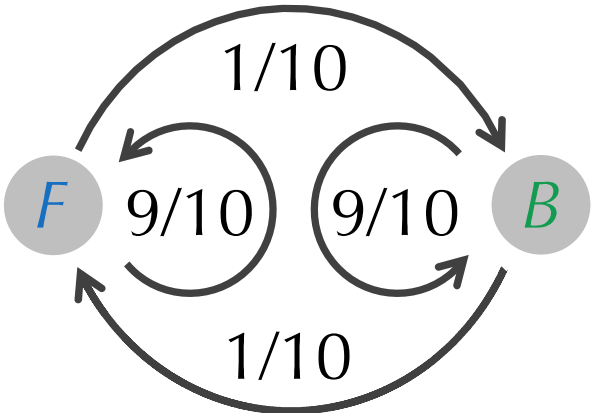
Building Manhattan for the Crooked Casino



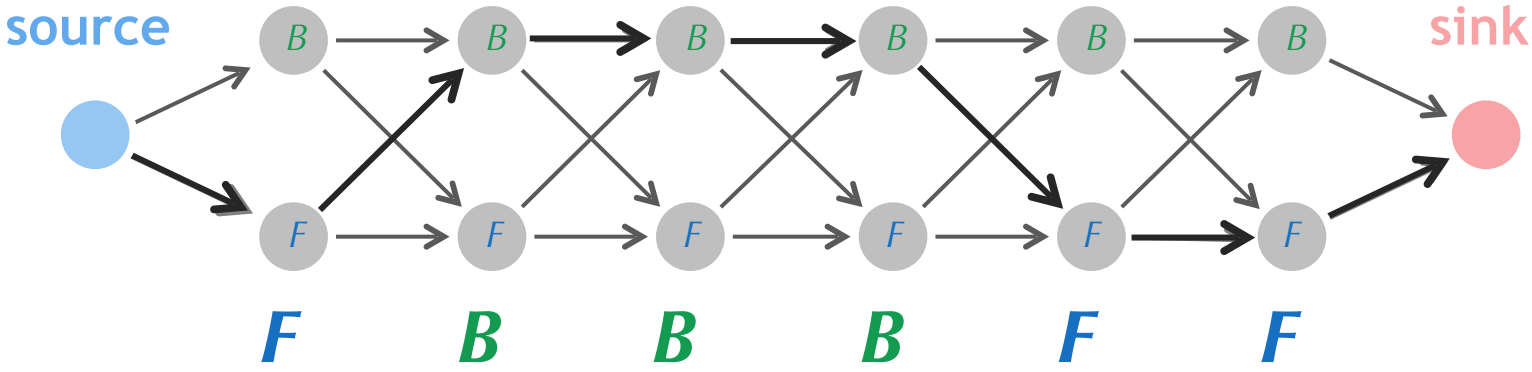
HMM diagram



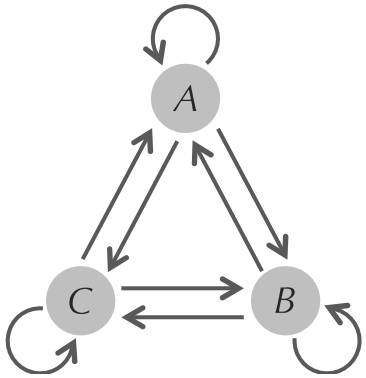
Building Manhattan for the Crooked Casino



HMM diagram

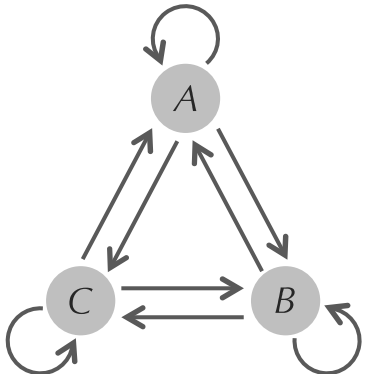


Building Manhattan for Decoding Problem

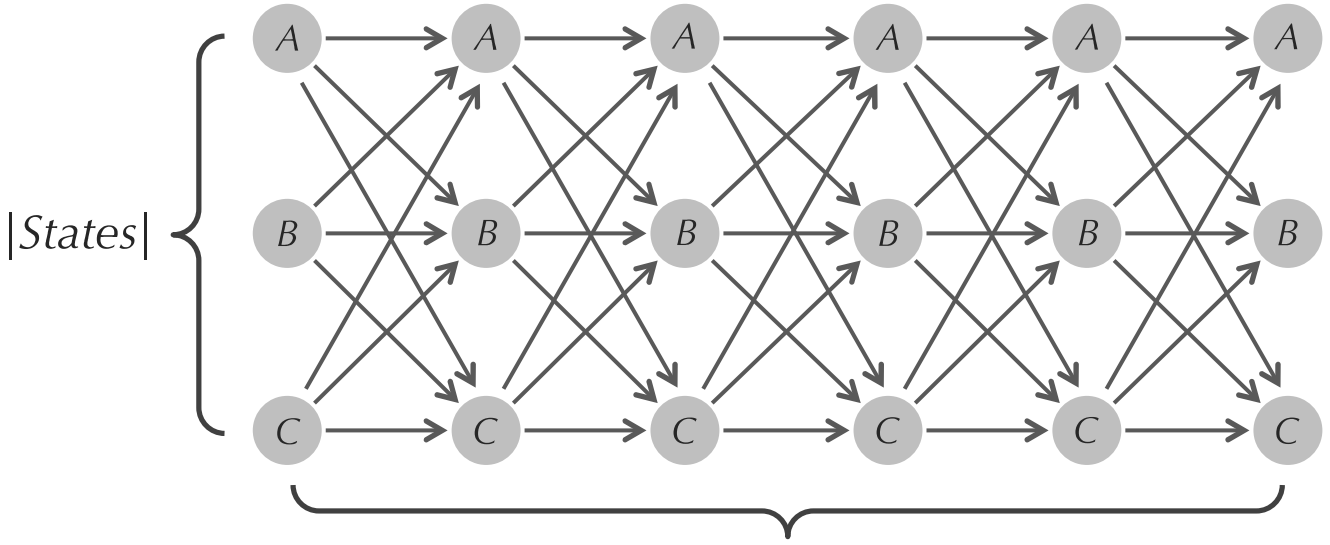


HMM diagram

Building Manhattan for Decoding Problem

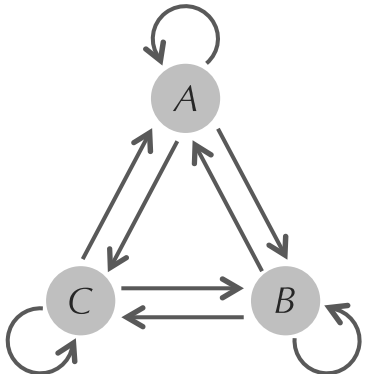


HMM diagram

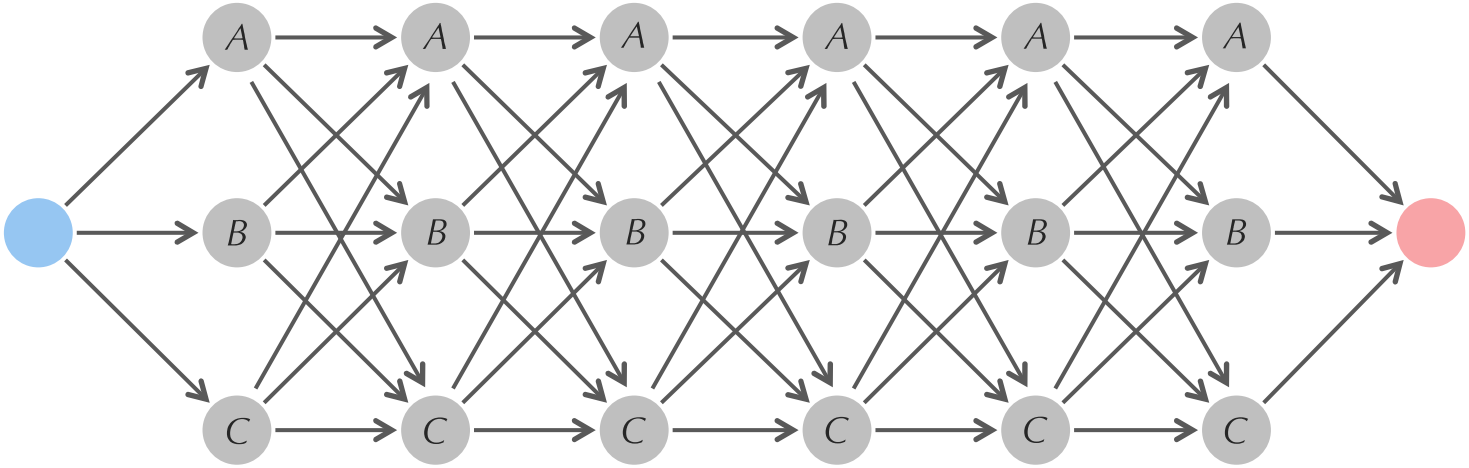


Viterbi diagram (almost)

Building Manhattan for Decoding Problem



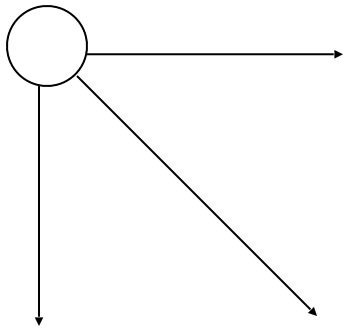
HMM diagram



Alignment Manhattan vs. Decoding Manhattan

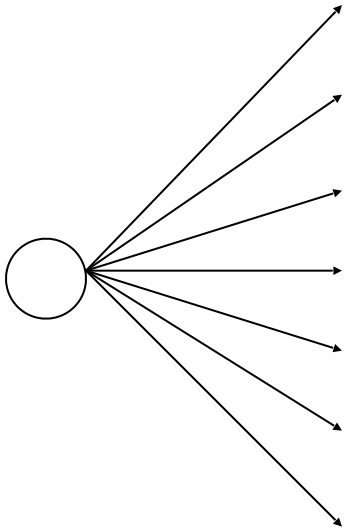
Alignment

three valid directions

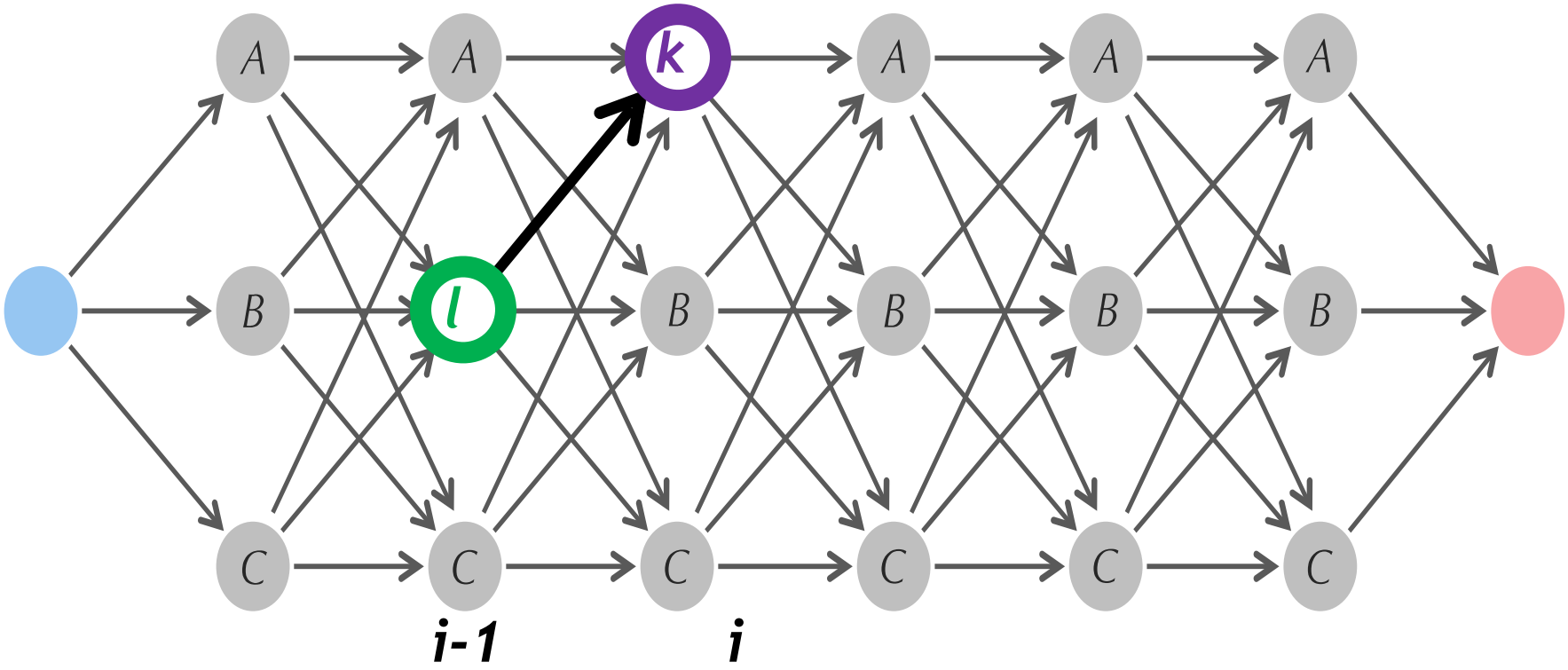


Decoding

many valid directions



Edge Weights in the HMM Manhattan



Edge (*l, k, i-1*) from node (*l, i-1*) to node (*k, i*):

- transitioning from state *l* to state *k* (with probability $transition_{l,k}$)
- emitting symbol x_i (with probability $emission_k(x_i)$)

$$\textbf{\textit{weight}}(l, k, i\text{-}1) = emission_k(x_i) * transition_{l,k}$$

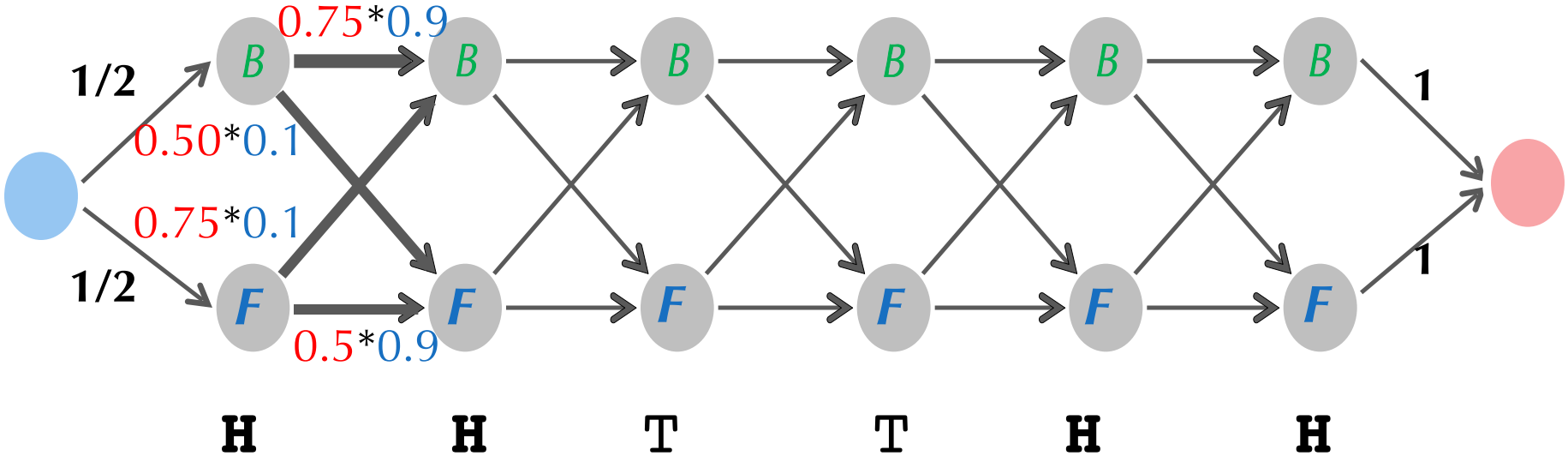
Edge Weights for the Crooked Casino

transition	F	B
F	0.9	0.1
B	0.1	0.9

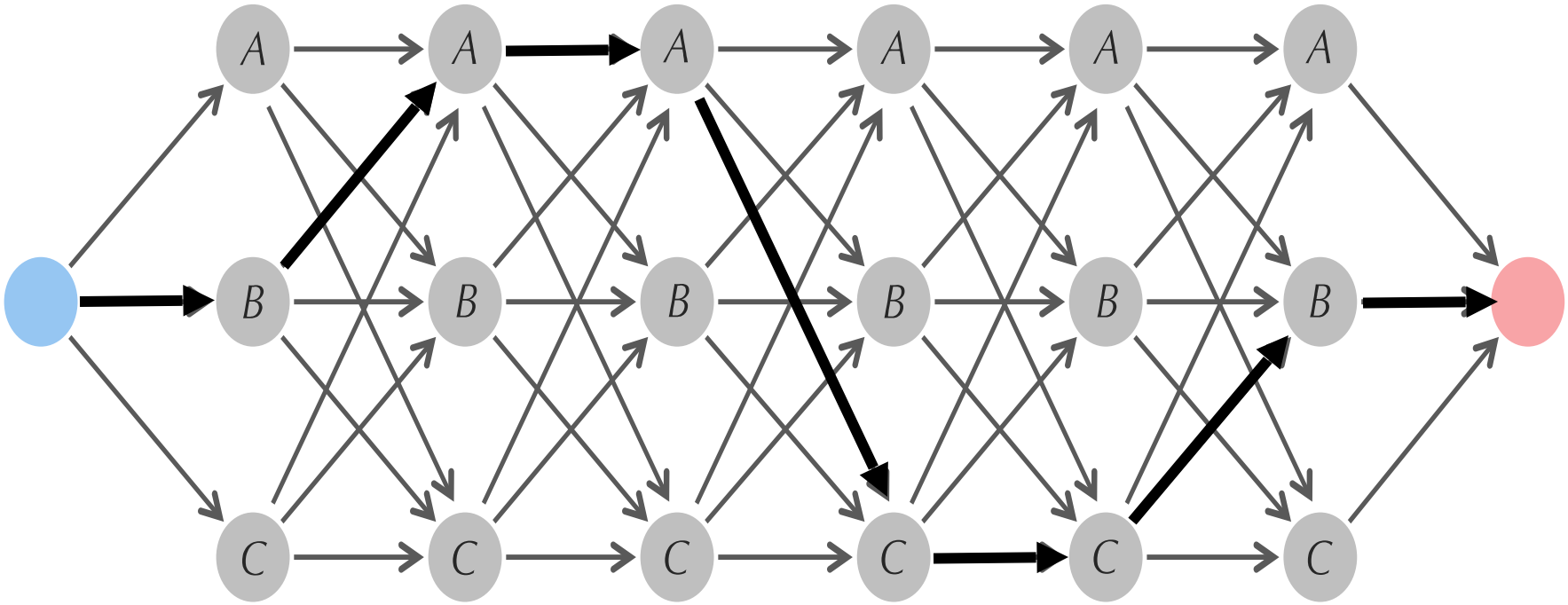
$weight(l,k,i\text{-}1) = emission_k(x_i) * transition_{l,k}$

emission	H	T
F	0.50	0.50
B	0.75	0.25

$weight_1(B,B,1) =$
$emission_B(\text{H}) * transition_{B,B} =$



Product Weight of a Hidden Path



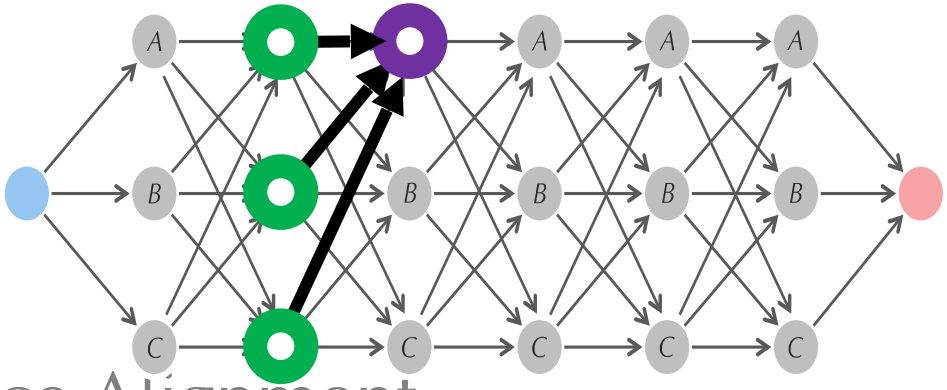
$$\Pr(x, \pi) = \prod_{i=1,n} emission_{\pi i}(x_i) * transition_{\pi i-1, \pi i}$$

$$= \prod_{i=1,n} \text{weight of the } i\text{-th edge in path } \pi$$

$$= \prod_{i=1,n} \mathbf{weight}(\pi_{i-1}, \pi_i, i-1)$$

Hidden Markov Models

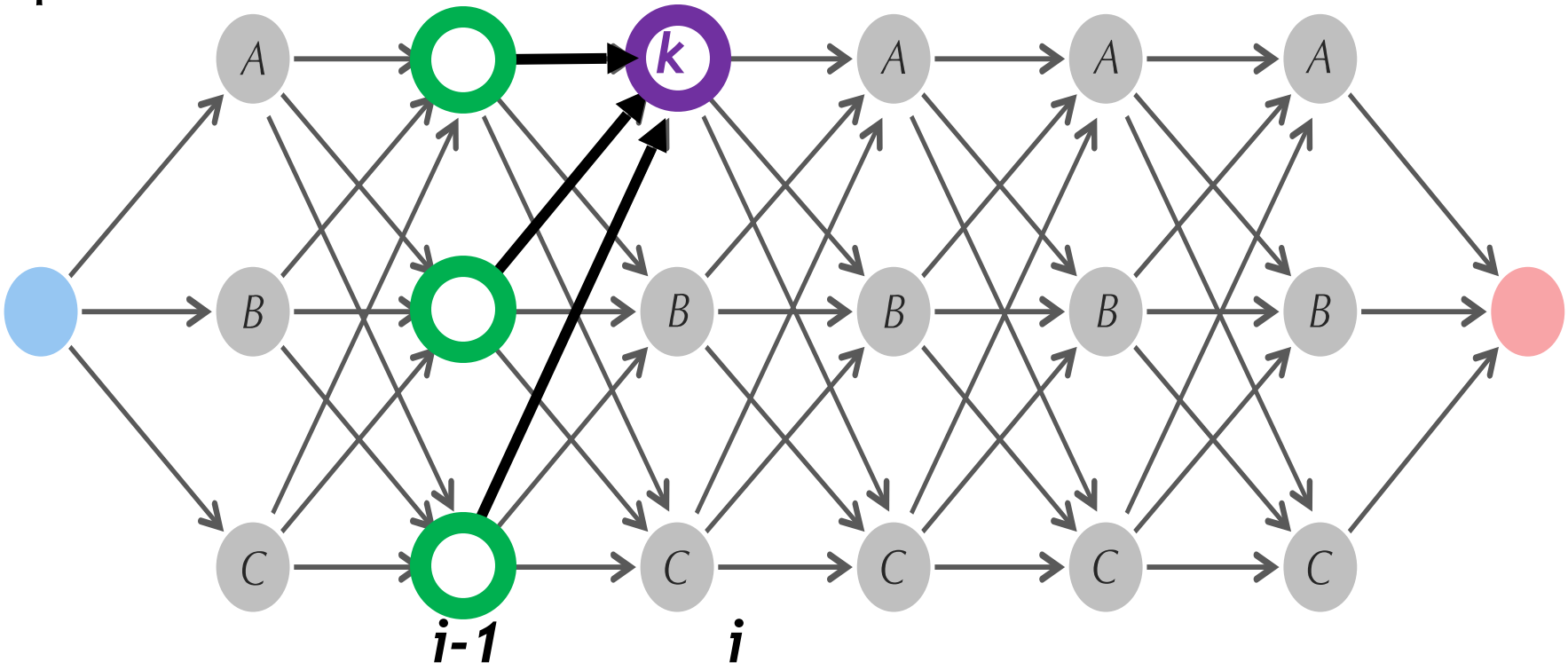
- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model
- Decoding Problem
- The Viterbi Algorithm



- Profile HMMs for Sequence Alignment
- Classifying proteins with profile HMMs
- Viterbi Learning
- Soft Decoding Problem
- Baum-Welch Learning

Dynamic Programming for Decoding Problem

$score_{k,i}$: the maximum product weight among all paths from *source* to node (k, i):



$$score_{k,i} = \max_{\text{all states } l} \{score_{l,i-1} \cdot \text{weight of edge from } (l,i\text{-}1) \text{ to } (k,i)\}$$

$$= \max_{\text{all states } l} \{score_{l,i-1} \cdot weight(l,k,\ i\text{-}1)\}$$

Recurrence for Viterbi Algorithm

- Recurrence:

$$score_{k,\,i} = \max_{\text{all states } l} \{score_{l,i\text{-}1} \cdot weight(l,k,i\text{-}1)\}$$

- Initialization:

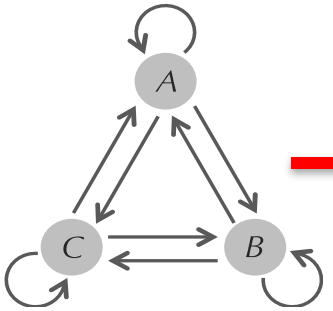
$$score_{source} = 1$$

- The maximum product weight over all paths from *source* to *sink*: (n = | x | , x = emitted string)

$$score_{sink} = \max_{\text{all states } l} \{score_{l,n}\}$$

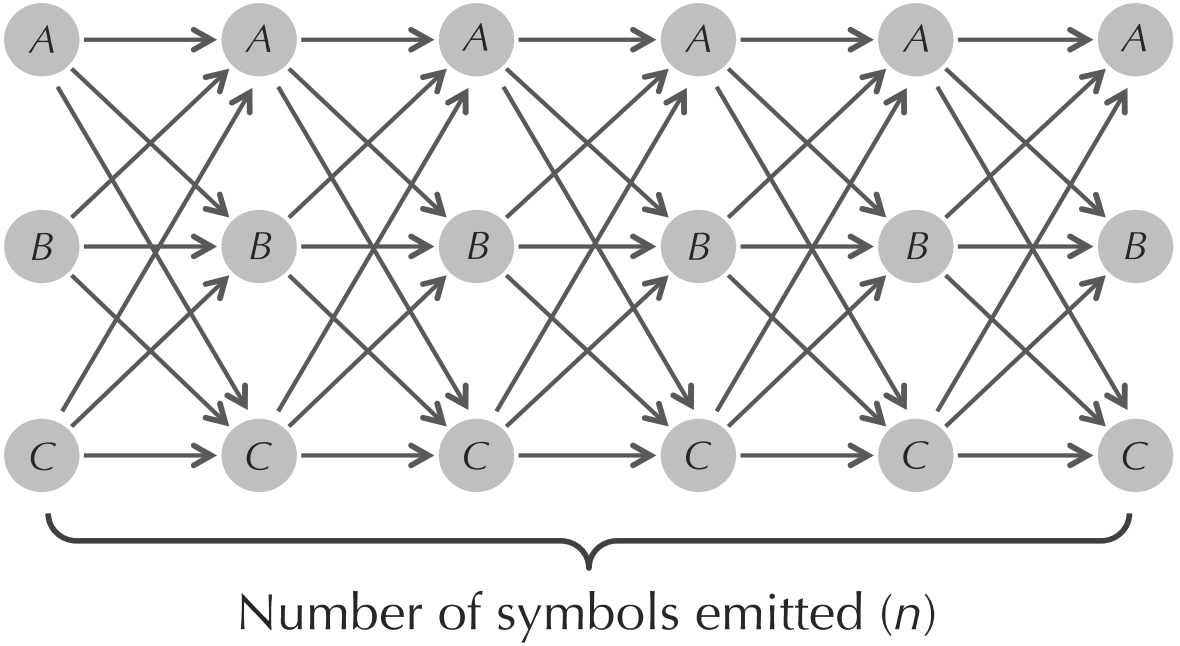
Running Time of the Viterbi Algorithm

HMM Diagram



|*States*|

Viterbi Diagram

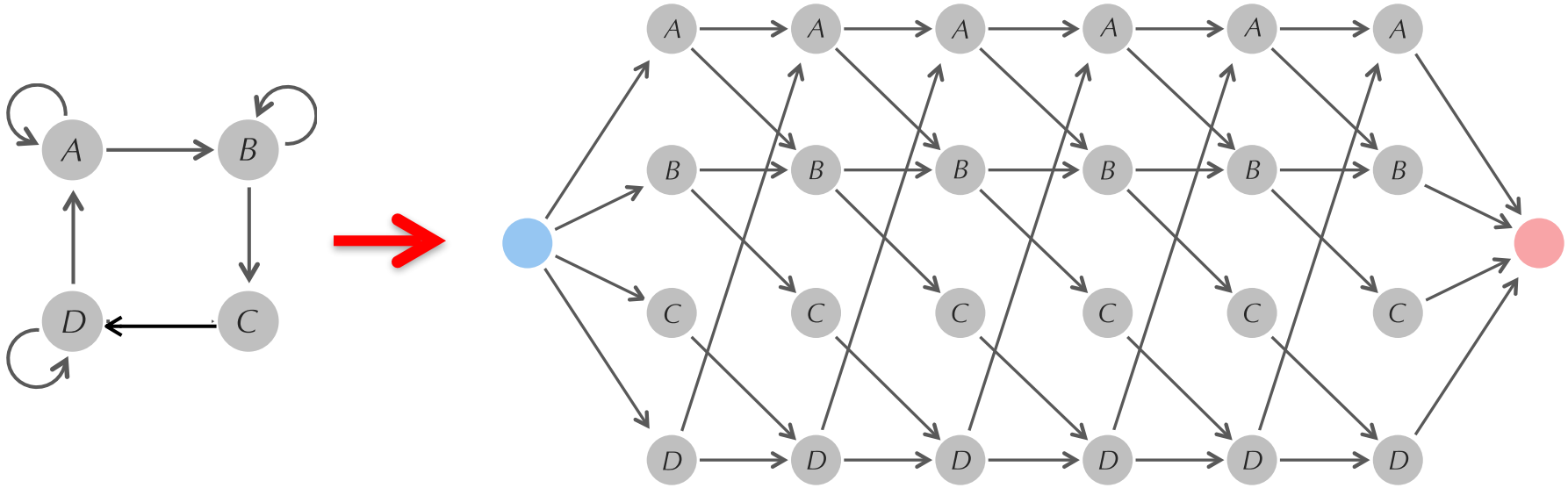


Running time ~ #edges in the Viterbi graph

~ O(|***States***|^2 • ***n***)

Running Time of the Viterbi Algorithm

Forbidden transition: an edge not represented in the HMM diagram.



Running time ~ #edges in the Viterbi graph
~ O(**#edges in the HMM diagram•*n***)

From Product of Weights to Sum of Their Logarithms

Since $score_{k,i}$ may become small (danger of underflow), we prefer to work with logarithms of scores:

$$score_{k,i} = \max_{\text{all states } l} \{ score_{l,i-1} \cdot weight(l,k,i-1) \}$$

$$\log(score_{k,i}) = \max_{\text{all states } l} \{ \log(score_{l,i-1}) + \log(weight(l,k,i-1)) \}$$

This transformation substitutes weights of edges by their logarithms:

product of weights → **sum** of weights

Computing Pr(π) Versus Computing Pr(x)

- Pr(x, π): the probability that an HMM follows the hidden path π and emits the string $x = x_1 x_2 \ldots x_n$.

x:	T	H	T	H	H	H	T	H	T	T	H
π:	F	F	F	B	B	B	B	B	F	F	F
	.5	.9	.9	.1	.9	.9	.9	.9	.1	.9	.9

$$\Pr(\pi) = \sum_{\text{all possible emitted strings } x} \Pr(x, \pi) = \prod_{i=1,n} transition_{\pi i-1, \pi i}$$

$$\Pr(x) = \sum_{\text{all possible hidden paths } \pi} \Pr(x, \pi)$$

$$\Pr(x) = \sum_{\text{all possible hidden paths } \pi} \text{product weight of } \pi$$

$$score_{sink} = \max_{\text{all possible hidden paths } \pi} \text{product weight of } \pi$$

What is the Most Likely Outcome of an HMM?

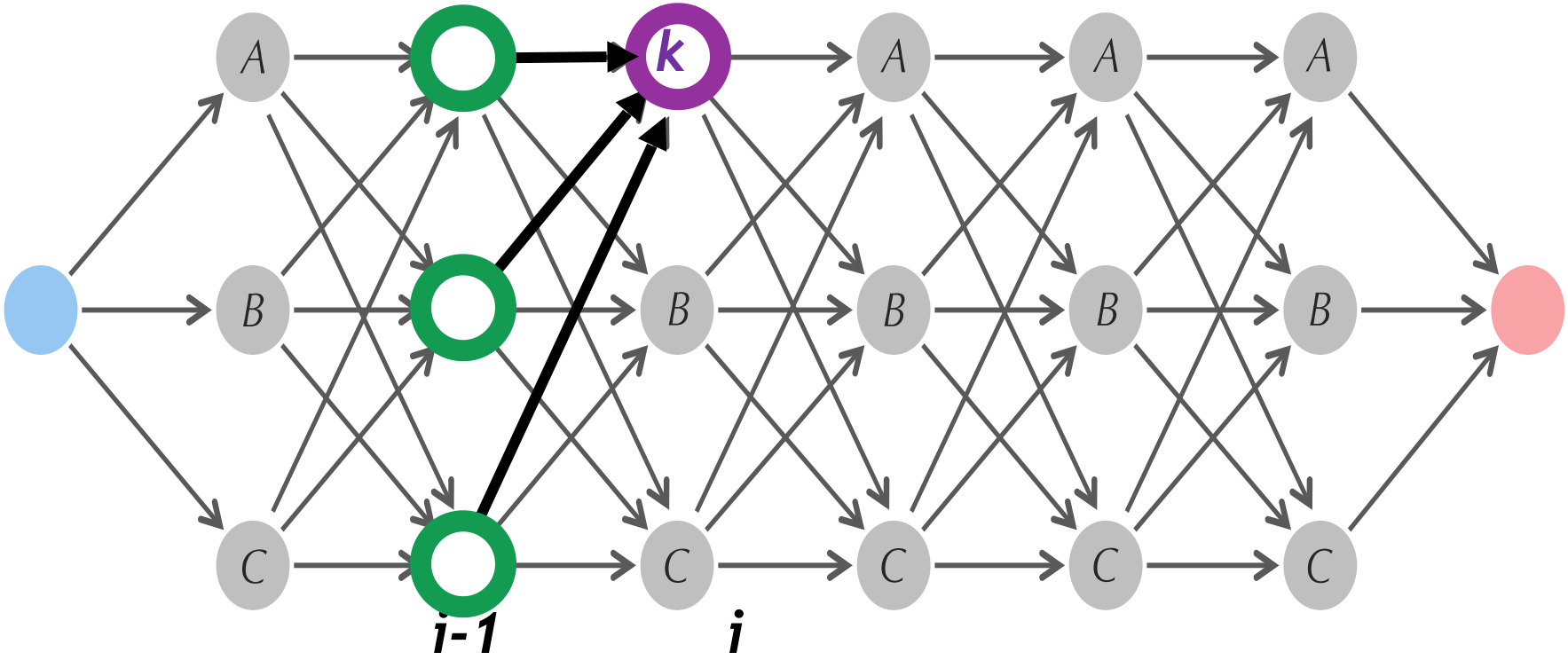
- **Outcome Likelihood Problem**. *Find the probability that an HMM emits a given string.*
- **Input:** A string $x = x_1 \ldots x_n$ emitted by an HMM ($\sum$, *States, Transition, Emission*).
- **Output**: The probability Pr(x) that the HMM emits x.

Can we solve the Outcome Likelihood Problem by making a *single change* in the Viterbi recurrence?

$$score_{k,\,i} = \max_{\text{all states } l} \{score_{l,i-1} \cdot weight(l,k,i-1)\} \; ?$$

Viterbi Algorithm: From MAX to $\sum$

- $forward_{k,i}$: **total** product weight of all paths to (k,i):

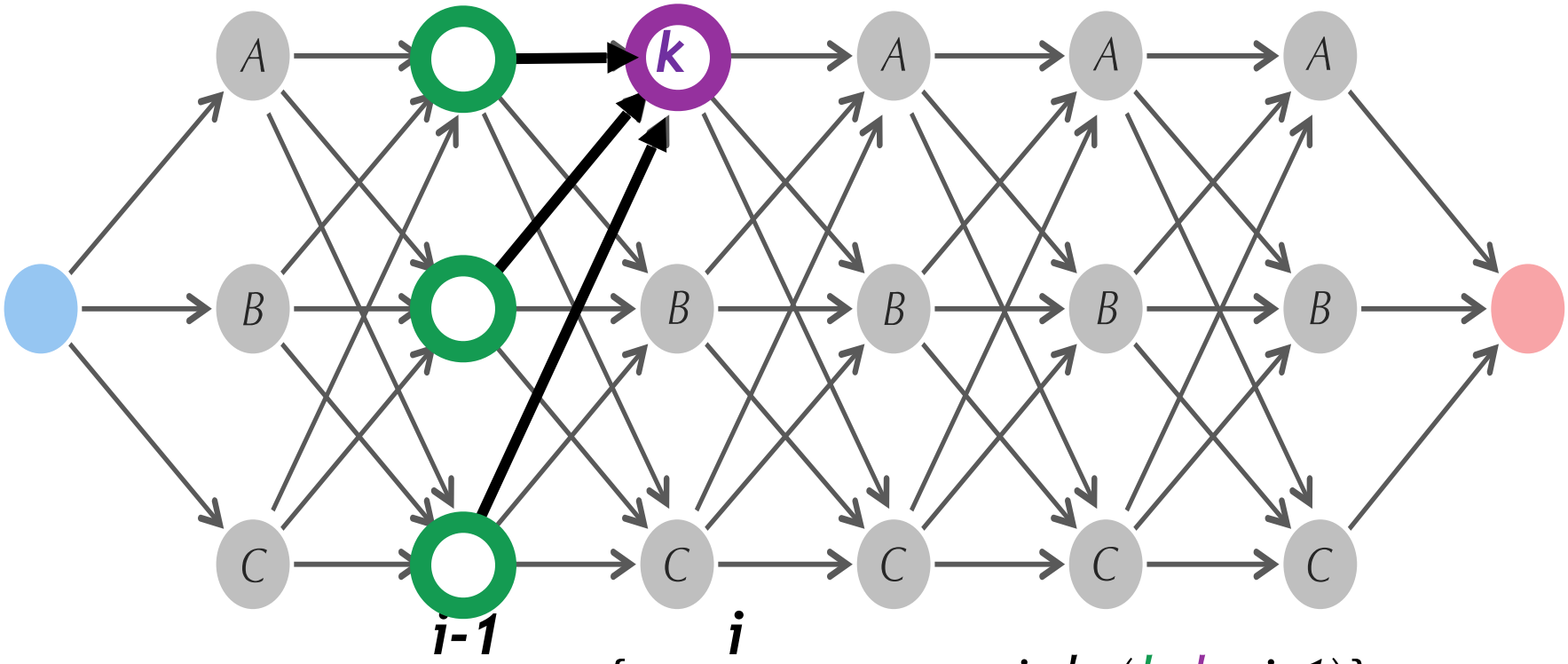


$$forward_{k,i} = \sum_{\text{all states } l} \{forward_{l,i-1} \cdot \text{weight of } (l,i\text{-}1) \rightarrow (k,i)\}$$

$$= \sum_{\text{all states } l} \{forward_{l,i-1} \cdot weight(l, k, i\text{-}1)\}$$

Viterbi Algorithm: From MAX to $\sum$

- $forward_{k,i}$: **total** product weight of all paths to (k,i):

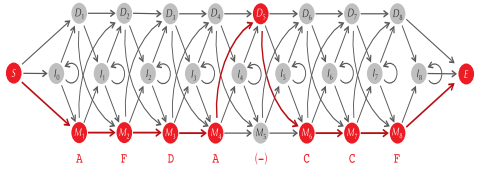


$$score_{k,i} = \mathbf{max}_{\text{all states } l} \{ score_{l,i-1} \cdot weight(l, k, i\text{-}1) \}$$

$$forward_{k,i} = \sum_{\text{all states } l} \{ forward_{l,i-1} \cdot \text{weight of } (l, k, i\text{-}1 \}$$

Hidden Markov Models

- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model
- Decoding Problem
- The Viterbi Algorithm
- Profile HMMs for Sequence Alignment

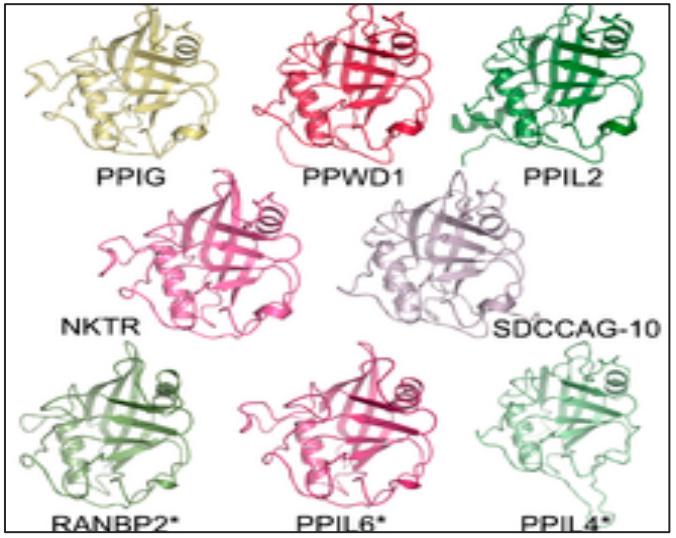


- Classifying proteins with profile HMMs
- Viterbi Learning
- Soft Decoding Problem
- Baum-Welch Learning

Classifying Proteins into Families

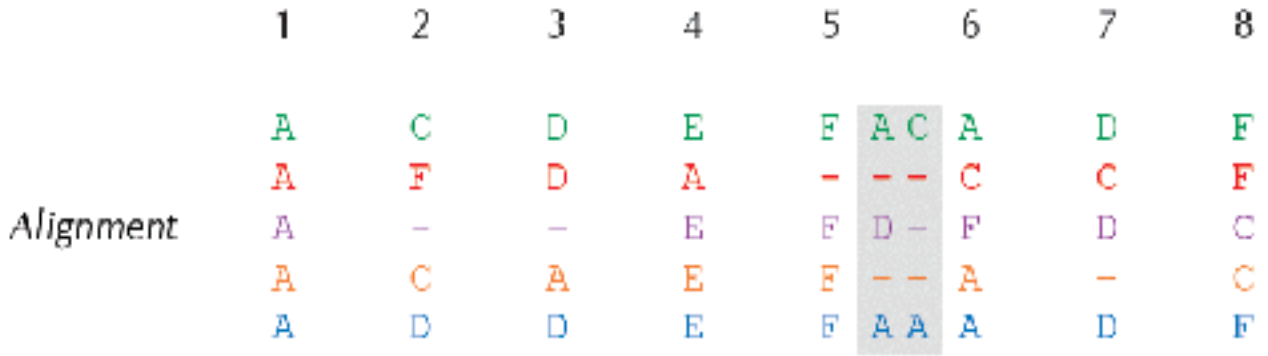
- Proteins are organized into **protein families** represented by multiple alignments.

- A distant cousin may have weak *pairwise* similarities with family members failing a significance test.



- However, it may have weak similarities with *many* family members, indicating a relationship.

From Alignment to Profile



Remove columns if the fraction of space symbols ("-") exceeds θ, **the maximum fraction of insertions threshold**.

From Alignment to Profile

	1	2	3	4	5		6	7	8
	A	C	D	E	F	A C	A	D	F
	A	F	D	A	–	– –	C	C	F
Alignment	A	–	–	E	F	D –	F	D	C
	A	C	A	E	F	– –	A	–	C
	A	D	D	E	F	A A	A	D	F
	A	C	D	E	F		A	D	F
	A	F	D	A	–		C	C	F
*Alignment**	A	–	–	E	F		F	D	C
	A	C	A	E	F		A	–	C
	A	D	D	E	F		A	D	F

From Alignment to Profile

		1	2	3	4	5		6	7	8
Alignment		A	C	D	E	F	A C	A	D	F
		A	F	D	A	–	– –	C	C	F
		A	–	–	E	F	D –	F	D	C
		A	C	A	E	F	– –	A	–	C
		A	D	D	E	F	A A	A	D	F
*Alignment**		A	C	D	E	F		A	D	F
		A	F	D	A	–		C	C	F
		A	–	–	E	F		F	D	C
		A	C	A	E	F		A	–	C
		A	D	D	E	F		A	D	F
PROFILE(*Alignment**)	A	1	0	0	1/5	0		3/5	0	0
	C	0	2/4	0	0	0		1/5	1/4	2/5
	D	0	1/4	3/4	0	0		0	3/4	0
	E	0	0	0	4/5	0		0	0	0
	F	0	1/4	0	0	1		1/5	0	3/5

From Profile to HMM

	1	2	3	4	5	6	7	8
Alignment	A	C	D	E	F	A C	A	D	F
	A	F	D	A	–	– –	C	C	F
	A	–	–	E	F	D –	F	D	C
	A	C	A	E	F	– –	A	–	C
	A	D	D	E	F	A A	A	D	F
*Alignment**	A	C	D	E	F		A	D	F
	A	F	D	A	–		C	C	F
	A	–	–	E	F		F	D	C
	A	C	A	E	F		A	–	C
	A	D	D	E	F		A	D	F

PROFILE(*Alignment**)

	1	2	3	4	5	6	7	8
A	1	0	0	1/5	0	3/5	0	0
C	0	2/4	0	0	0	1/5	1/4	2/5
D	0	1/4	3/4	0	0	0	3/4	0
E	0	0	0	4/5	0	0	0	0
F	0	1/4	0	0	1	1/5	0	3/5

HMM diagram

A D D A F F D F

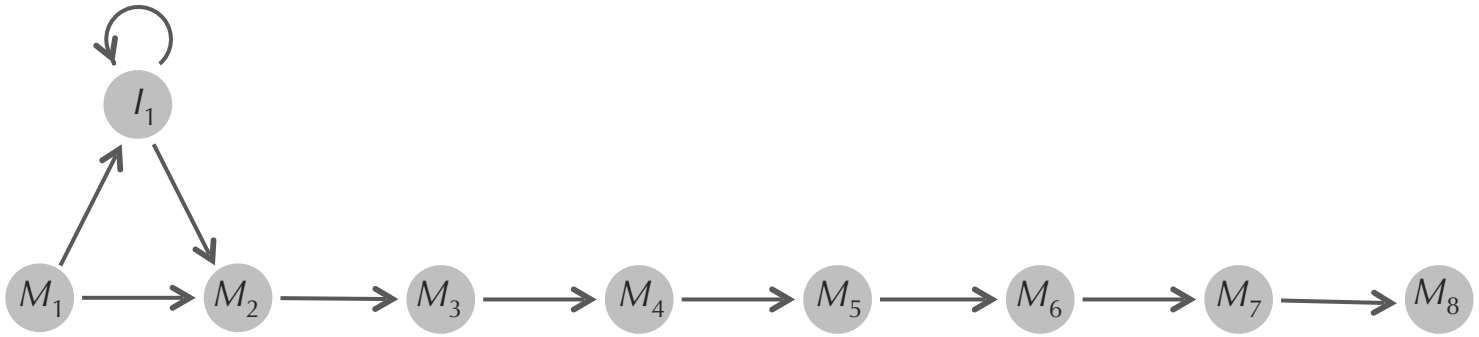
1 * .25 * .75 * .20 * 1 * .20 * .75 * .60

Toward a Profile HMM

A **F** D D A F F D F

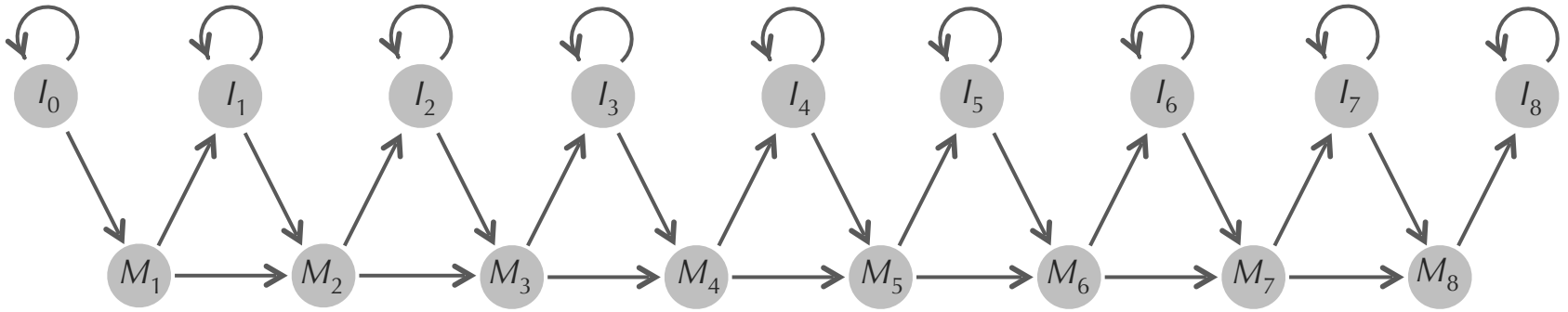
How do we model insertions?

Toward a Profile HMM: Insertions



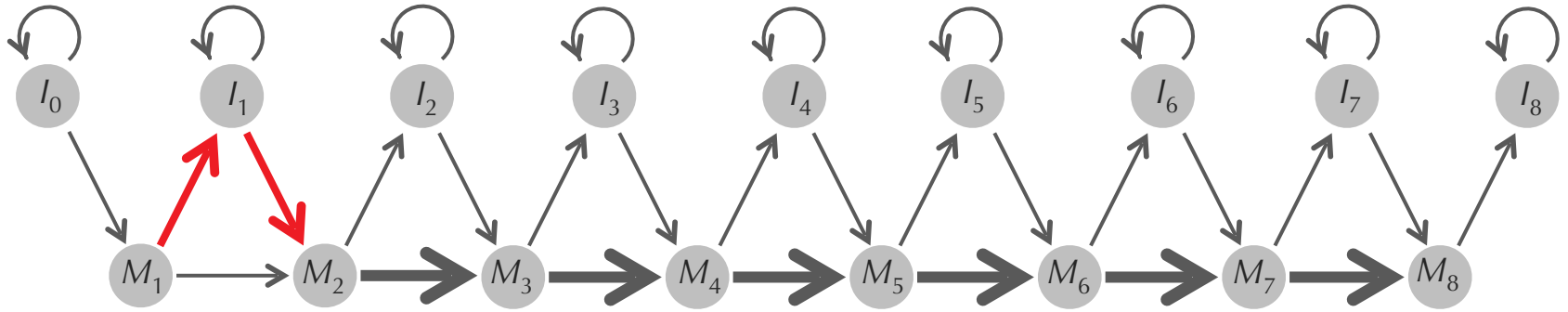
A F D D A F F D F

Toward a Profile HMM: Insertions



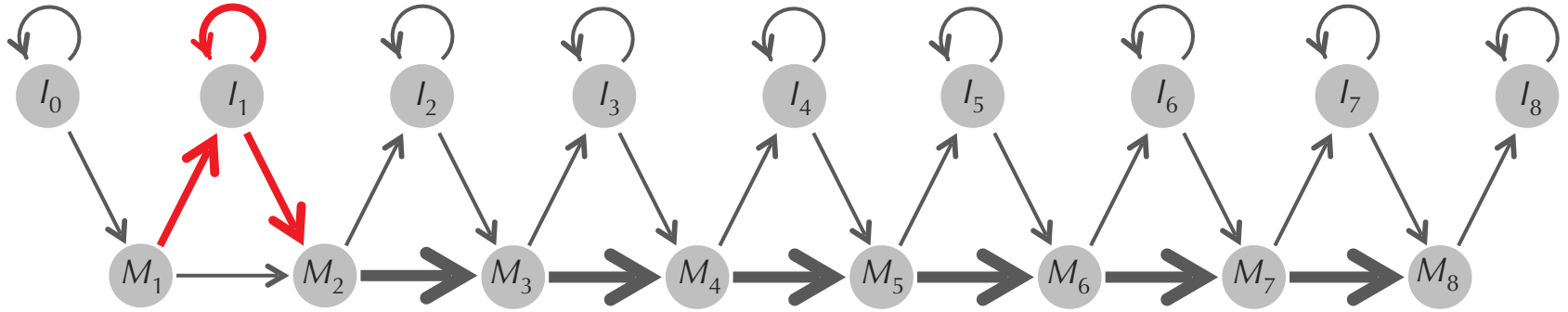
A F D D A F F D F

Toward a Profile HMM: Insertions



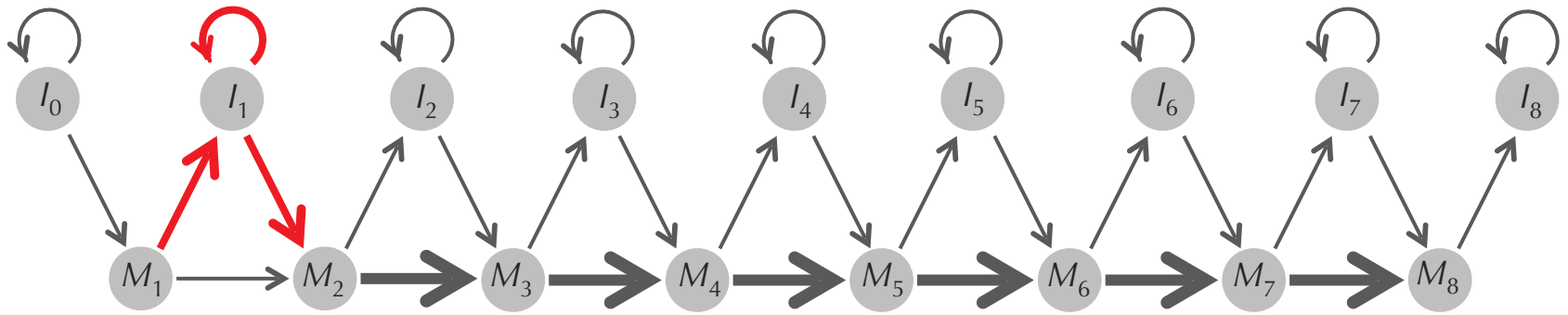
A **F** D D A F F D F

Toward a Profile HMM: Insertions



A **F** D D A F F D F

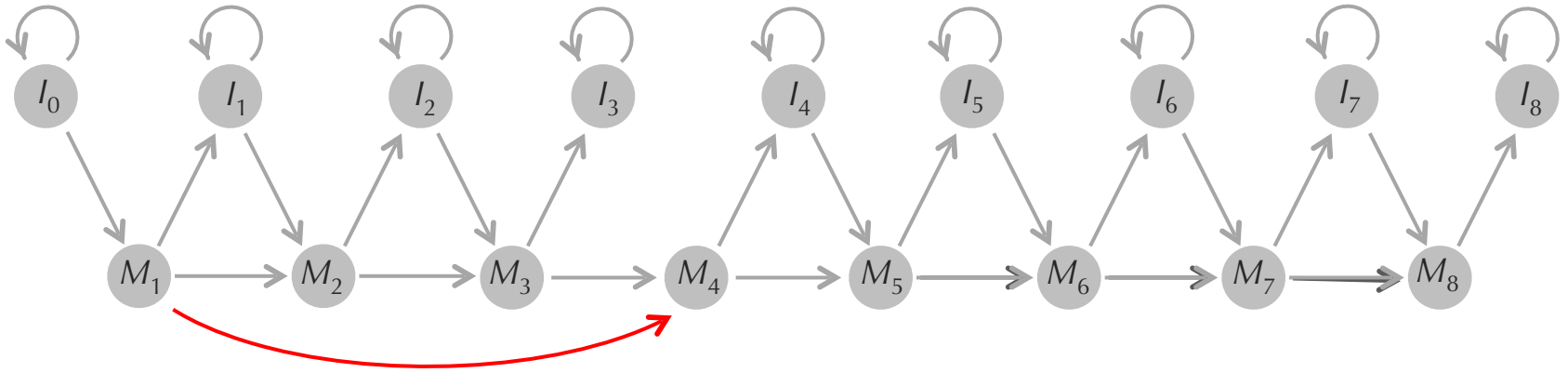
Toward a Profile HMM: Insertions



A **F** D D A F F D F

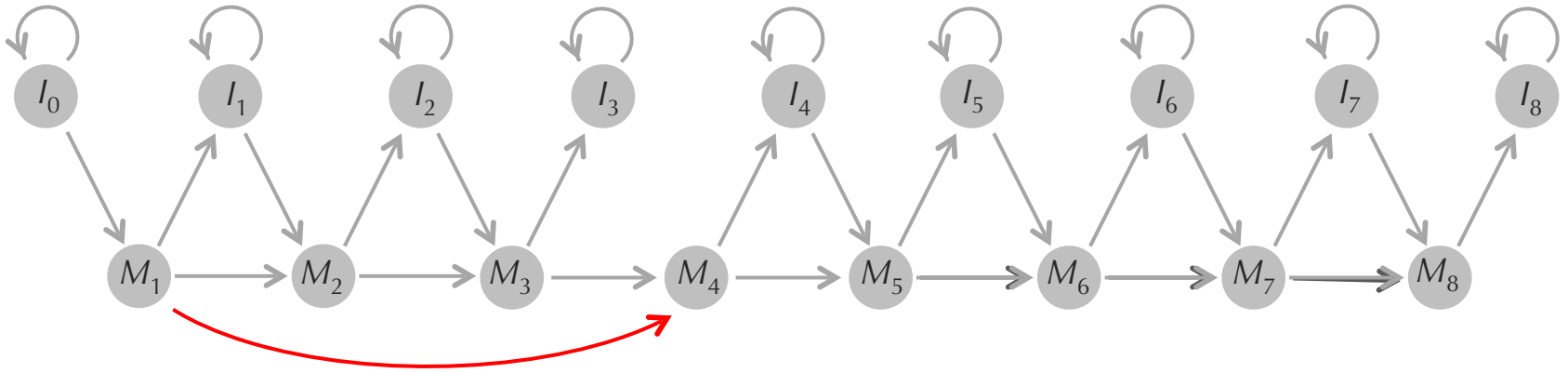
How do we model deletions?

Toward a Profile HMM: Deletions



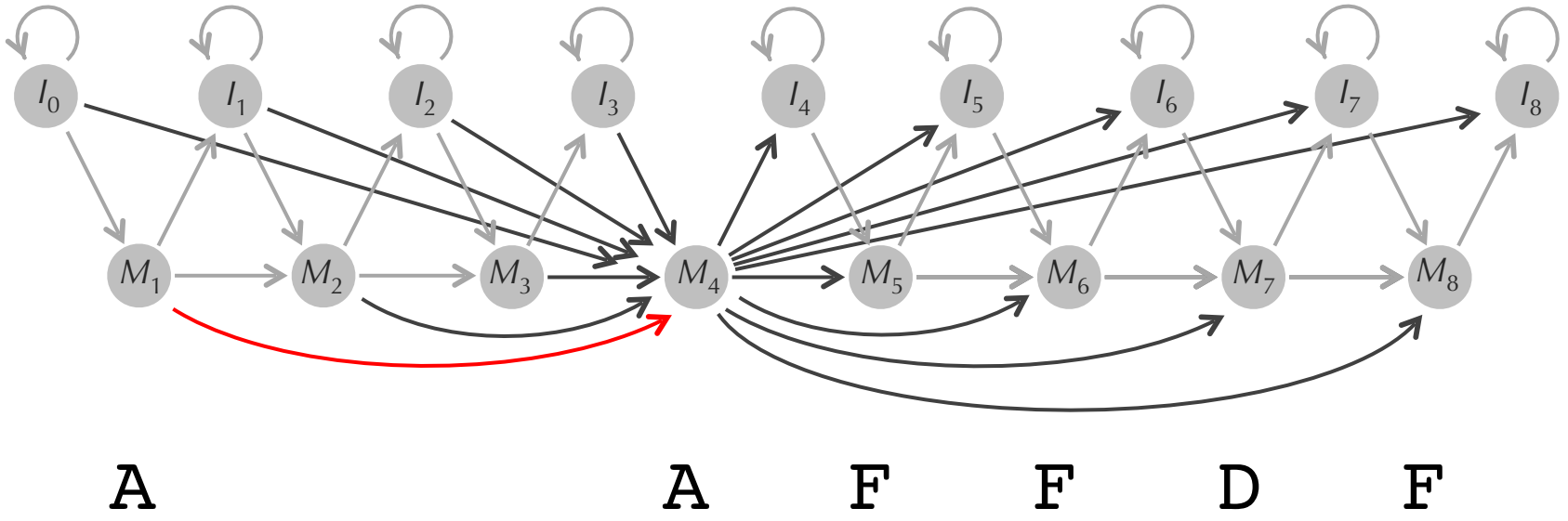
A A F F D F

Toward a Profile HMM: Deletions



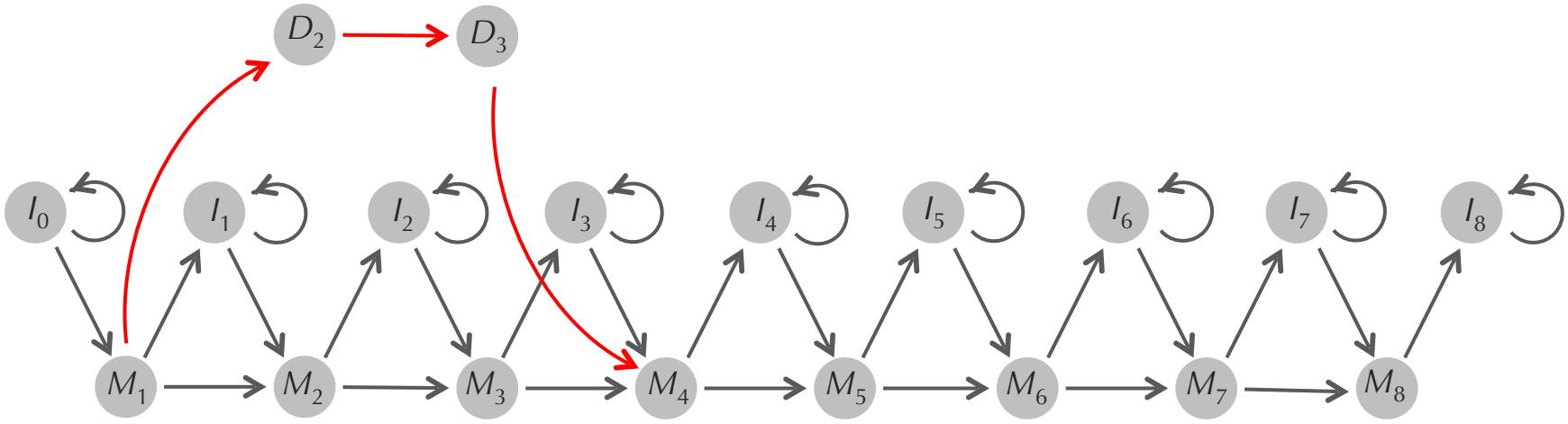
A A F F D F

Toward a Profile HMM: Deletions



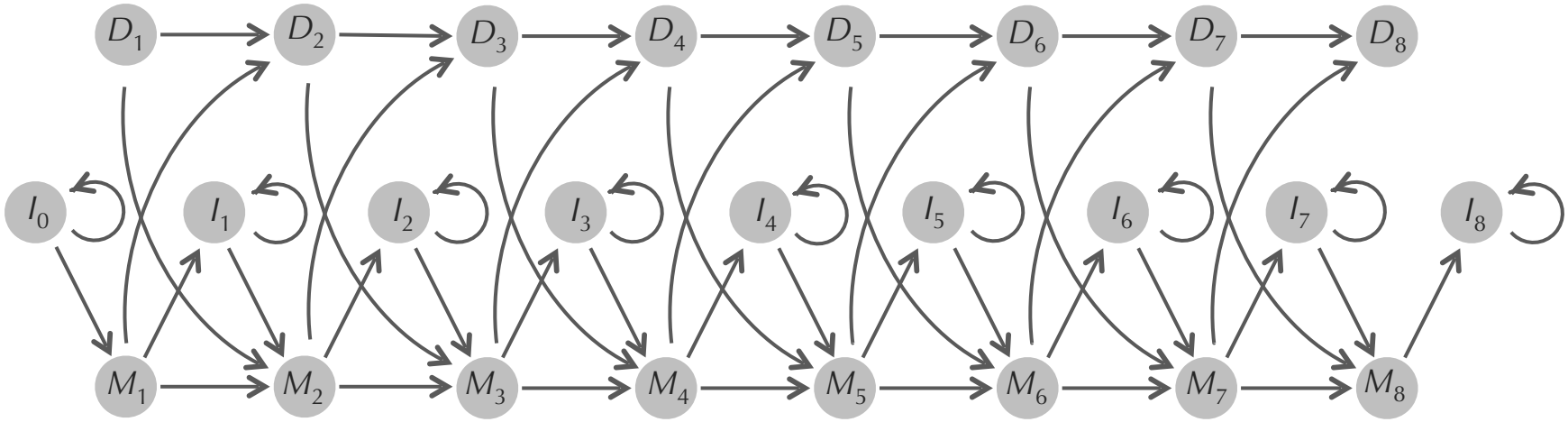
How many edges are in this HMM diagram?

Adding "Deletion States"



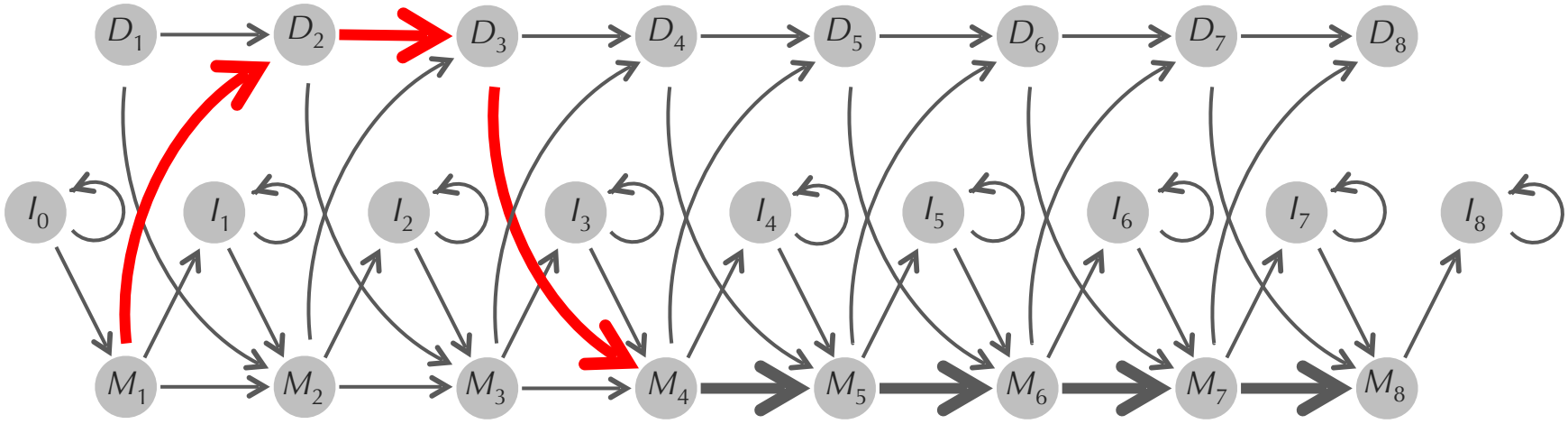
A A F F D F

Adding "Deletion States"



A A F F D F

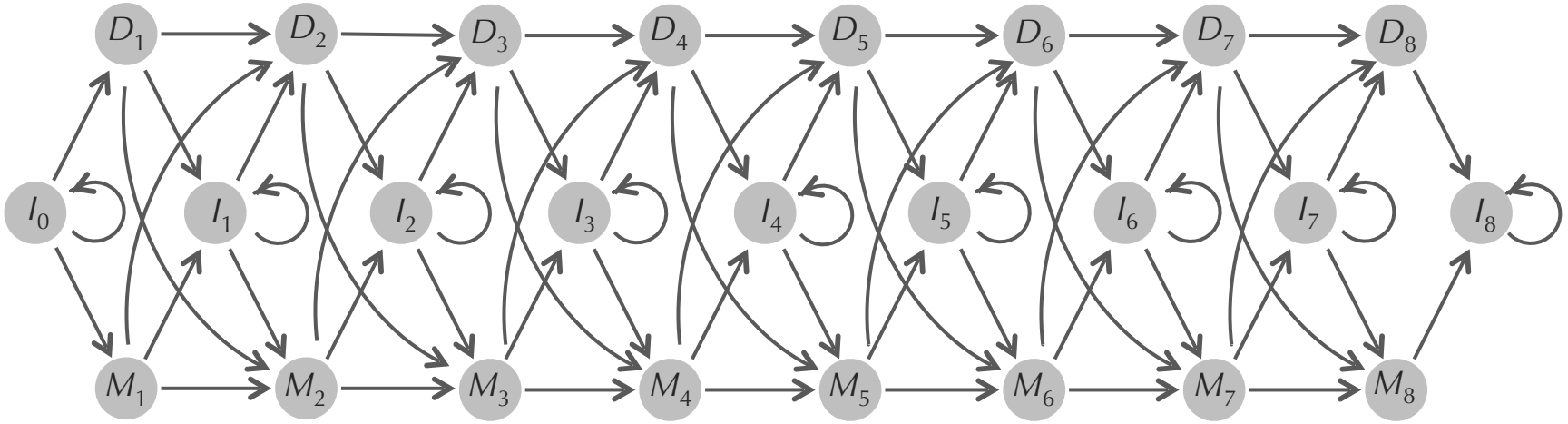
Adding "Deletion States"



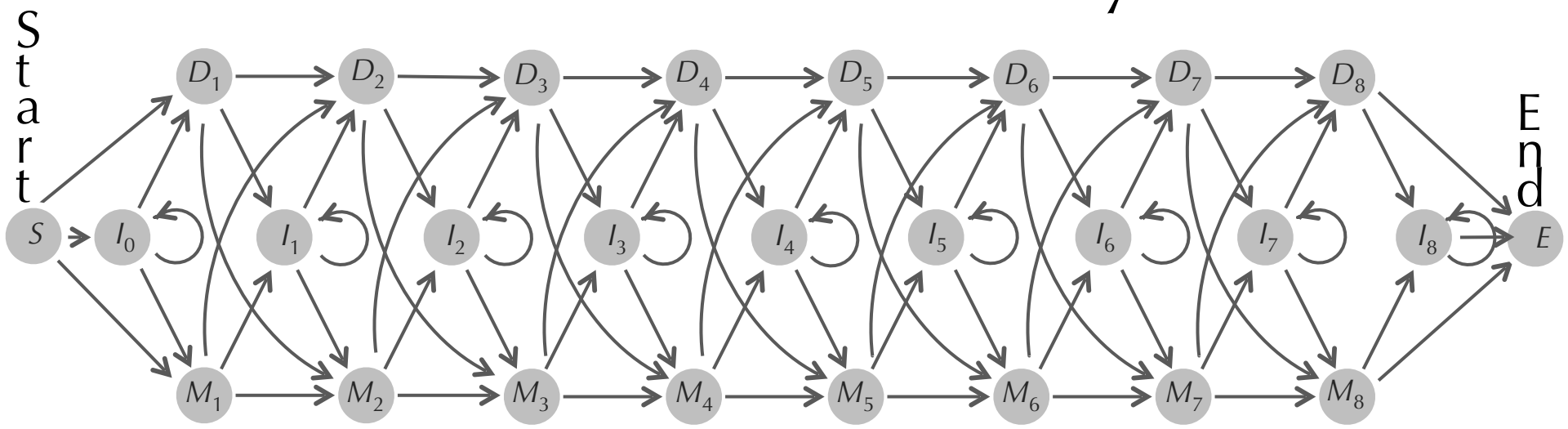
A A F F D F

Are any edges still missing in this HMM diagram?

Adding Edges Between Deletion/Insertion States



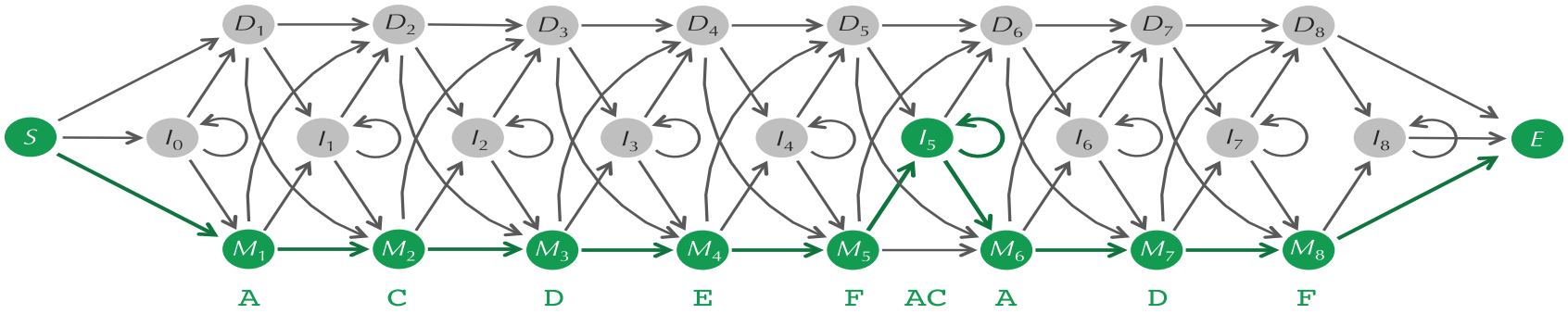
The Profile HMM is Ready to Use!

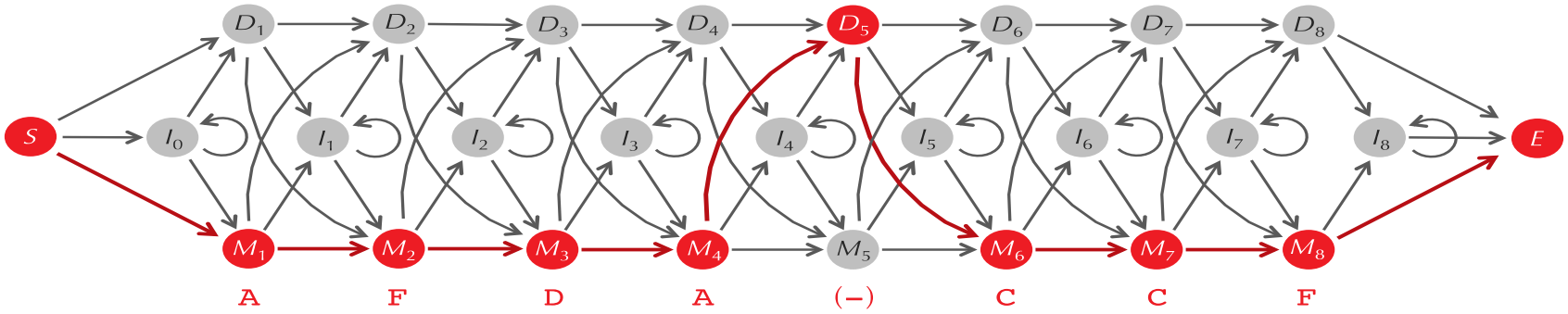


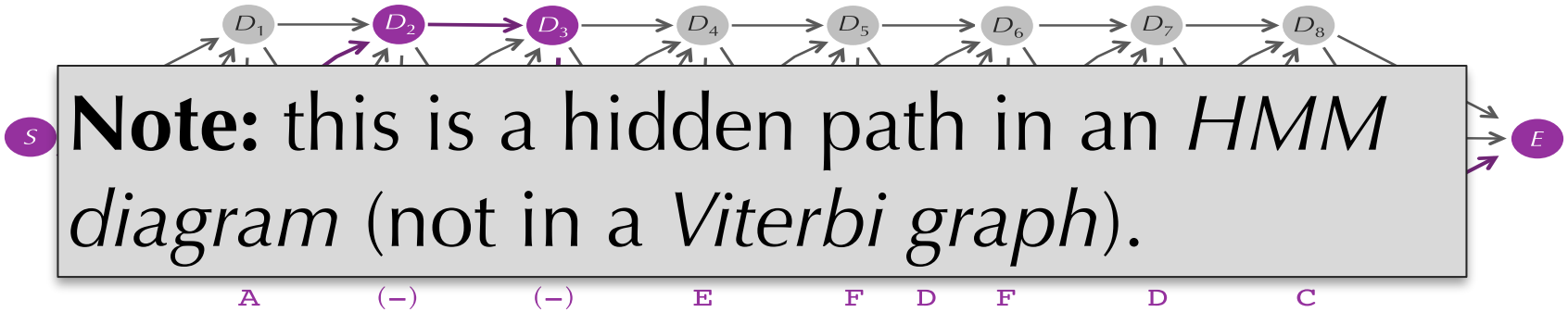
Profile HMM Problem: *Construct a profile HMM from a multiple alignment.*

- **Input:** A multiple alignment *Alignment* and a threshold θ (maximum fraction of insertions per column).
- **Output:** Transition and emission matrices of the profile HMM *HMM*(*Alignment*,θ).

Hidden Paths Through Profile HMM

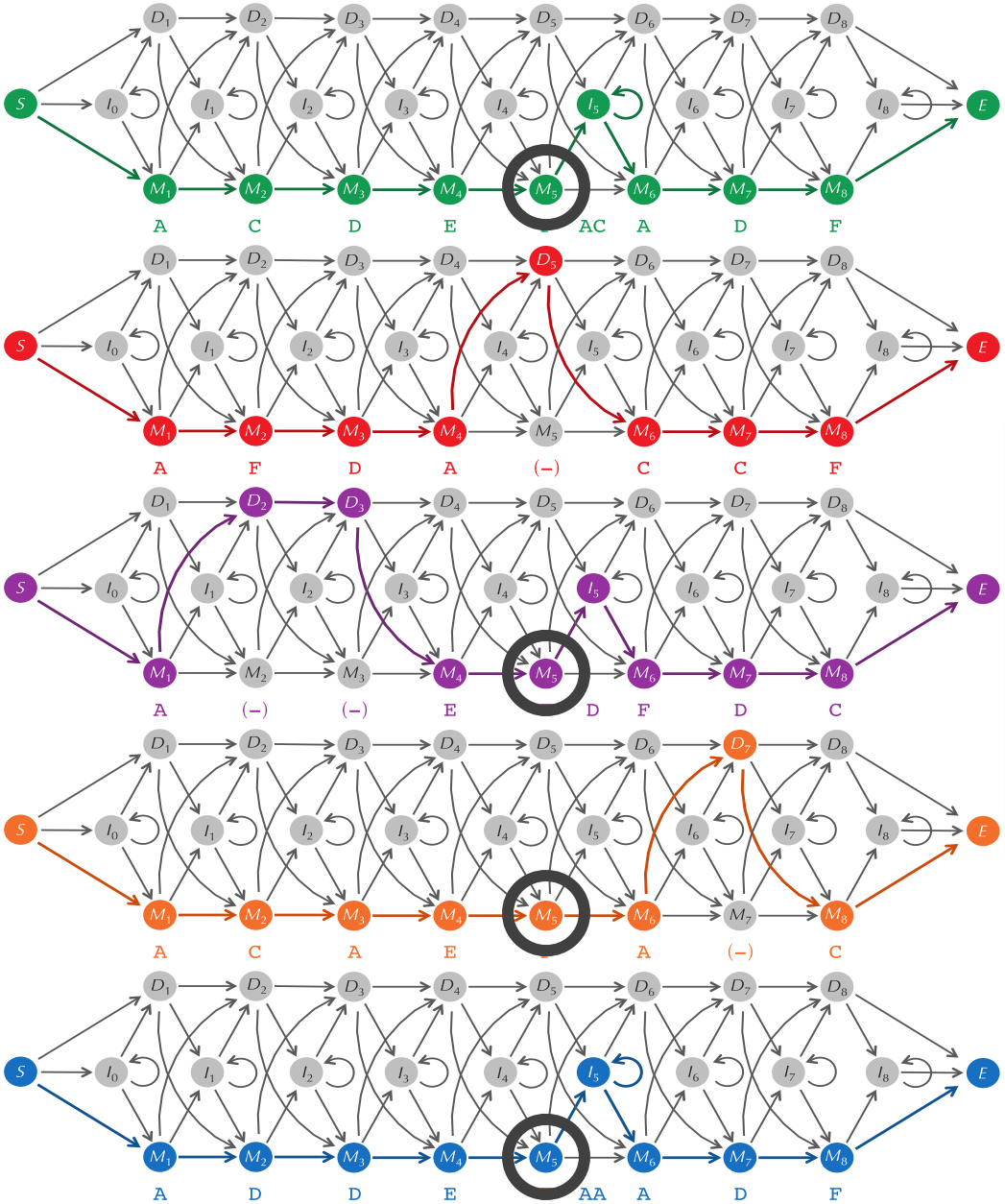






Note: this is a hidden path in an *HMM diagram* (not in a *Viterbi graph*).

Transition Probabilities of Profile HMM



4 transitions from M_5:

1 + 1 + 1 = 3 into I_5

1 into M_6

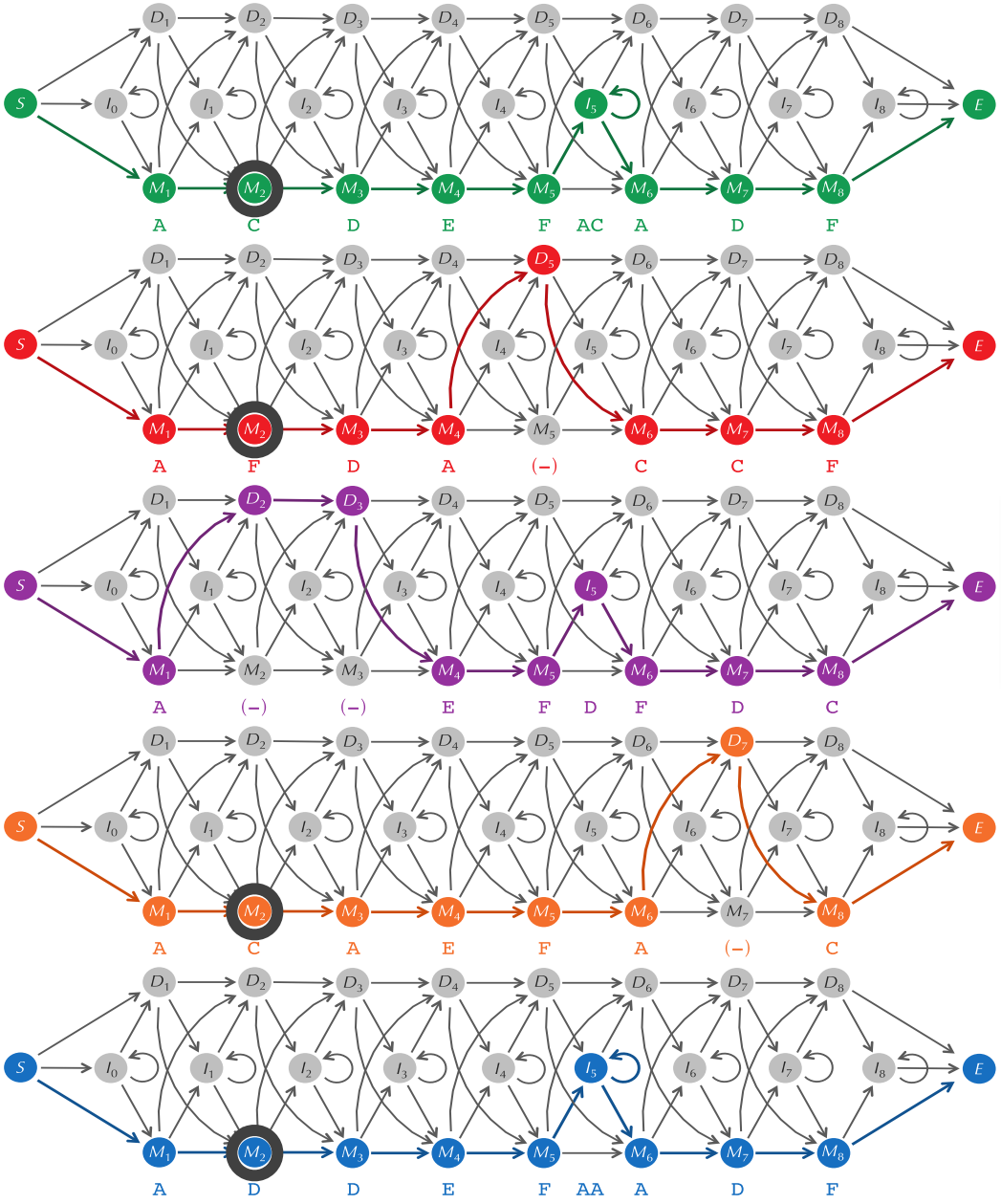
0 into D_6

$transition_{Match(5),Insertion(5)} = 3/4$

$transition_{Match(5),Match(6)} = 1/4$

$transition_{Match(5),Deletion(6)} = 0$

Emission Probabilities of Profile HMM



symbols emitted from M_2:

C, F, C, D

$emission_{Match(2)}(A) = 0$

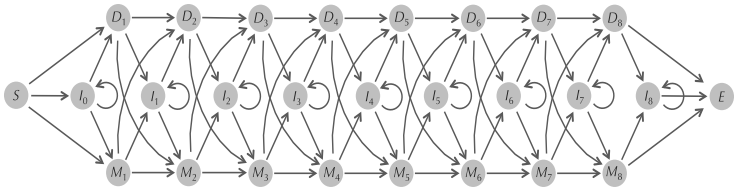
$emission_{Match(2)}(C) = 2/4$

$emission_{Match(2)}(D) = 1/4$

$emission_{Match(2)}(E) = 0$

$emission_{Match(2)}(F) = 1/4$

Forbidden Transitions



	S	I_0	M_1	D_1	I_1	M_2	D_2	I_2	M_3	D_3	I_3	M_4	D_4	I_4	M_5	D_5	I_5	M_6	D_6	I_6	M_7	D_7	I_7	M_8	D_8	I_8	E
S			1																								
I_0																											
M_1						.8	.2																				
D_1																											
I_1																											
M_2								1																			
D_2																											
I_2																											
M_3											1																
D_3																											
I_3											1																
M_4															.8	.2											
D_4																											
I_4																											
M_5																	.25	.75									
D_5																	.33	.67									
I_5																		1									
M_6																					.8	.2					
D_6																											
I_6																											
M_7																								1			
D_7																											
I_7																								1			
M_8																											1
D_8																											
I_8																											
E																											

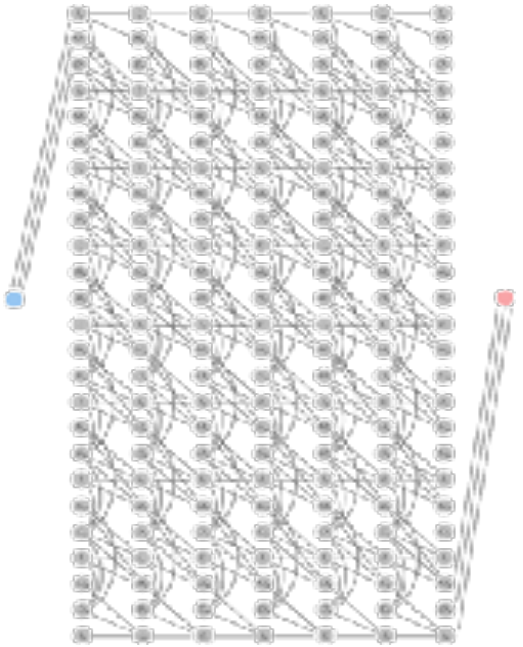
Gray cells: edges in the HMM diagram.

Clear cells: *forbidden* transitions.

Don't forget pseudocounts: *HMM*(*Alignment*,θ,σ)

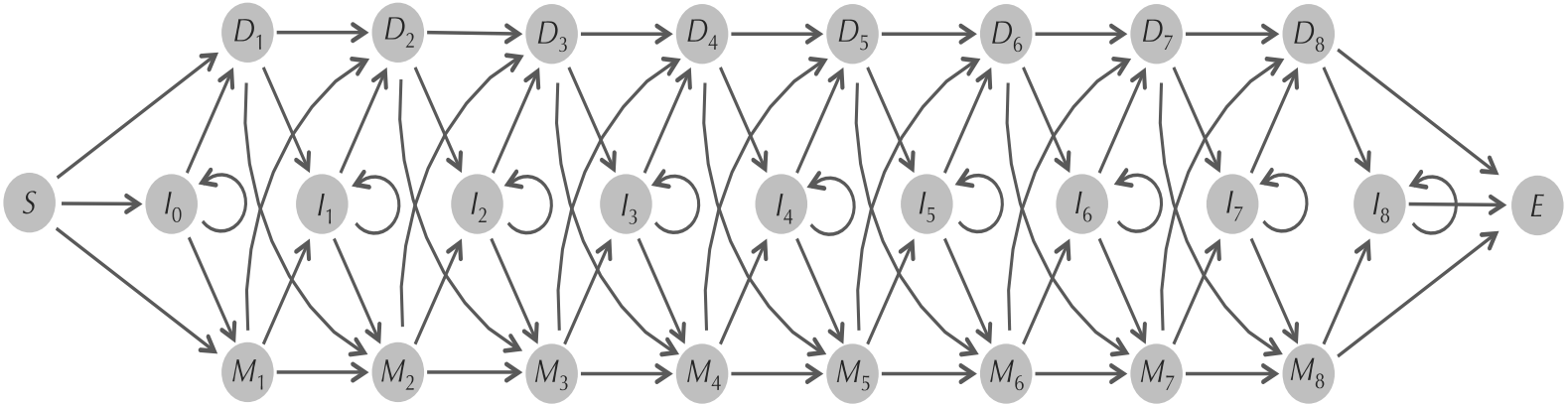
Hidden Markov Models

- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model
- Decoding Problem
- The Viterbi Algorithm
- Profile HMMs for Sequence Alignment
- Classifying proteins with profile HMMs



- Viterbi Learning
- Soft Decoding Problem
- Baum-Welch Learning

Aligning a Protein Against a Profile HMM

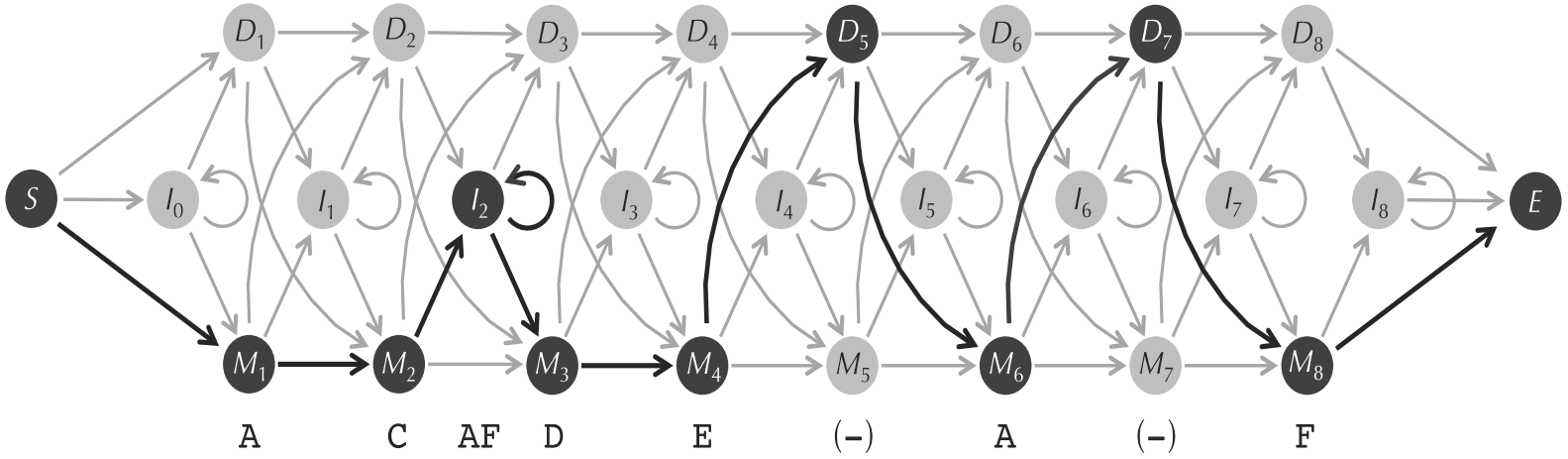


Alignment

A	C	–	–	D	E	F	A	C	A	D	F
A	F	–	–	D	A	–	–	–	C	C	F
A	–	–	–	–	E	F	D	–	F	D	C
A	C	–	–	A	E	F	–	–	A	–	C
A	D	–	–	D	E	F	A	A	A	D	F

Protein **ACAFDEAF**

Aligning a Protein Against a Profile HMM



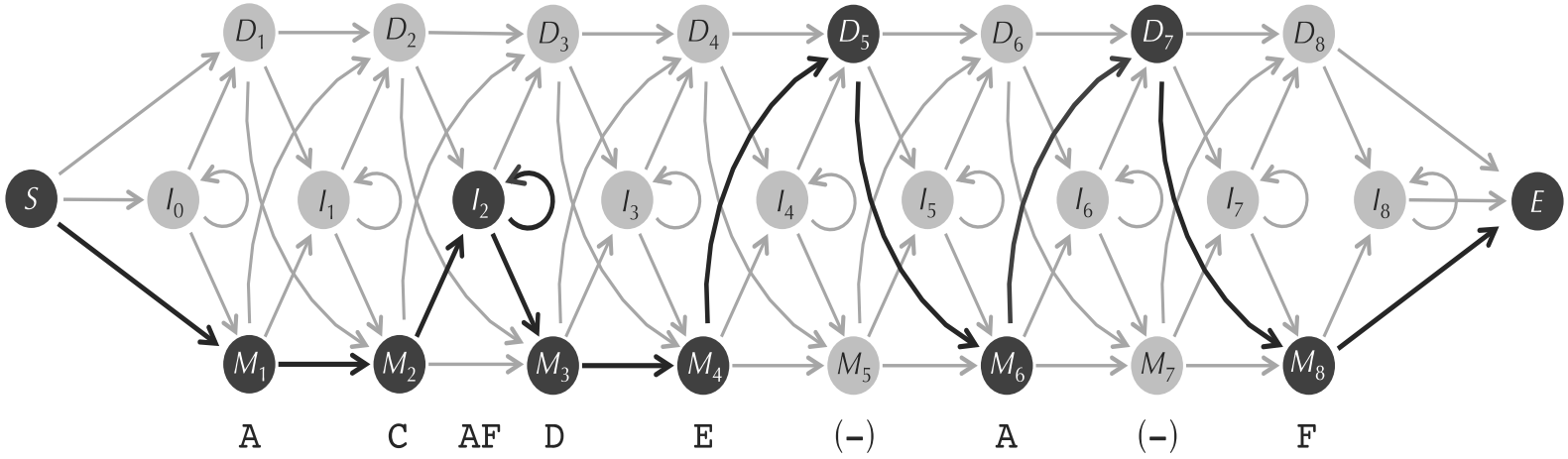
Alignment

A	C	– –	D	E	F	AC	A	D	F
A	F	– –	D	A	–	– –	C	C	F
A	–	– –	–	E	F	D –	F	D	C
A	C	– –	A	E	F	– –	A	–	C
A	D	– –	D	E	F	AA	A	D	F

Protein **ACAFDEAF**

Apply Viterbi algorithm to find optimal hidden path!

Aligning a Protein Against a Profile HMM

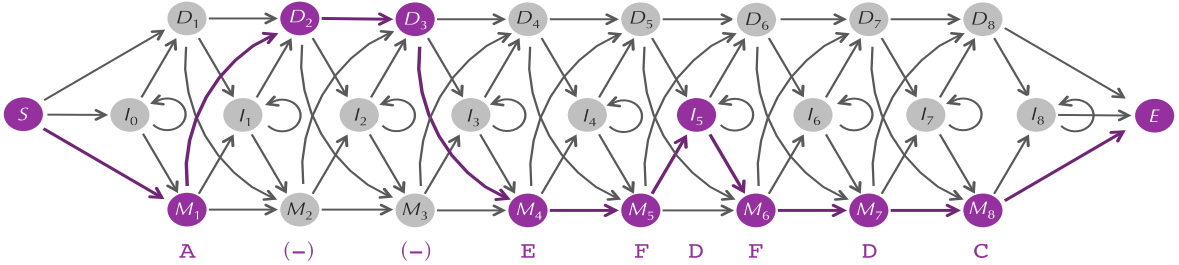


Alignment

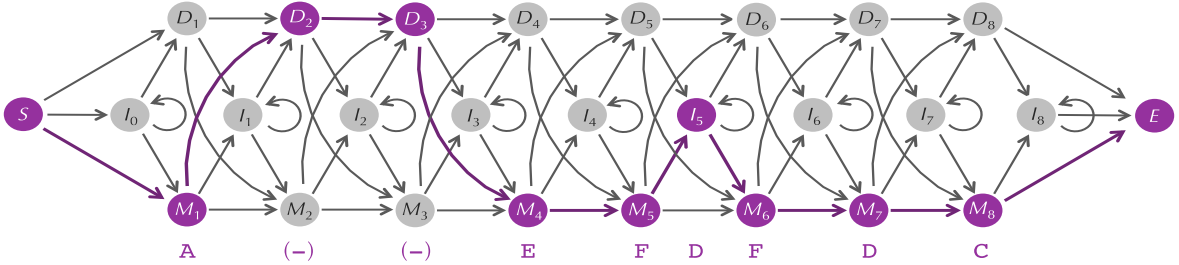
A	C	– –	D	E	F	AC	A	D	F
A	F	– –	D	A	–	– –	C	C	F
A	–	– –	–	E	F	D –	F	D	C
A	C	– –	A	E	F	– –	A	–	C
A	D	– –	D	E	F	AA	A	D	F
Protein									
A	C	AF	D	E	–	– –	A	–	F

Apply Viterbi algorithm to find optimal hidden path!

Profile HMM diagram

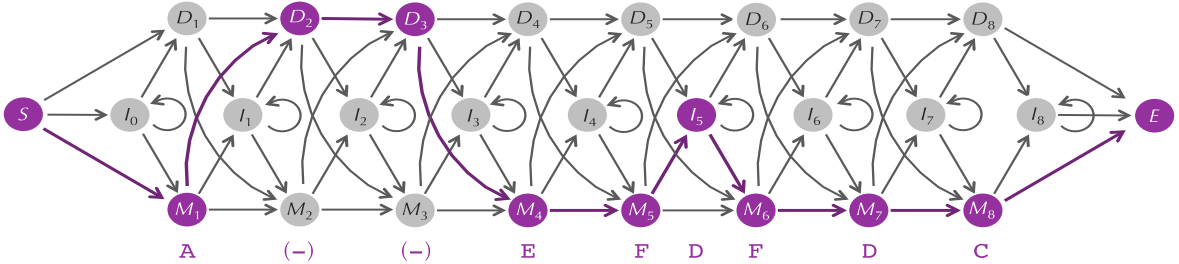


Profile HMM diagram

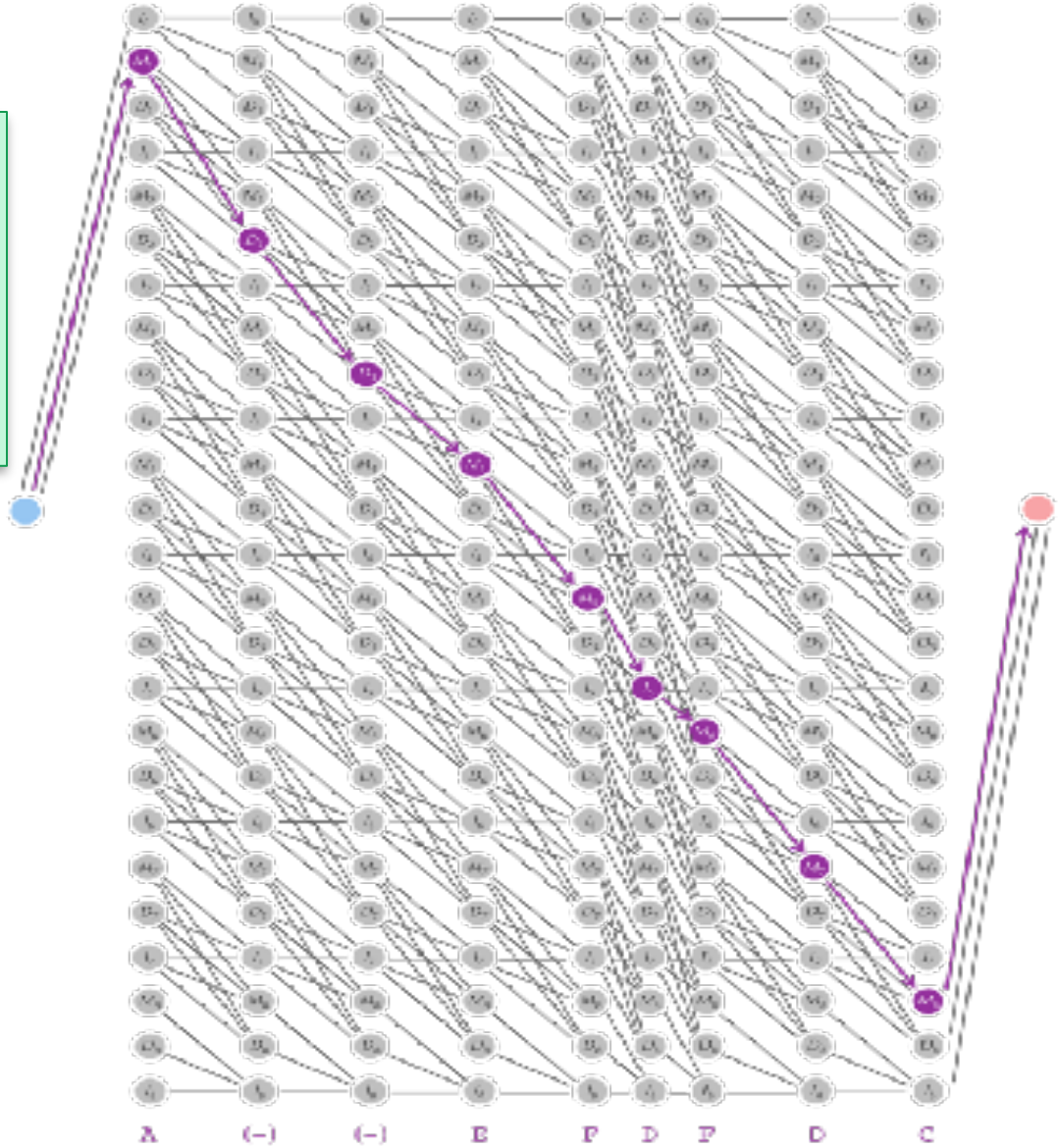


How many rows and columns does the Viterbi graph of this profile HMM have?

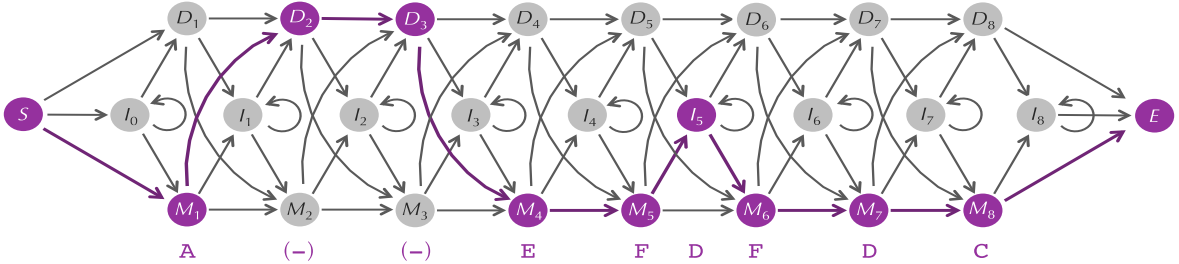
Profile HMM diagram



Viterbi graph of profile HMM:
#columns=
#visited states



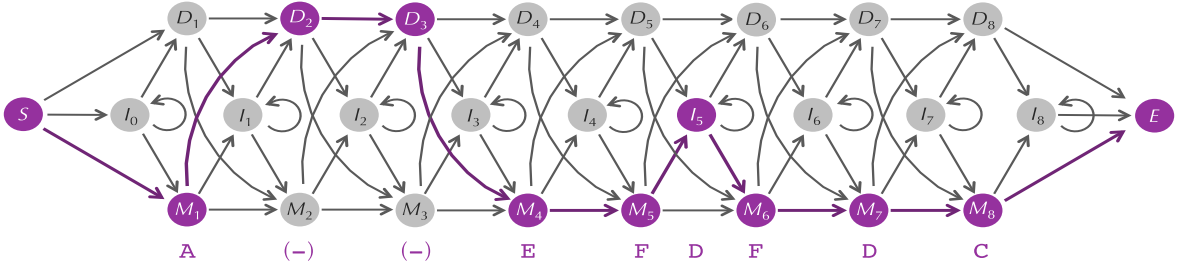
Profile HMM diagram



Viterbi graph of profile HMM:
#columns=
#visited states

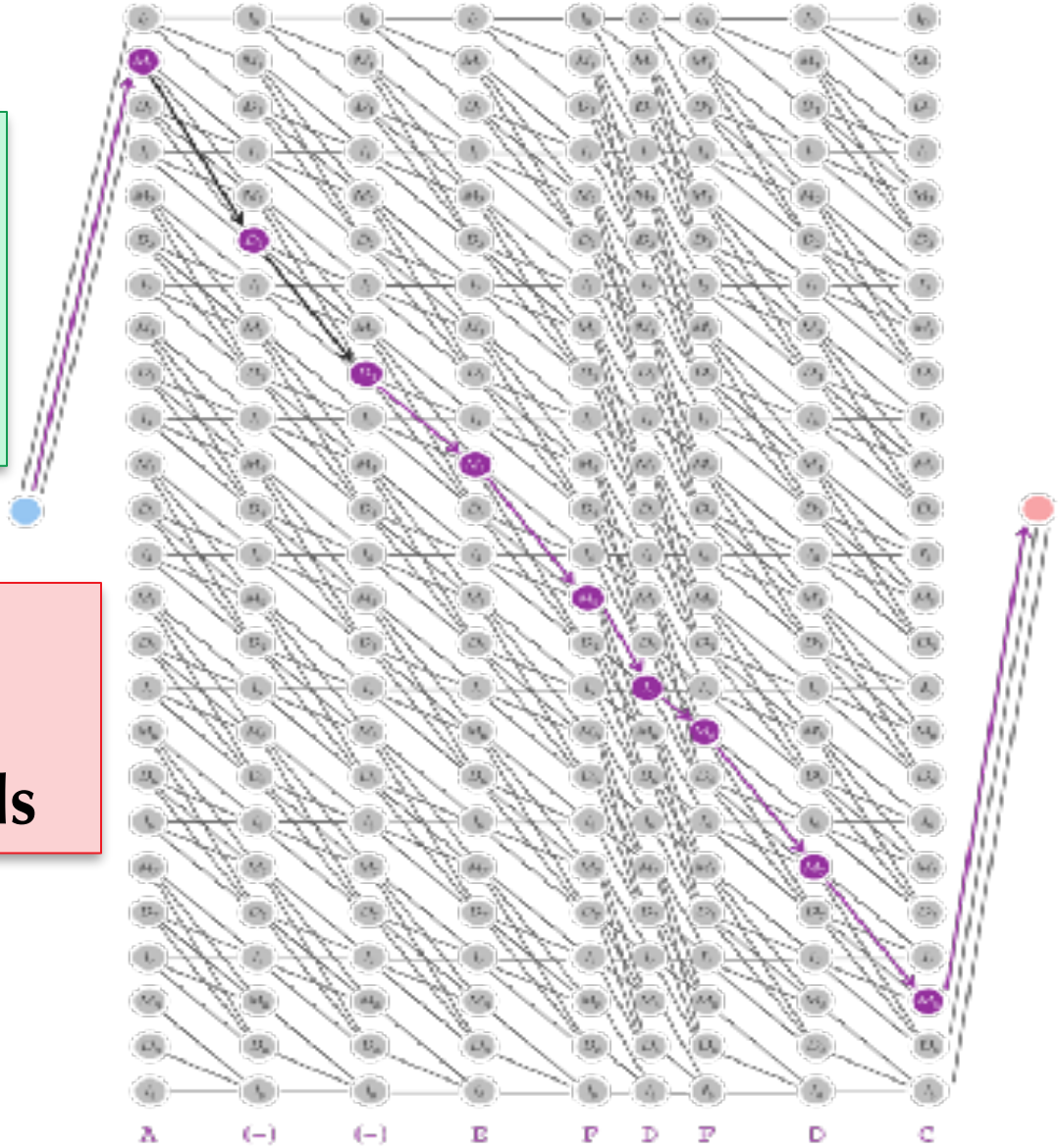
What is wrong with this Viterbi graph?

Profile HMM diagram

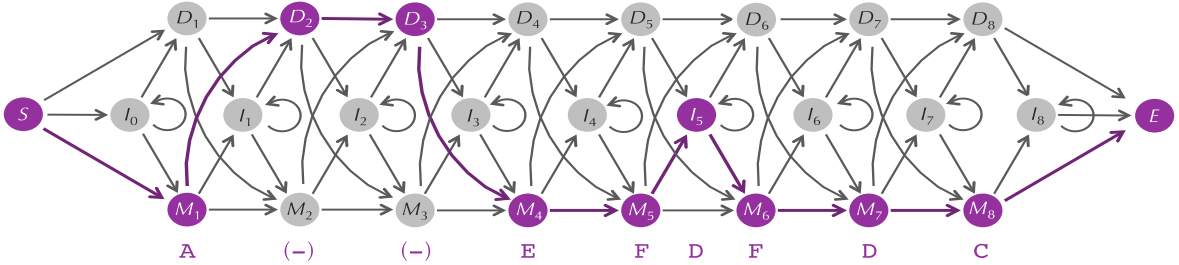


Viterbi graph of profile HMM:
#columns=
#visited states

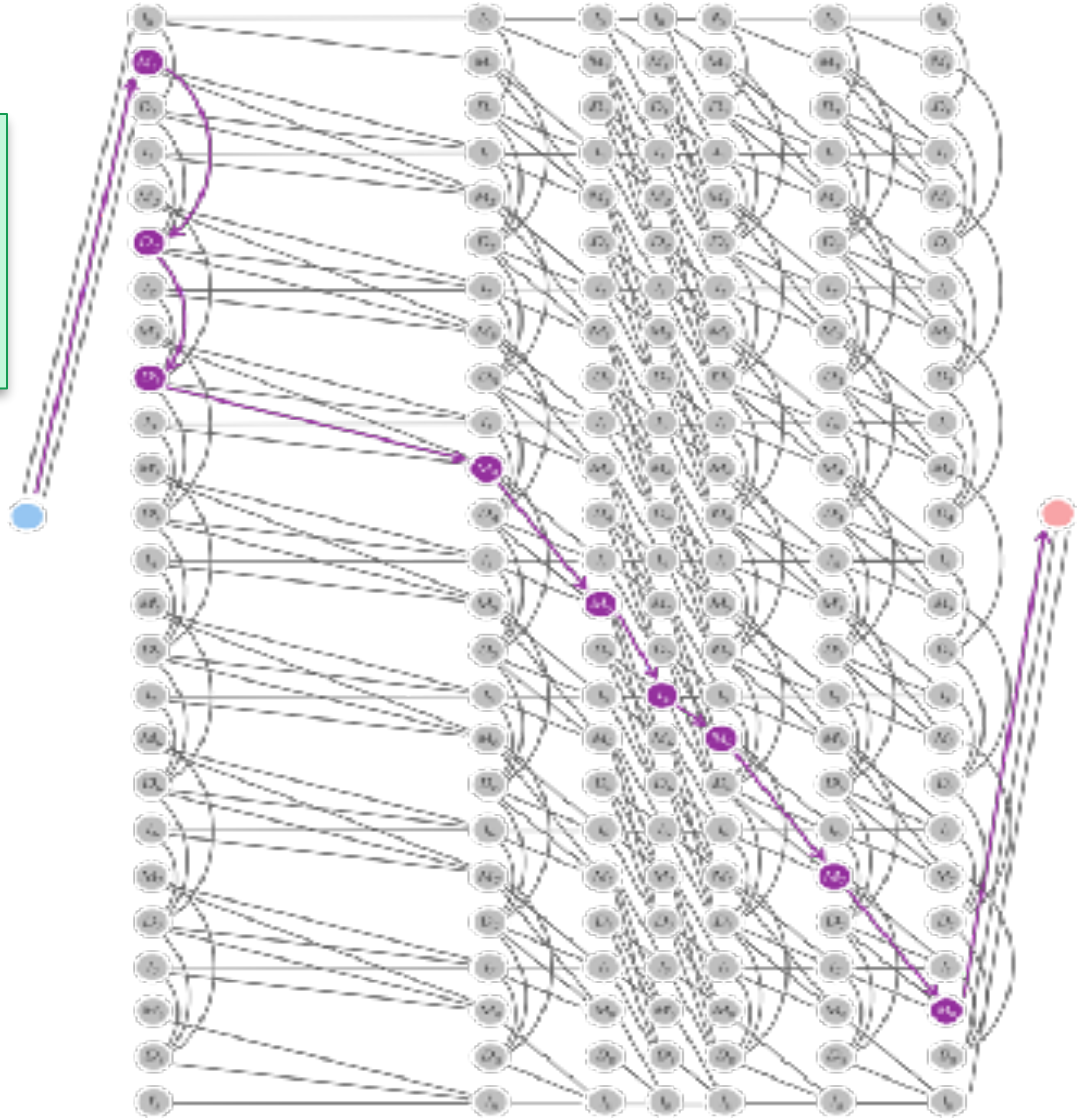
By definition,
#columns =
#emitted symbols



Profile HMM diagram

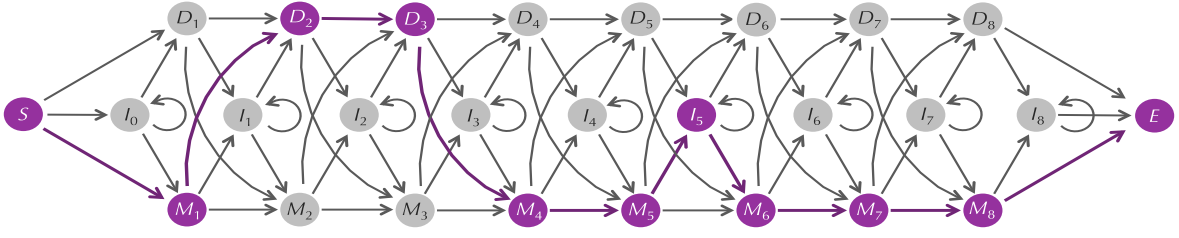


Nearly correct Viterbi graph of profile HMM:



Vertical edges enter "silent" deletion states

Profile HMM diagram

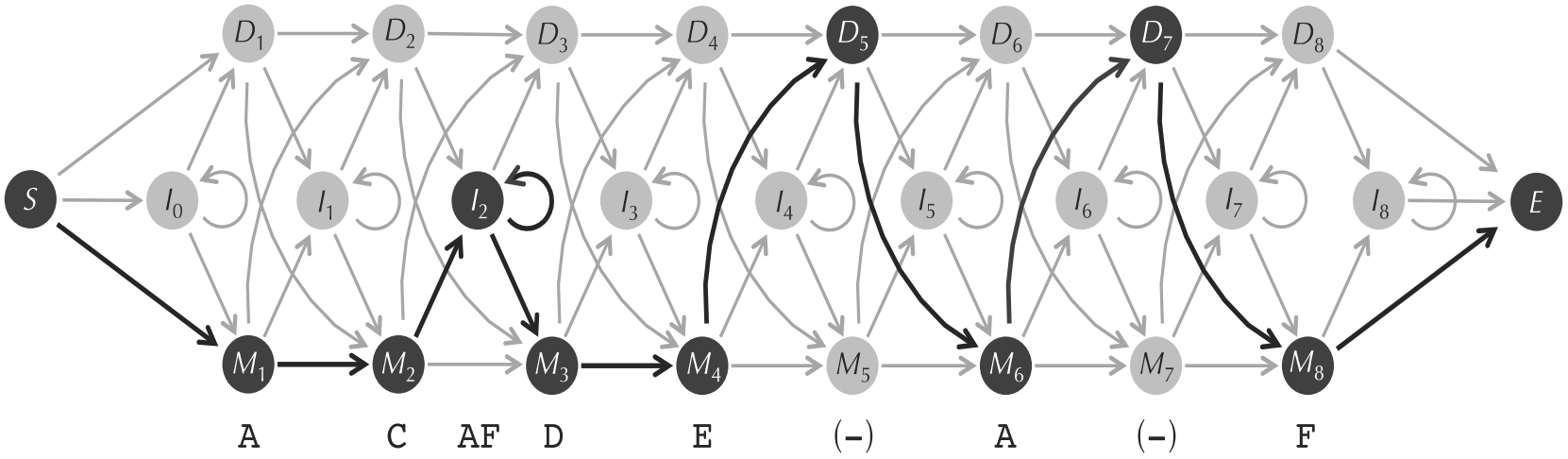


A (–) (–) E F D F D C

Correct Viterbi graph of profile HMM:

Adding 0-th column that contains only silent states

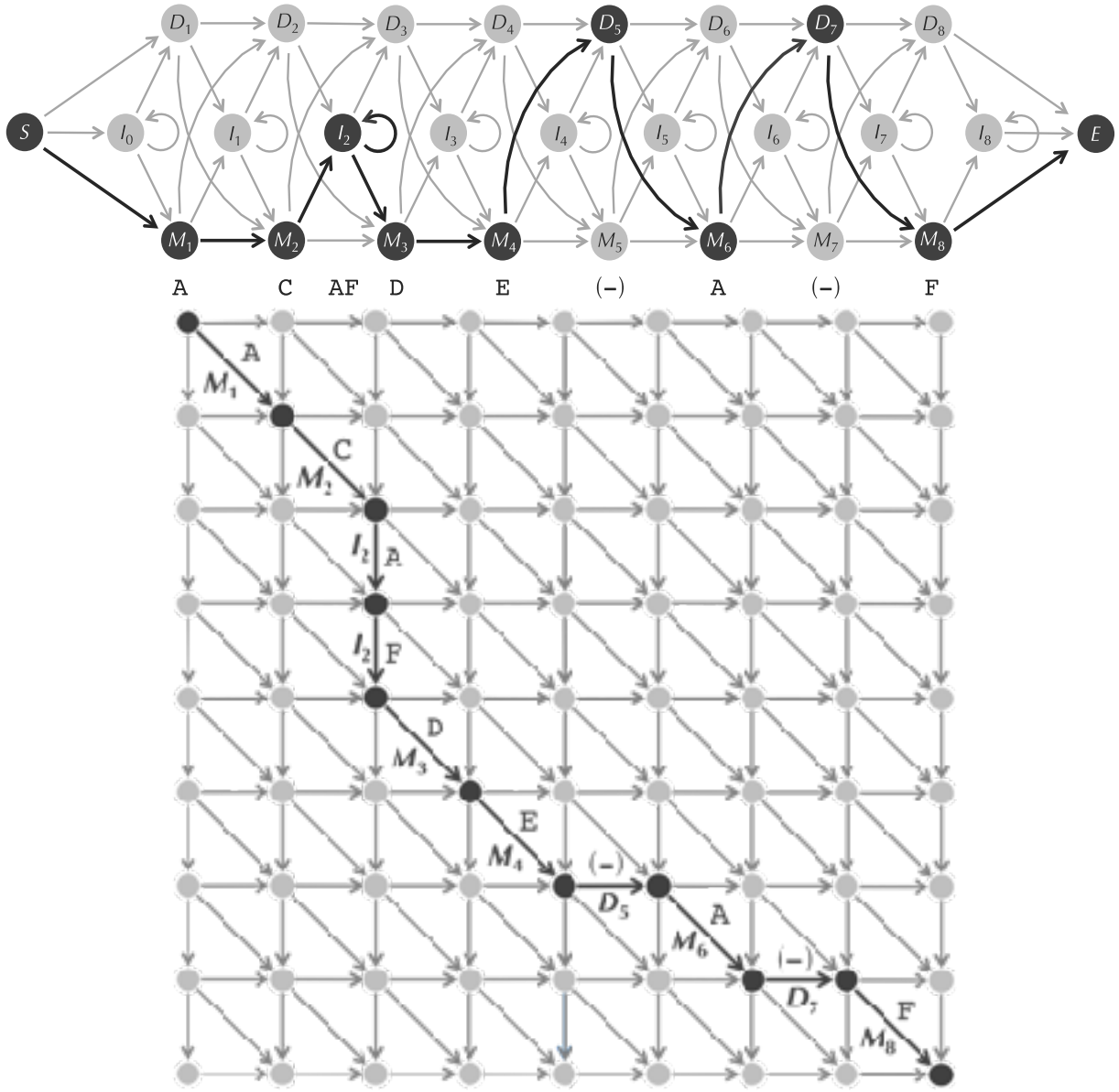
Alignment with a Profile HMM



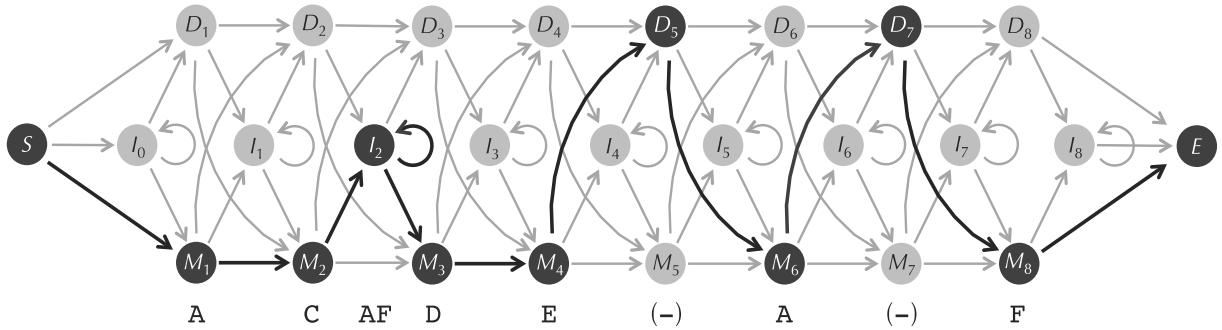
Sequence Alignment with Profile HMM Problem: *Align a new sequence to a family of aligned sequences using a profile HMM.*

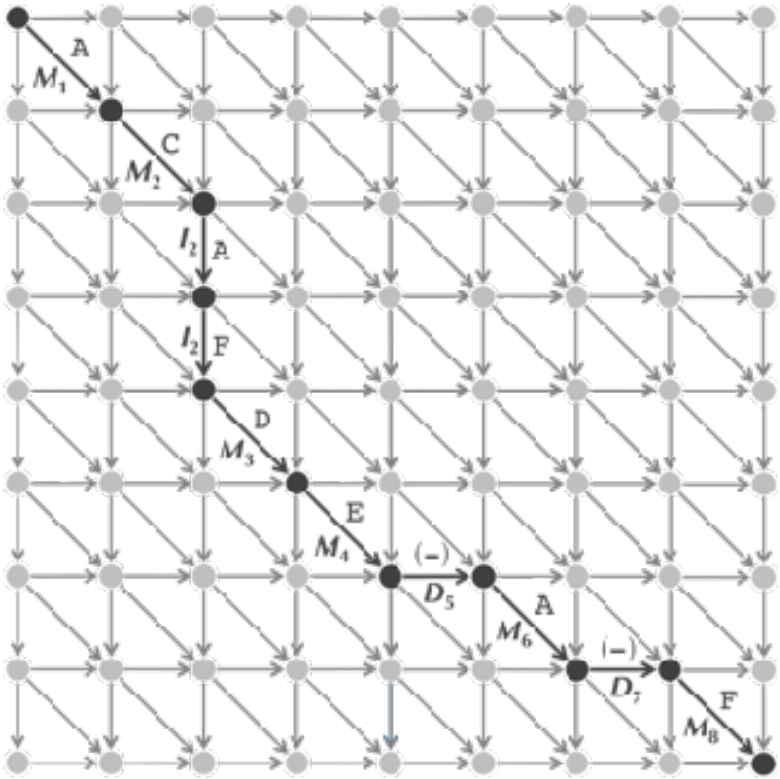
- **Input:** A multiple alignment *Alignment*, a string *Text*, a threshold θ (maximum fraction of insertions per column), and a pseudocount σ.
- **Output:** An optimal hidden path emitting *Text* in the profile HMM *HMM*(*Alignment*, θ, σ).

HMM vs Global Pairwise Alignment



HMM vs Global Pairwise Alignment



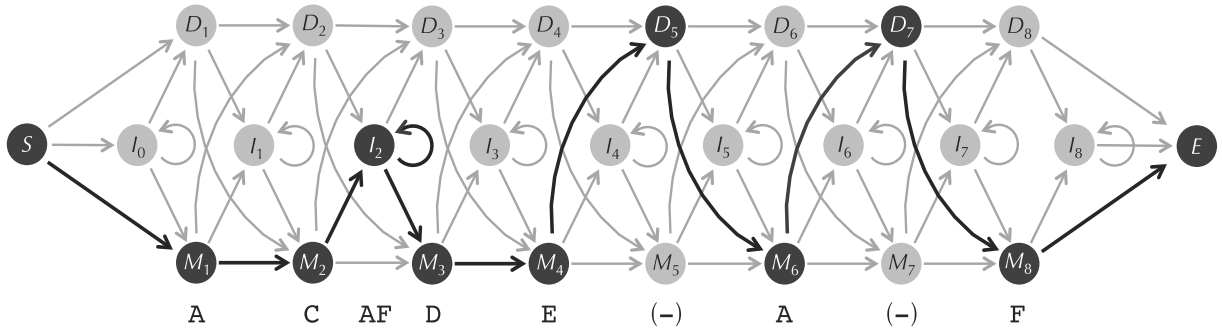


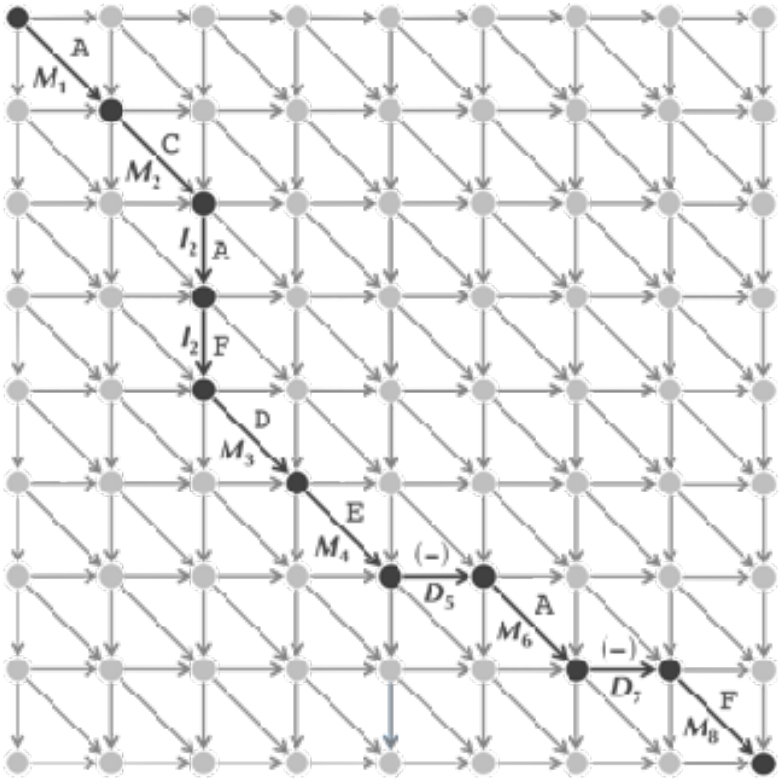
$$s_{M(j),i} = \max \begin{cases} s_{I(j-1),i-1} * weight(I(j\text{-}1),M(j),i\text{-}1) \\ s_{D(j-1),i-1} * weight(D(j\text{-}1),\ M(j),i\text{-}1) \\ s_{M(j-1),i-1} * weight(M(j\text{-}1),\ M(j),i\text{-}1) \end{cases}$$

$$s_{i,\,j} = \max \begin{cases} s_{i\text{-}1,\,j} + score(v_i,\ \text{-}) \\ s_{i,\,j\text{-}1} + score(\text{-},w_j) \\ s_{i\text{-}1,\,j\text{-}1} + score(v_i,w_j) \end{cases}$$

The choice of alignment path is now based on varying transition and emission probabilities!

HMM vs Global Pairwise Alignment



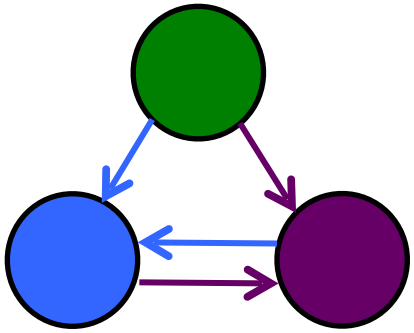


$$s_{M(j),i} = \max \begin{cases} s_{I(j-1),i-1} * weight_{i-1}(I(j\text{-}1), M(j)) \\ s_{D(j-1),i-1} * weight_{i-1}(D(j\text{-}1),\ M(j)) \\ s_{M(j-1),i-1} * weight_{i-1}(M(j\text{-}1),\ M(j)) \end{cases}$$

Individual scoring parameters for each edge in the alignment graph capture subtle similarities that evade traditional alignments.

Hidden Markov Models

- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model
- Decoding Problem
- The Viterbi Algorithm
- Profile HMMs for Sequence Alignment
- Classifying proteins with profile HMMs
- Viterbi Learning



- Soft Decoding Problem
- Baum-Welch Learning

HMM Parameter Estimation

- Thus far, we have assumed that the transition and emission probabilities are known.
- Imagine that you only know that the crooked dealer is using two coins and observe:

HHTHHHTHHHTTTTHTTTTH

What are the biases of the coins and how often does the dealer switch coins?

Can we develop an algorithm for parameter estimation for an *arbitrary* HMM?

If Dealer Reveals the Hidden Path...

HMM Parameter Estimation Problem: *Find optimal parameters explaining the emitted string and the hidden path.*

- **Input:** A string $x = x_1 \ldots x_n$ emitted by a k-state HMM with unknown transition and emission probabilities following a ***known*** hidden path $\pi = \pi_1 \ldots \pi_n$.
- **Output**: *Transition* and *Emission* matrices that maximize $\Pr(x, \pi)$ over all possible matrices of transition and emission probabilities.

HHTHHHTHHHTTTTHTTTTH

BFFBFFFBBFFBFFBBBBFF

If the Hidden Path is Known...

- $T_{l,k}$: #transitions from state l to state k in path π.

$transition_{l,k}$=
#transitions from state l to state k / # all transitions from l
$= T_{l,k} / \sum_{\text{all states } j} T_{l,j}$

HHTHHHTHHHTTTTHTTTTH

BFFBFFFBBFFBFFBBBBFF

$transition_{B,F} = 5/9$

If the Hidden Path is Known...

- $T_{l,k}$: # transitions from state l to state k in path π.
- $E_k(b)$: # times symbol b is emitted when path π is in state k.

$$transition_{l,k} = \frac{\#\text{transitions from state } l \text{ to state } k}{\#\text{ all transitions from } l} = \frac{T_{l,k}}{\sum_{\text{all states } j} T_{l,j}}$$

$$emission_k(b) = \frac{\#\text{times symbol } b \text{ is emitted in state } k}{\#\text{ all symbols emitted in state } k} = \frac{E_k(b)}{\sum_{\text{all symbols c in the alphabet}} E_k(c)}$$

HHTHHHTHHHTTTTHTTTH

$emission_F(T) = 6/11$

BFFBFFFBBFFBFFBBBFF

$T_{B,F} = 5$

From emitted string to hidden path&*Parameters*

(emitted string, ???, *Parameters*) → hidden path

Viterbi algorithm for Decoding Problem

(emitted string, hidden path, ???) → *Parameters*

algorithm for HMM Parameter Estimation Problem

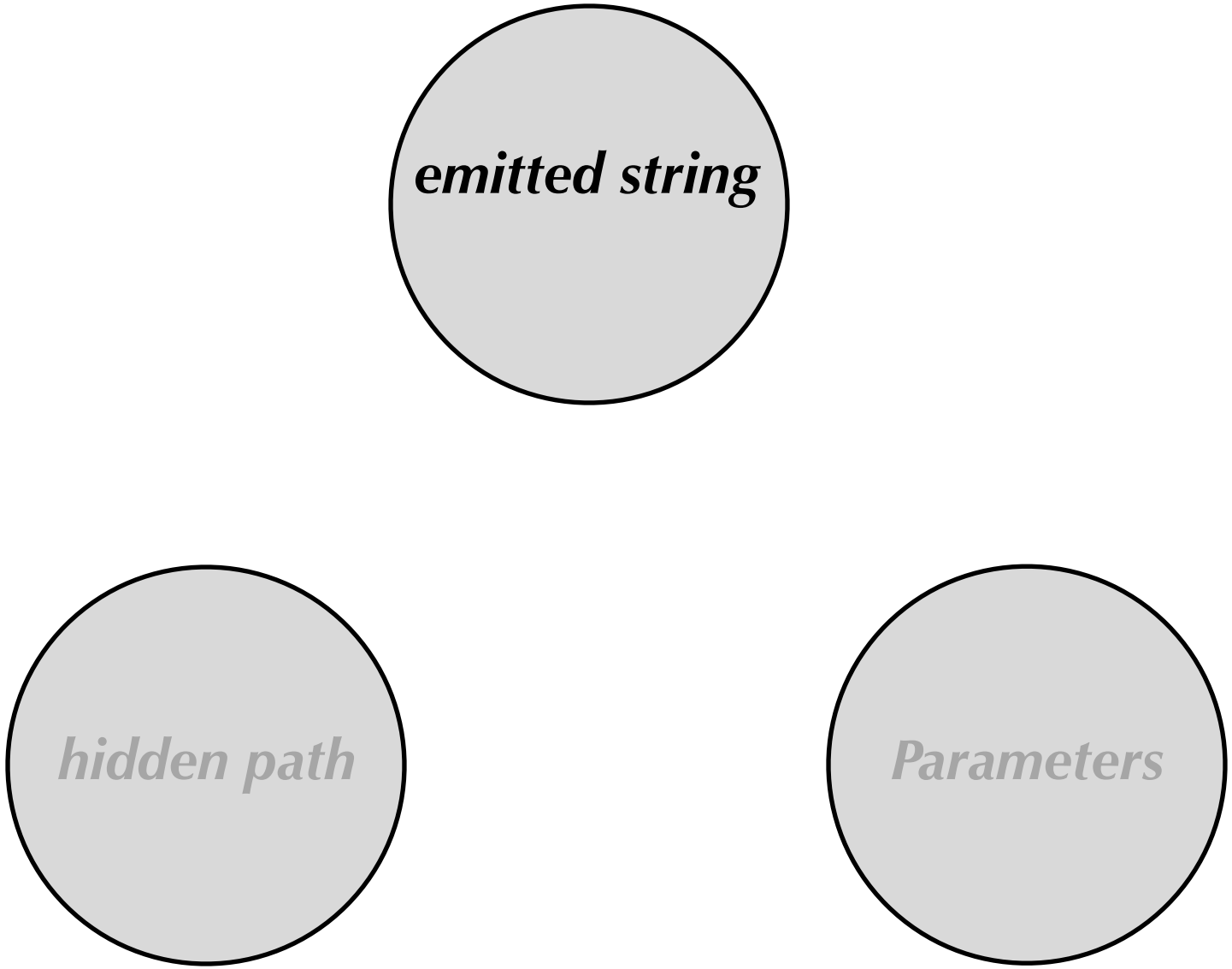
(emitted string, ???, ???) → hidden path & *Parameters*

When BOTH *HiddenPath* and *Parameters* Are Unknown

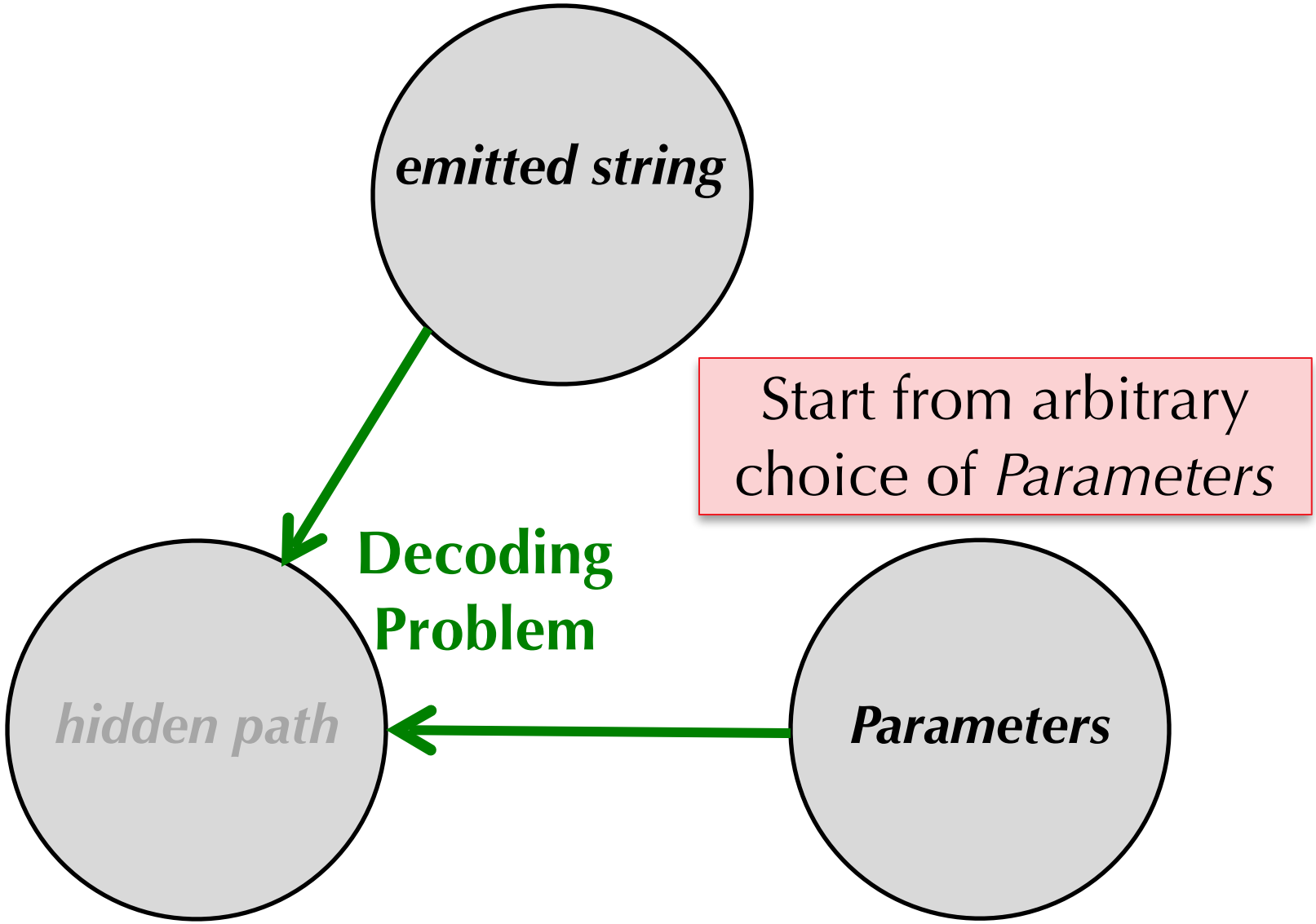
HMM Parameter Learning Problem. *Estimate the parameters of an HMM explaining an emitted string.*

- **Input:** A string $x = x_1 \ldots x_n$ emitted by a k-state HMM with unknown transition and emission probabilities.
- **Output**: Matrices *Transition* and *Emission* that maximize $\Pr(x, \pi)$ over all possible transition and emission matrices and over all hidden paths π.

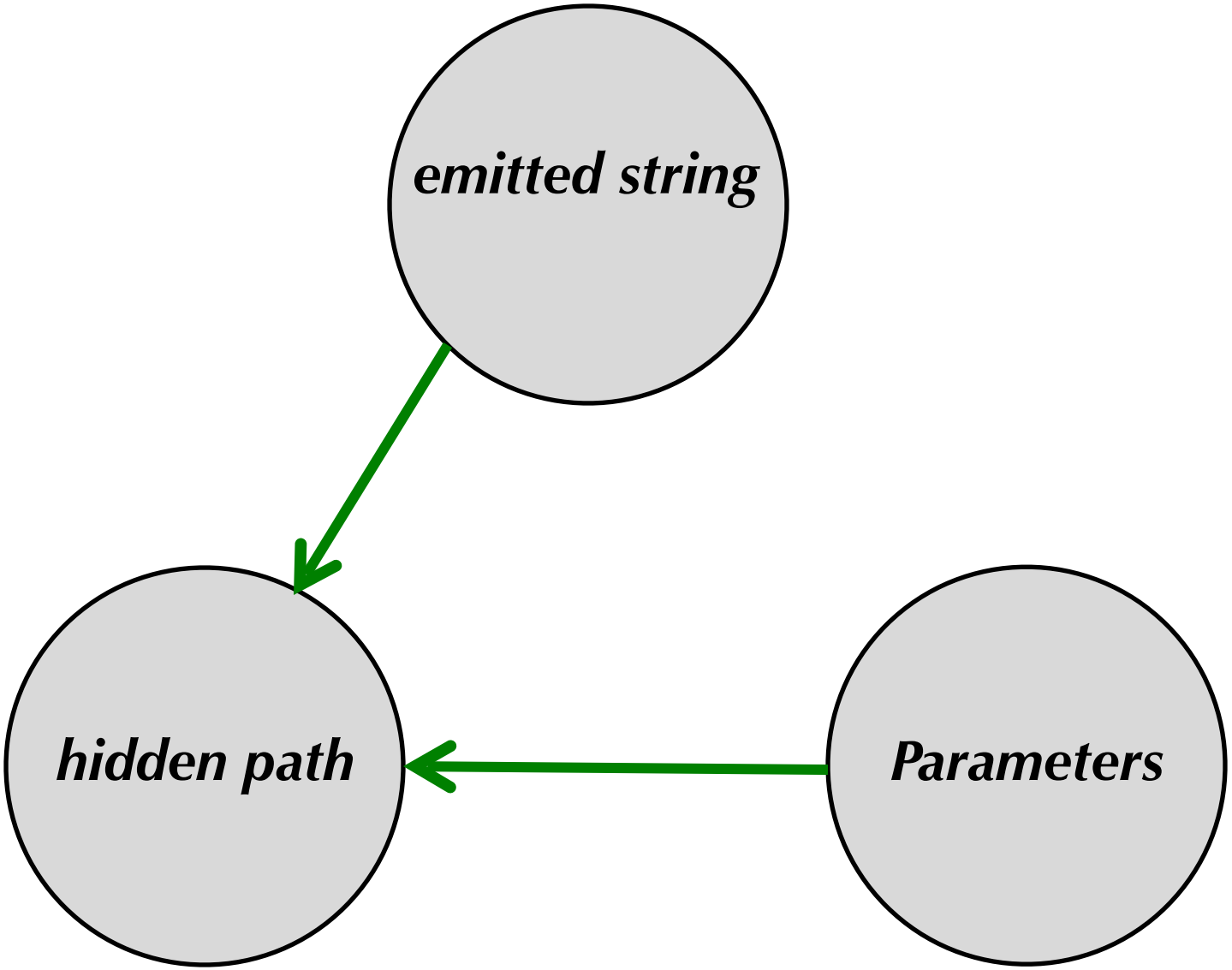
Reconstructing *HiddenPath* AND *Parameters*



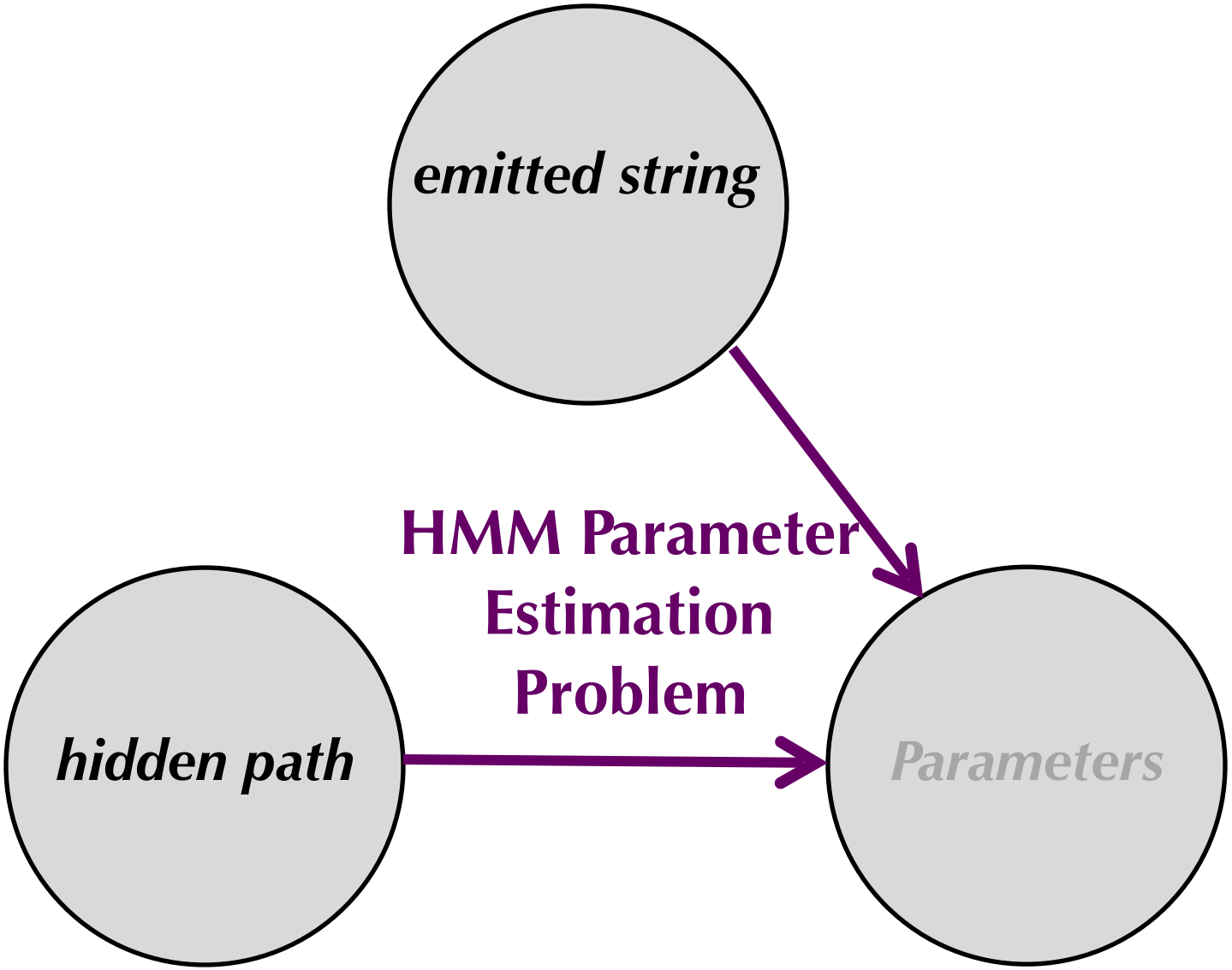
Reconstructing *HiddenPath* AND *Parameters*



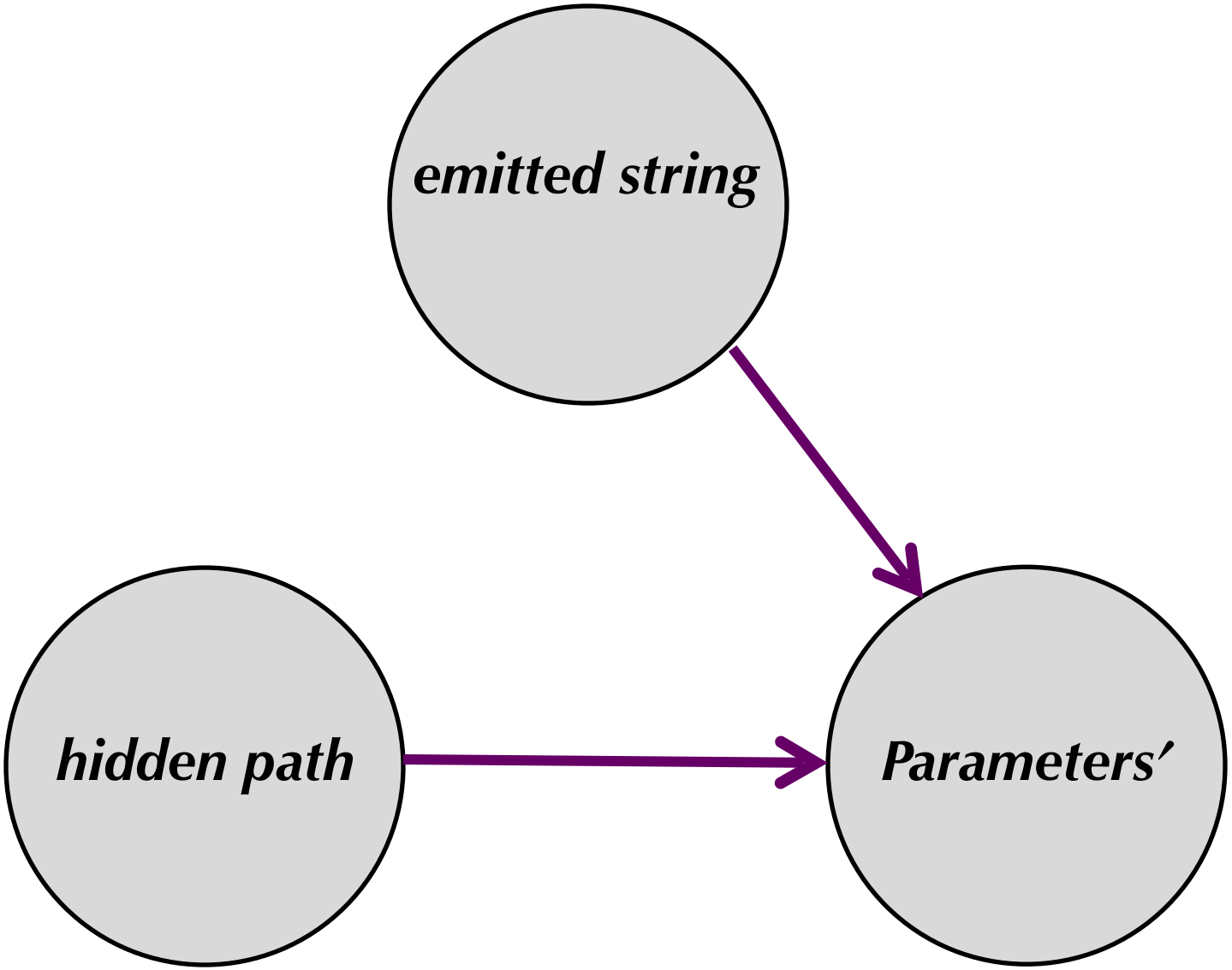
Reconstructing *HiddenPath* AND *Parameters*



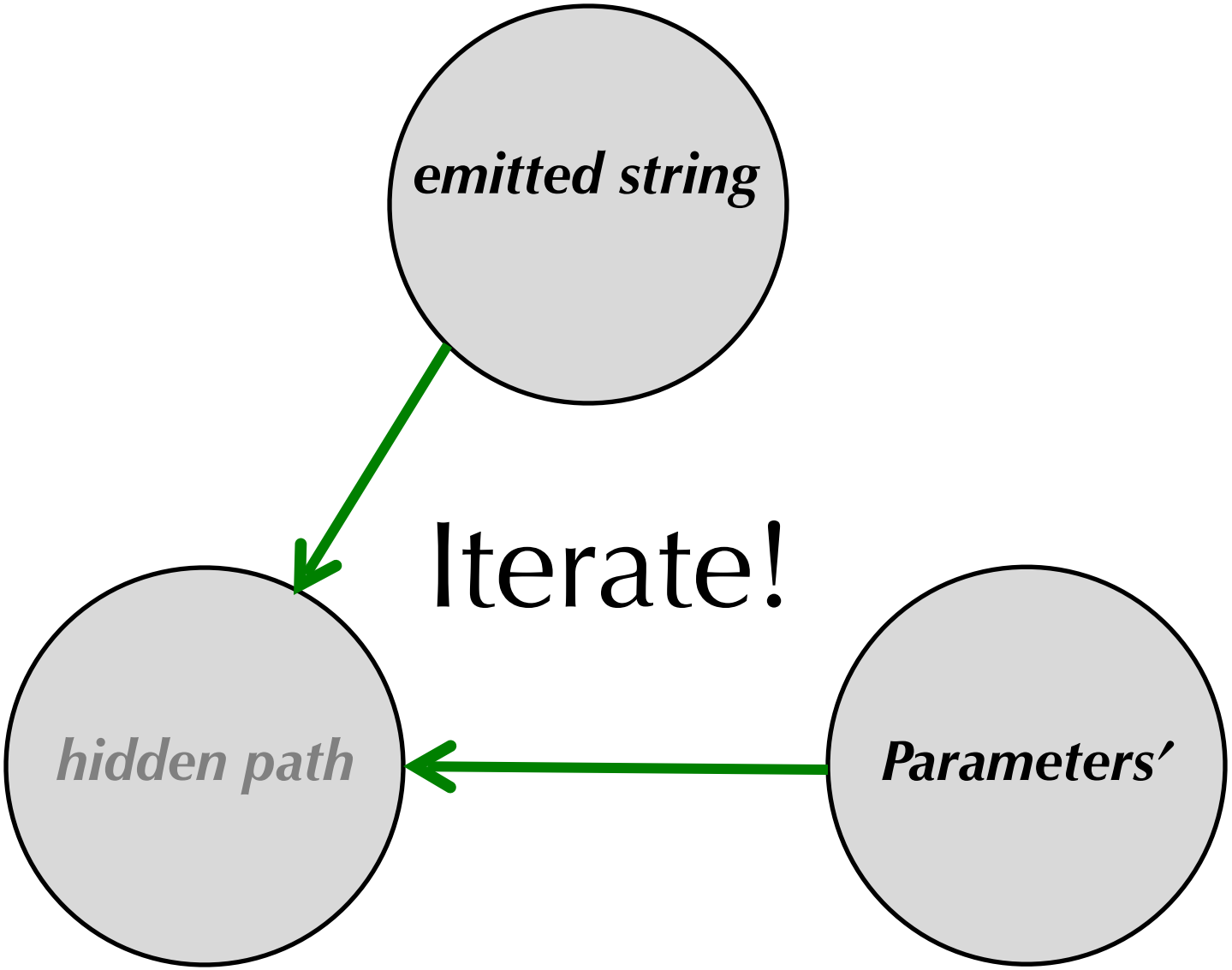
Reconstructing *HiddenPath* AND *Parameters*



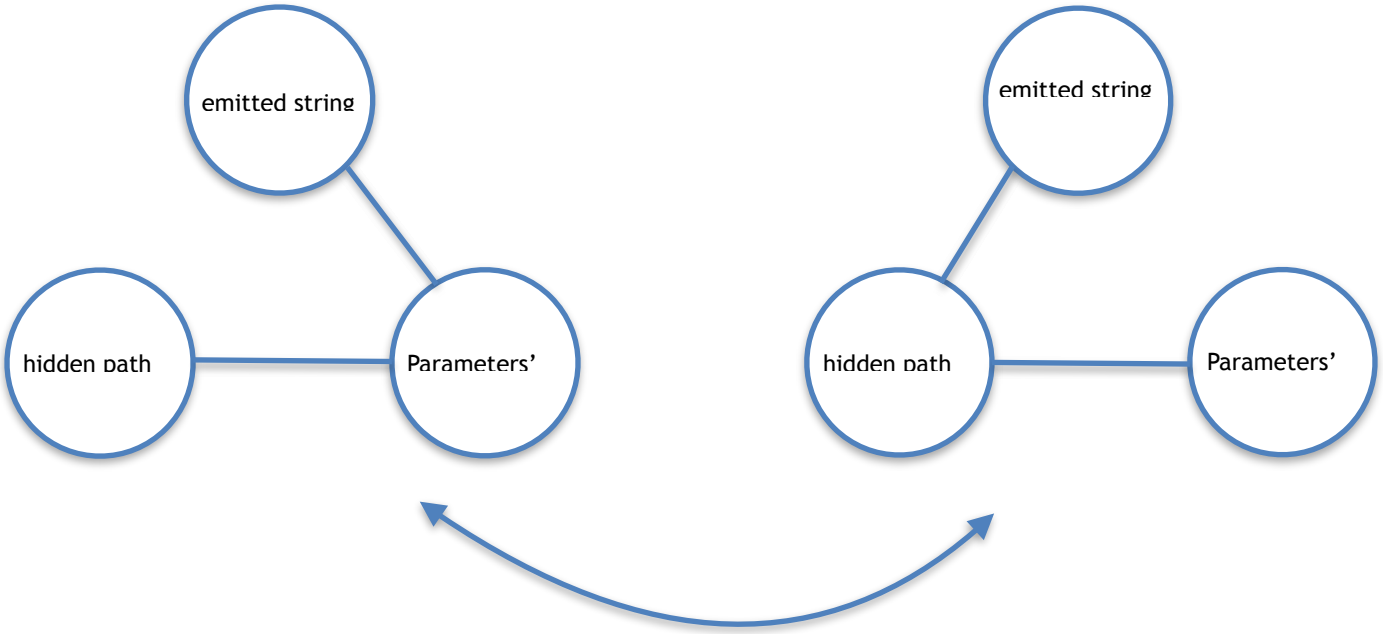
Reconstructing *HiddenPath* AND *Parameters*



Viterbi Learning



Viterbi Learning



- Termination practices
  - # iterations > threshold, or
  - Δ *Pr(x, Π) < threshold*
- initial *parameters* guess is arbitrary … *repeat and save best*

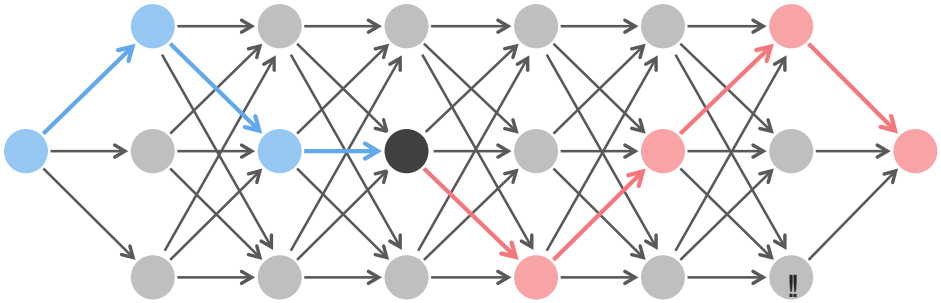
Changing the Question

- The Viterbi algorithm gives a "yes" or "no" answer to the question: *"Was the HMM in state k at time i given that it emitted string x?"*

This question fails to account for how certain we are in the "yes"/"no" answer. How can we change this **hard** answer into a **soft** one?

Hidden Markov Models

- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model
- Decoding Problem
- The Viterbi Algorithm
- Profile HMMs for Sequence Alignment
- Classifying proteins with profile HMMs
- Viterbi Learning
- Soft Decoding Problem



- Baum-Welch Learning

Changing the Question

- The Viterbi algorithm gives a "yes" or "no" answer to the question: *"Was the HMM in state k at time i given that it emitted string x?"*

This question fails to account for how certain we are in the "yes"/"no" answer. How can we change this **hard** question into a **soft** one?

What Is $\Pr(\pi_i = k, x)$?

$\Pr(\pi_i = k, x)$: the *unconditional* probability $\Pr(\pi_i = k, x)$ that a hidden path will pass through state k at time i *and* emit x.

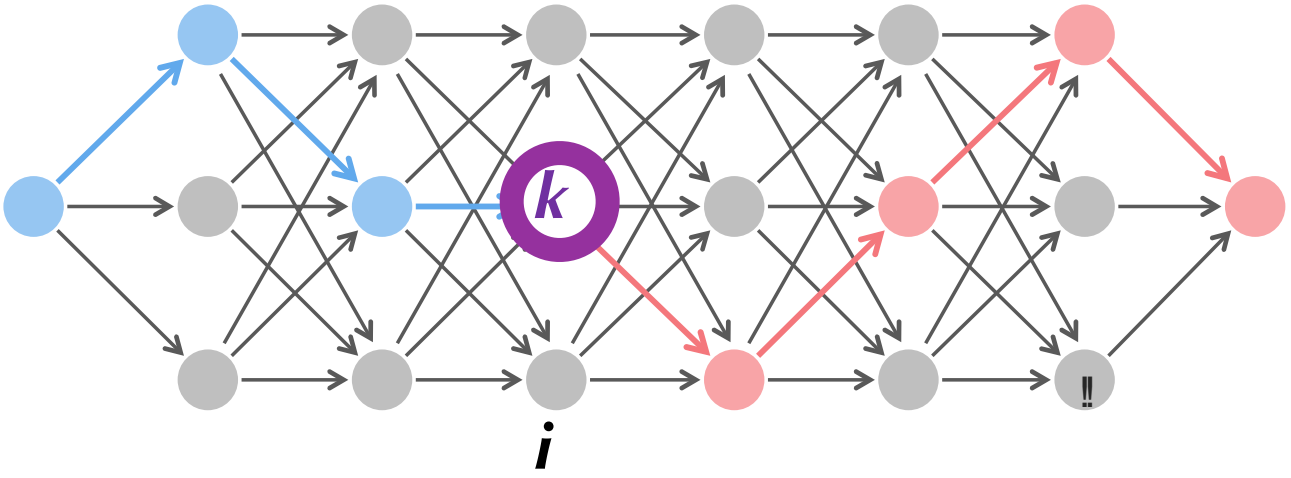
What is the probability that the dealer was using the Fair coin at the 5th flip ***given*** that he generated a sequence of flips **HH**T**H**T**HHH**TT?

$\Pr(\pi_i = k, x)$:

Total Product Weight of All Paths Through

$\Pr(\pi_i = k, x)$: the *unconditional* probability that a hidden path will pass through state k at time i *and* emit x.

$$\Pr(\pi_i = k, x) = \sum_{\text{all paths } \pi \text{ with } \pi i = k} \Pr(x, \pi)$$



What Is $\Pr(\pi_i = k | x)$?

$\Pr(\pi_i = k | x)$: the *conditional* probability that the HMM was in state k at time i *given* that it emitted string x.

What is the probability that the dealer was using the Fair coin at the 5th flip *given* that he generated a sequence of flips **HH**T**H**T**HHH**TT?

What Is $\Pr(\pi_i = k|x)$?

$\Pr(\pi_i = k|x)$: the *conditional* probability that the HMM was in state k at time i *given* that it emitted string x.

What is the probability that the dealer was using the Fair coin at the 5th flip *given* that he generated a sequence of flips **HH**T**H**T**HHH**TT?

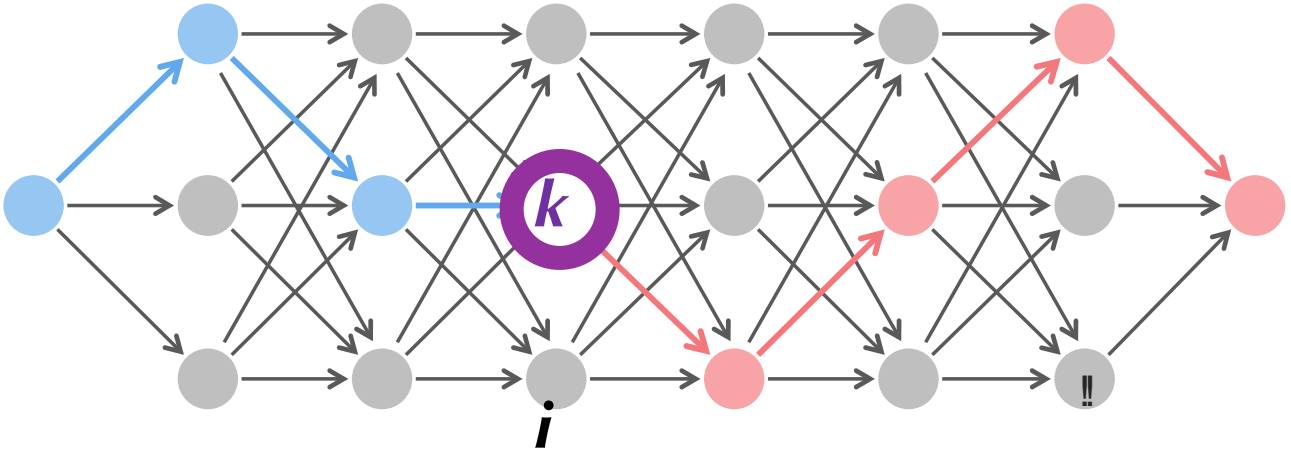
Compare with:

$\Pr(\pi_i = k, x)$: the *unconditional* probability that a hidden path will pass through state k at time i *and* emit x.

What is the probability that the dealer will generate a sequence of flips **HH**T**H**T**HHH**TT?
and, while using the Fair coin at the 5th flip?

What Is $\Pr(\pi_i = k|x)$?

$\Pr(\pi_i = k|x)$: the *conditional* probability that the HMM was in state k at time i *given* that it emitted string x.



$\Pr(\pi_i = k|x)$: the fraction of the product weight of paths visiting k over the weight of all paths:

$$\Pr(\pi_i = k|x) = \Pr(\pi_i = k, x) / \Pr(x)$$

$$= \sum_{\text{all paths } \pi \text{ with } \pi i = k} \Pr(x, \pi) / \sum_{\text{all paths } \pi} \Pr(x, \pi)$$

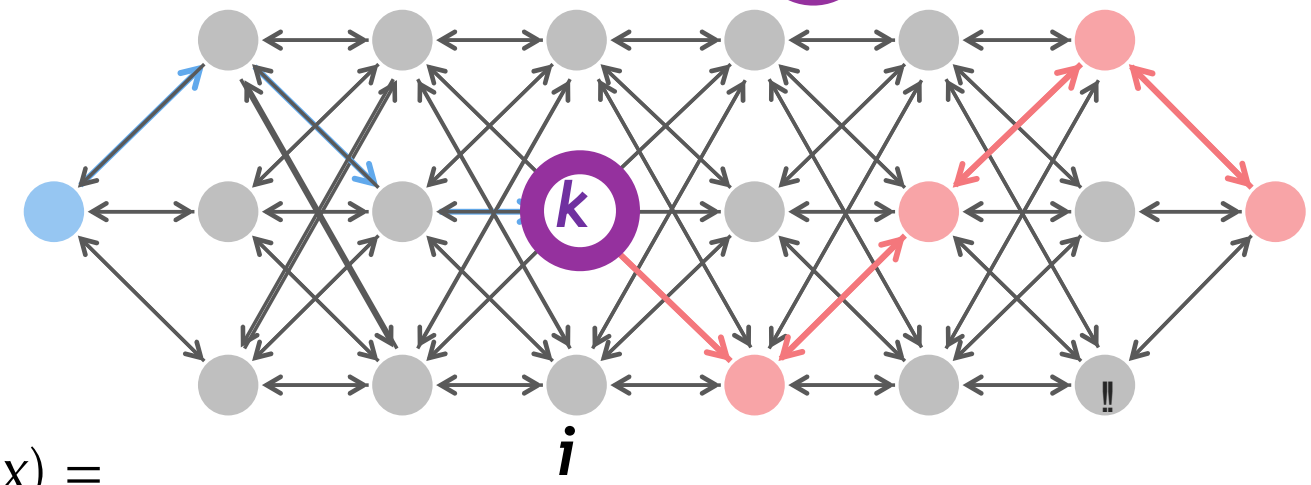
Soft Decoding Problem

Soft Decoding Problem: *Find the probability that an HMM was in a particular state at a particular moment, given its output.*

- **Input:** A string $x = x_1 \ldots x_n$ emitted by an HMM $(\sum, States, Transition, Emission)$.
- **Output**: The conditional probability $\Pr(\pi_i = k|x)$ that the HMM was in state k at step i, given x.

Computing $\Pr(\pi_i=k, x)$

- $\Pr(\pi_i=k, x)$ = total product weights of all paths through the Viterbi graph for x that pass through the node (k, i).
- Each such path is formed by a blue subpath ending in node k and a red subpath starting in node k



Viterbi graph with all edges reversed

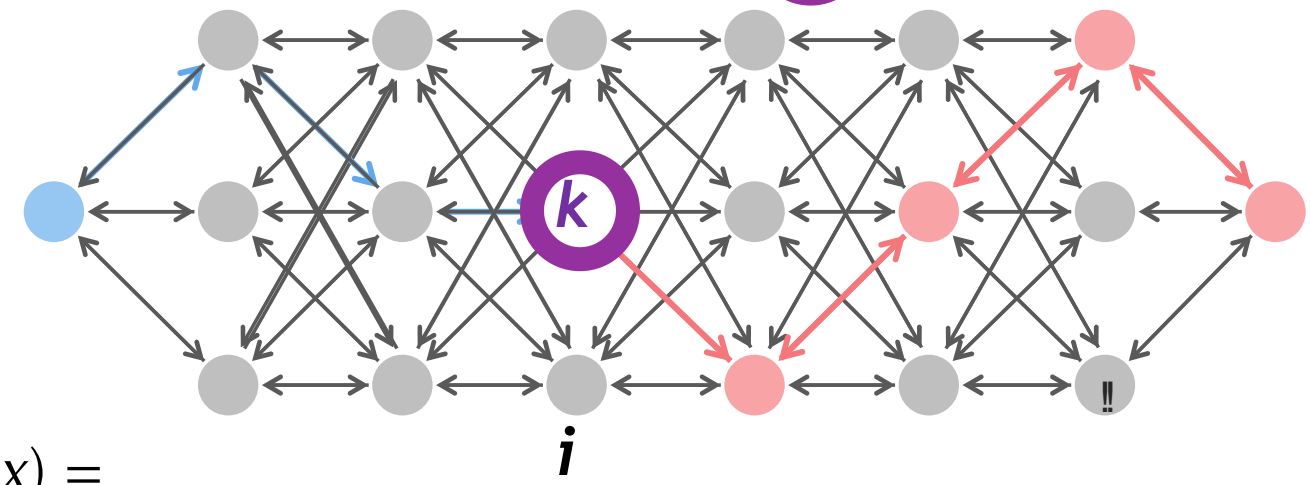
$\Pr(\pi_i=k, x)$ =

$\sum$ product weights of all blue paths * $\sum$ product weights of all red paths

$forward_{k,i}$ * ???

Computing $\Pr(\pi_i=k, x)$

- $\Pr(\pi_i=k, x)$ = total product weights of all paths through the Viterbi graph for x that pass through the node (k, i).
- Each such path is formed by a blue subpath ending in node k and a red subpath starting in node k



Viterbi graph with all edges reversed

$\Pr(\pi_i=k, x)$ =

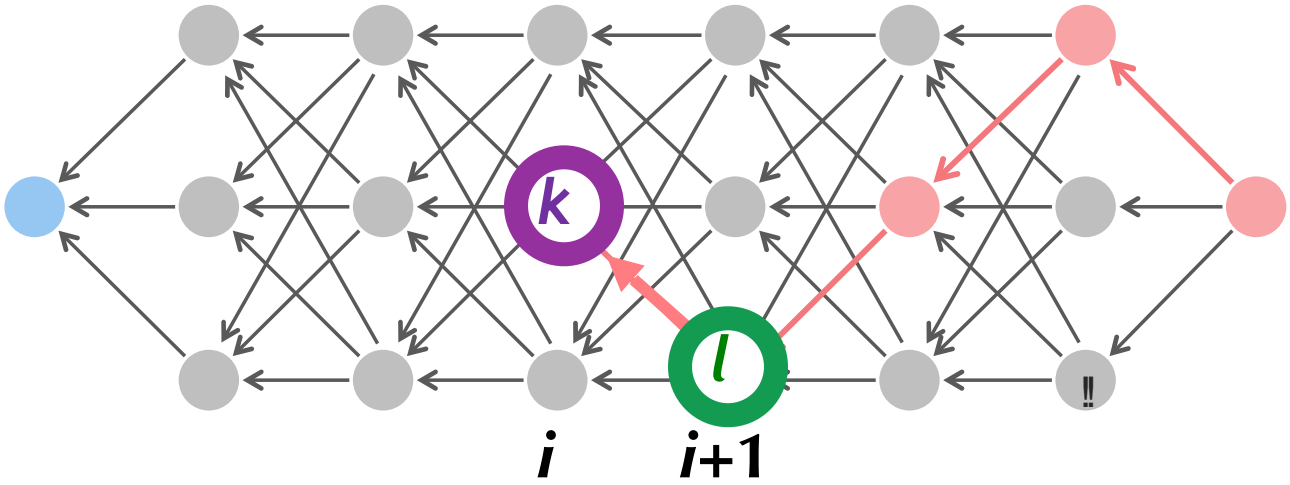
$\sum$ product weights of all blue paths * $\sum$ product weights of all red paths

$forward_{k,i}$ * $backward_{k,i}$

Forward-Backward Algorithm

- Since the reverse edge connecting node (l, i+1) to node (k, i) in the reversed graph has *weight*(k, l, i):

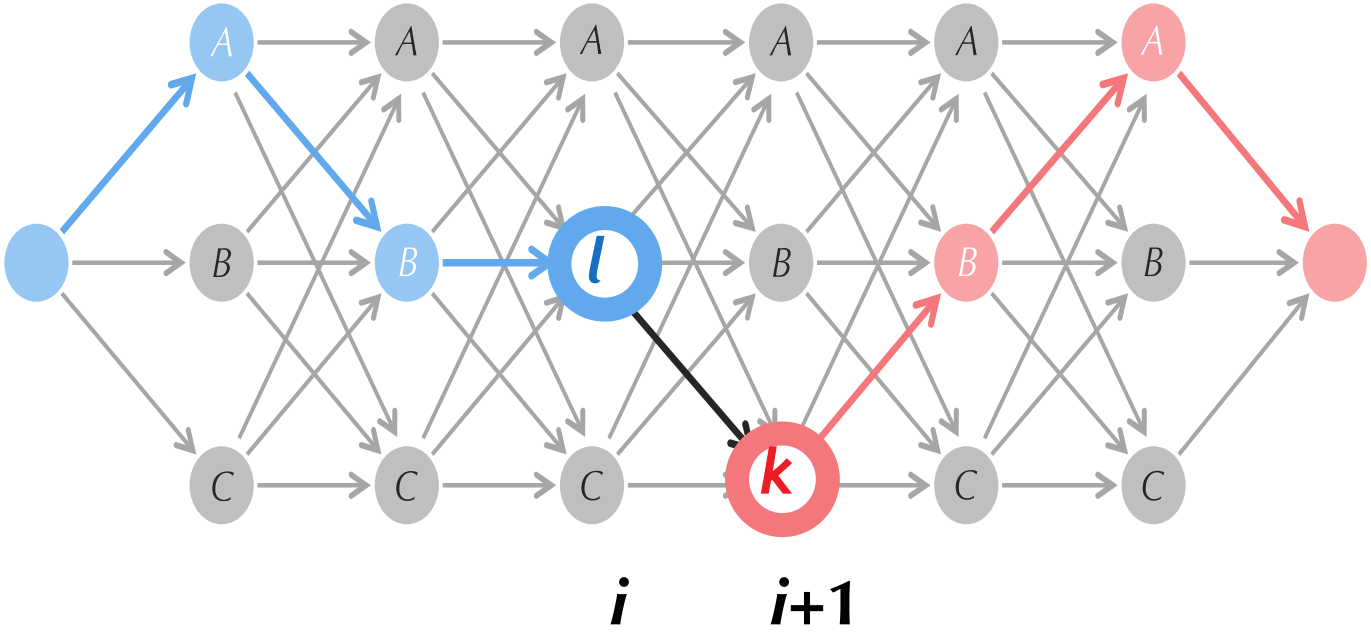
$$backward_{k,i} = \sum_{\text{all states } l} backward_{l,i+1} \cdot weight(k, l, i)$$



- Combining the forward-backward algorithm with the solution to the Outcome Likelihood Problem yields

$$\Pr(\pi_i = k \mid x) = \Pr(\pi_i = k, x)/\Pr(x) = \frac{Forward_{k,i} * backward_{k,i}}{forward(sink)}$$

The Conditional Probability $\Pr(\pi_i = l, \pi_{i+1} = k | x)$ that the HMM Passes Through an ***Edge*** in the Viterbi Graph

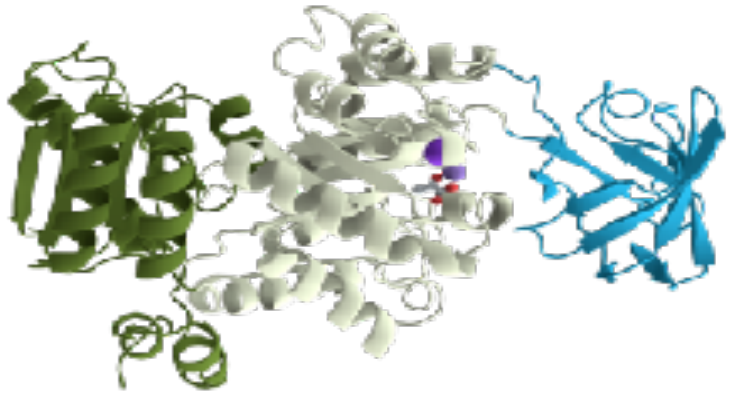


∑ weights of blue paths * weight of black edge * ∑ weights of red paths

$$\Pr(\pi_i = l, \pi_{i+1} = k | x) = \frac{forward_{l,i} * weight(l, k, i) * backward_{k,i+1}}{forward(sink)}$$

Hidden Markov Models

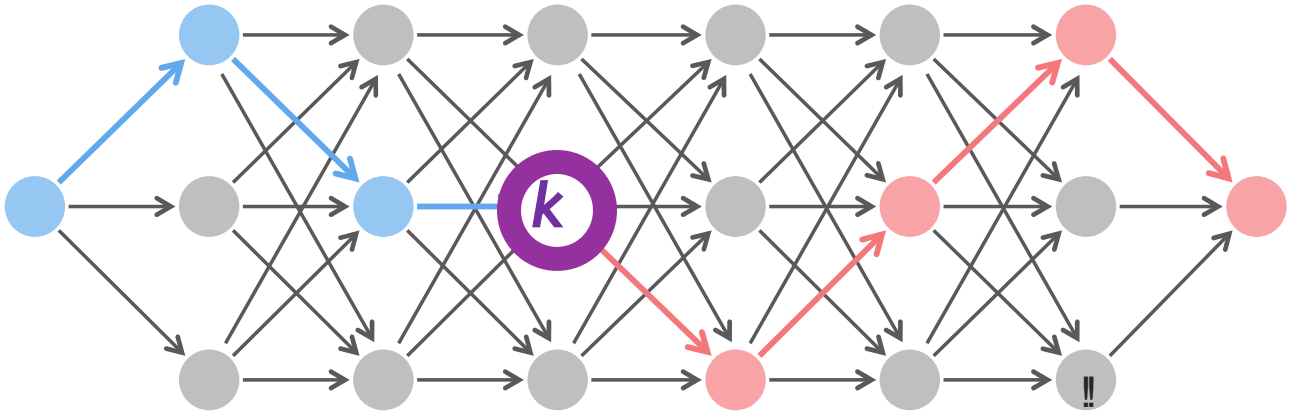
- Gambling with Yakuza
- From a Crooked Casino to a Hidden Markov Model
- Decoding Problem
- The Viterbi Algorithm
- Profile HMMs for Sequence Alignment
- Classifying proteins with profile HMMs
- Viterbi Learning
- Soft Decoding Problem
- Baum-Welch Learning



Node Responsibility Matrix

- Node responsibility matrix $\Pi^* = (\Pi^*_{k,i})$:

$$\Pi^*_{k,i} = \Pr(\pi_i = k \mid x)$$



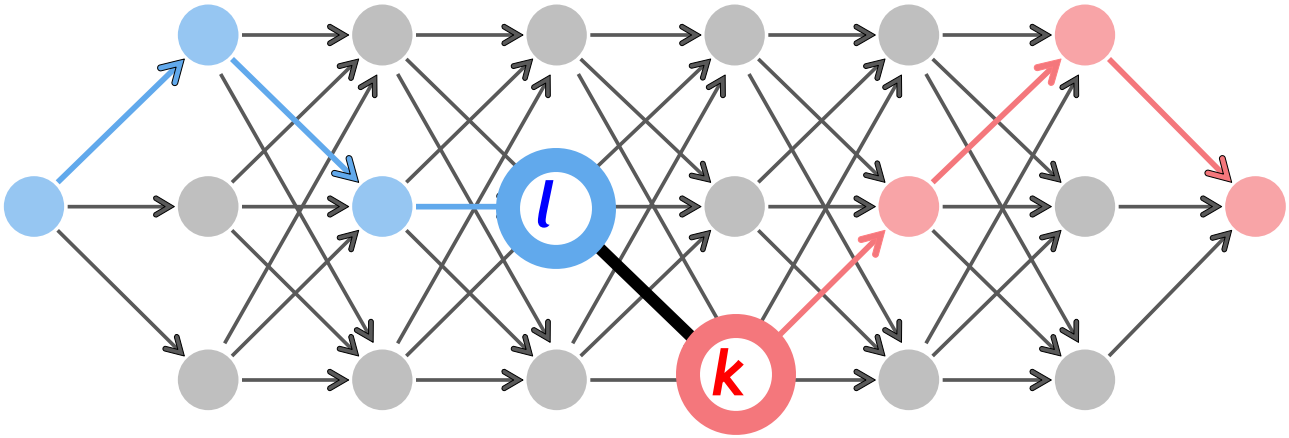
Node responsibility matrix for the crooked casino

	T	H	T	H	H	H	T	H	T	T	H
F	0.636	0.593	0.600	0.533	0.515	0.544	0.627	0.633	0.692	0.686	0.609
B	0.364	0.407	0.400	0.467	0.485	0.456	0.373	0.367	0.308	0.314	0.391

Edge Responsibility Matrix

- Edge responsibility matrix

$$\Pi^{**}_{l,k,i} = \Pr(\pi_i = l,\ \pi_{i+1} = k \mid x)$$



Edge responsibility matrix for the crooked

Every element of Π^* (Π^{**}) corresponds to a node (edge) in the Viterbi graph. Define

$$\Pi = (\Pi^*, \Pi^{**})$$

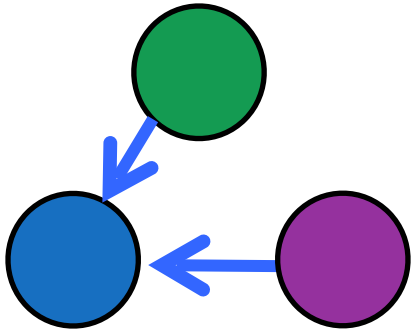
0.426 0.418 0.351 0.322 0.282 0.265 0.293

Baum-Welch Learning

Baum-Welch learning alternates between two steps:

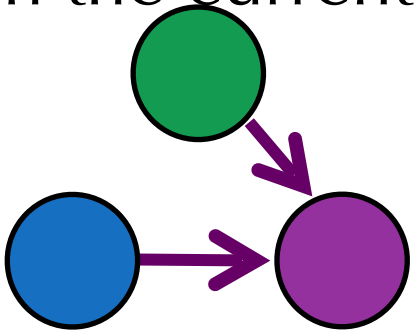
- Re-estimating the responsibility profile Π given the current HMM parameters (the **E-step**):

  (emitted string, ?, *Parameters*) → Π



- Re-estimating the HMM parameters given the current responsibility profile (the **M-step**):

  (emitted string, Π, ?) → *Parameters*



Using a Responsibility Matrix to Compute *Parameters*

- We have defined a transformation

$$(x, \pi, ?) \rightarrow \textit{Parameters}$$

that uses estimators $T_{l,k}$ and $E_k(b)$ based on a path π.

- We now want to define a transformation:

$$(x, \Pi, ?) \rightarrow \textit{Parameters}$$

but the **path is unknown**.

Idea: Use *expected* values $T_{l,k}$ and $E_k(b)$ over *all* possible paths.

Redefining Estimators for *Parameters* (for a known path π)

- $T_{l,k}$: #transitions from state *l* to state *k* in path π

$$T^i_{l,k} = \begin{cases} 1 \text{ if } \pi_i = l \text{ and } \pi_{i+1} = k \\ 0 \text{ otherwise} \end{cases}$$

Rewriting estimators: $T_{l,k} = \sum_{i=1,n-1} T^i_{l,k}$

HHTHHHTHHHTTTTHTTTTH

BFFBFFFBBFFBFFBBBBFF

$T^i_{B,F}$ = 1001000001001000001100

$T_{B,F} = 5$

Redefining Estimators for *Parameters* (for a known path π)

- $T_{l,k}$: #transitions from state l to state k in path π
- $E_k(b)$: #times b is emitted when the path π is in state k
- We now define

$$T^i_{l,k} = \begin{cases} 1 \text{ if } \pi_i = l \text{ and } \pi_{i+1} = k \\ 0 \text{ otherwise} \end{cases}$$

$$E^i_k(b) = \begin{cases} 1 \text{ if } \pi_i = k \text{ and } x_i = b \\ 0 \text{ otherwise} \end{cases}$$

How would you redefine these estimators if π is unknown?

Rewriting estimators:

$$T_{l,k} = \sum_{i=1,n-1} T^i_{B,F} \qquad E_k(b) = \sum_{i=1,n} E^i_k(b)$$

$E^i_F(T)$ =
0 0 **1** 0 0 0 **1** 0 0 0 **1** 0 **11** 0 0 0 0 **1** 0
H H T H H H H T H H H H T T T T H T T T T H
B F F B F F F F B B F F B F F F B B B B F F

$E_F(T)=6$

$T_{B,F} = 5$

$T^i_{B,F}$ = **1** 0 0 **1** 0 0 0 0 **1** 0 0 **1** 0 0 0 0 0 **1** 0 0

Redefining the Estimators $T^i_{l,k}$ and $E^i_k(b)$ When the Path is Unknown

- $T_{l,k}$: #transitions from state l to state k in path π
- $E_k(b)$: #times b is emitted when the path π is in state k
- We now define

$$T^i_{l,k}=\begin{cases}1 \text{ if } \pi_i = l \text{ and } \pi_{i+1} = k\\ 0 \text{ otherwise}\end{cases}$$

$$E^i_k(b)=\begin{cases}1 \text{ if } \pi_i = k \text{ and } x_i = b\\ 0 \text{ otherwise}\end{cases}$$

$$T^i_{l,k}=\Pr(\pi_i = l,\ \pi_{i+1} = k|x)$$

$$E^i_k(b)=\begin{cases}\Pr(\pi_i = k|x) \text{ if } x_i = b\\ 0 \text{ otherwise}\end{cases}$$

$$T^i_{l,k} = \Pi^{**}_{l,k,i}$$

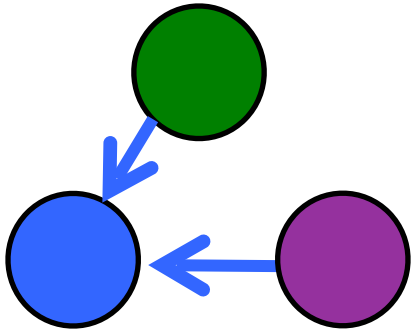
$$E^i_k(b)=\begin{cases}\Pi^{*}_{k,i} \text{ if } x_i = b\\ 0 \text{ otherwise}\end{cases}$$

Baum-Welch Learning

Baum-Welch learning alternates between two steps:

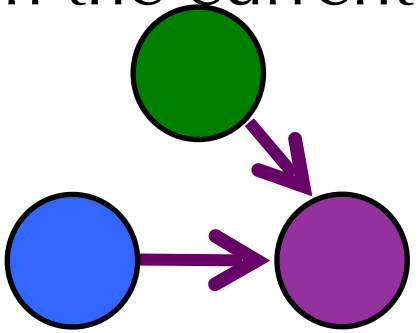
- Re-estimating the responsibility profile Π given the current HMM parameters (the **E-step**):

  (emitted string, ?, *Parameters*) → Π



- Re-estimating the HMM parameters given the current responsibility profile (the **M-step**):

  (emitted string, Π, ?) → *Parameters*



Stopping Rules for the Baum-Welch Learning

- Compute the probability that the HMM emits the string *x* under current *Parameters*:

$$\Pr(\textit{emitted string} \mid \textit{Parameters})$$

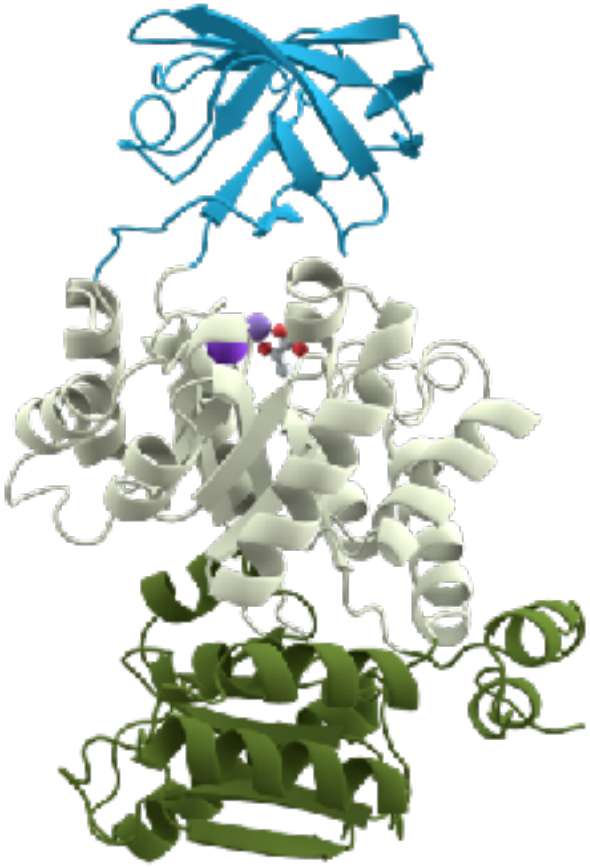
  - Compare with the probability for previous values of *Parameters* and stop if the difference is small.
  - Stop after a certain number of iterations.

Nature is a Tinkerer and Not an Inventor

Protein domain: a conserved part of a protein that often can function independently.

Nature uses domains as building blocks, shuffling them to create **multi-domain proteins**.

Goal: classify domains into families even though sequence similarities between domains from the same family can be low.



A multi-domain protein

Searching for Protein Domains with Profile HMMs

1. Use alignments to break proteins into domains.

```
ABCDEFGHKLMNP
ERGHKLNPABTD
KLSNPACDEFTH
```

Searching for Protein Domains with Profile HMMs

1. Use alignments to break proteins into domains.

```
ABCDEFGHKLMNP
ERGHKLNPABTD
KLSNPACDEFTH
```

Searching for Protein Domains with Profile HMMs

1. Use alignments to break proteins into domains.
2. Construct alignment of domains from a given family (starting from highly similar domains whose attribution to a family is non-controversial).

```
ABCDEFGHKLMNP
ERGHKLNPABTD
KLSNPACDEFTH
```

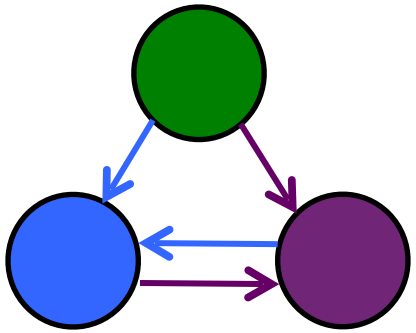
```
ABCD   KLMNP  EFGH
ABTD   KL-NP  ERGH
AC-D   KLSNP  EFTH
```

Searching for Protein Domains with Profile HMMs

1. Use alignments to break proteins into domains.
2. Construct alignment of domains from a given family (starting from highly similar domains whose attribution to a family is non-controversial).
3. For each family, construct a profile HMM and estimate its parameters.

```
ABCDEFGHKLMNP
ERGHKLNPABTD
KLSNPACDEFTH
```

```
ABCD   KLMNP   EFGH
ABTD   KL-NP   ERGH
AC-D   KLSNP   EFTH
```

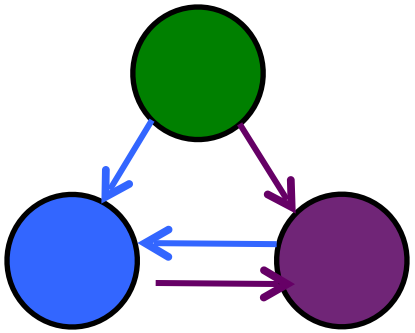


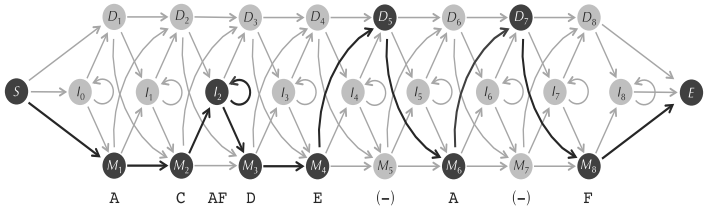
Searching for Protein Domains with Profile HMMs

1. Use alignments to break proteins into domains.
2. Construct alignment of domains from a given family (starting from highly similar domains whose attribution to a family is non-controversial).
3. For each family, construct a profile HMM and estimate its parameters.
4. Align the new sequence against each such HMM to find the best fitting HMM.

```
ABCDEFGHKLMNP
ERGHKLNPABTD
KLSNPACDEFTH
```

```
ABCD   KLMNP   EFGH
ABTD   KL-NP   ERGH
AC-D   KLSNP   EFTH
```





Pfam: Profile HMM Database

Each domain family in Pfam has:

- **Seed alignment**: Initial multiple alignment of domains in this family.
- **HMM**: Built from seed alignment for new searches.
- **Full alignment**: Enlarged multiple alignment generated by aligning new domains against the seed HMM.