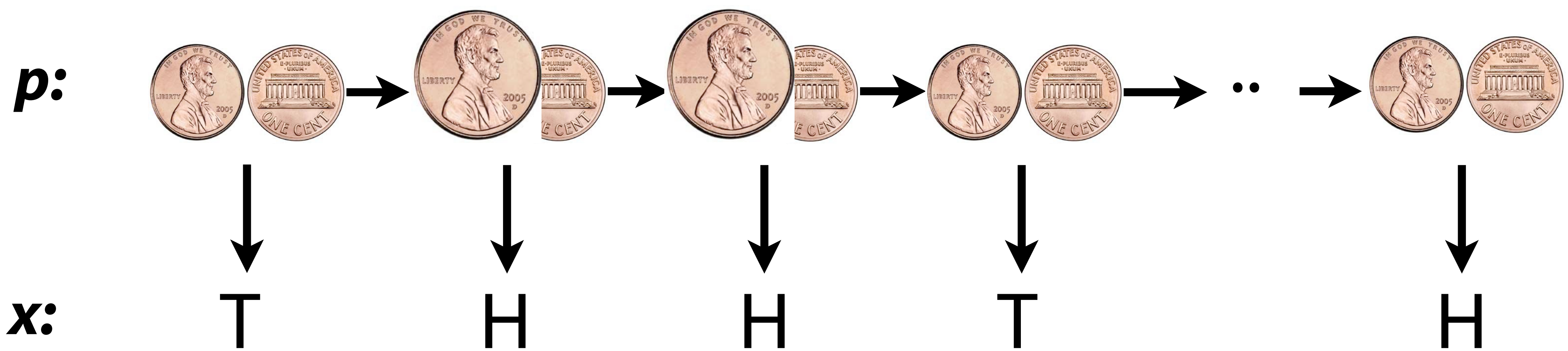
HMMs part 2: decoding with Viterbi

Ben Langmead

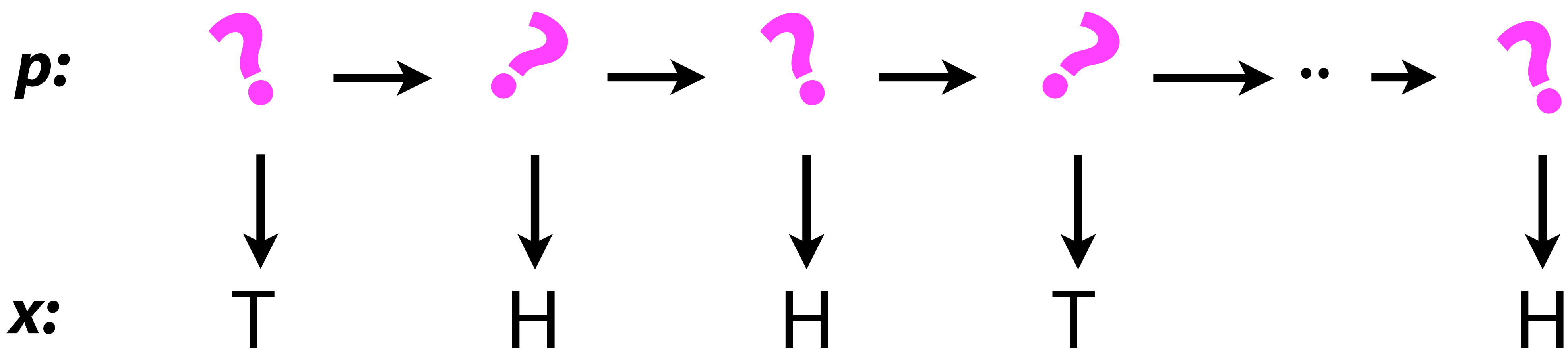
For original Keynote files, email me (ben.langmead@gmail.com)

Hidden Markov Model



Hidden Markov Model

When was the dealer using the loaded coin?



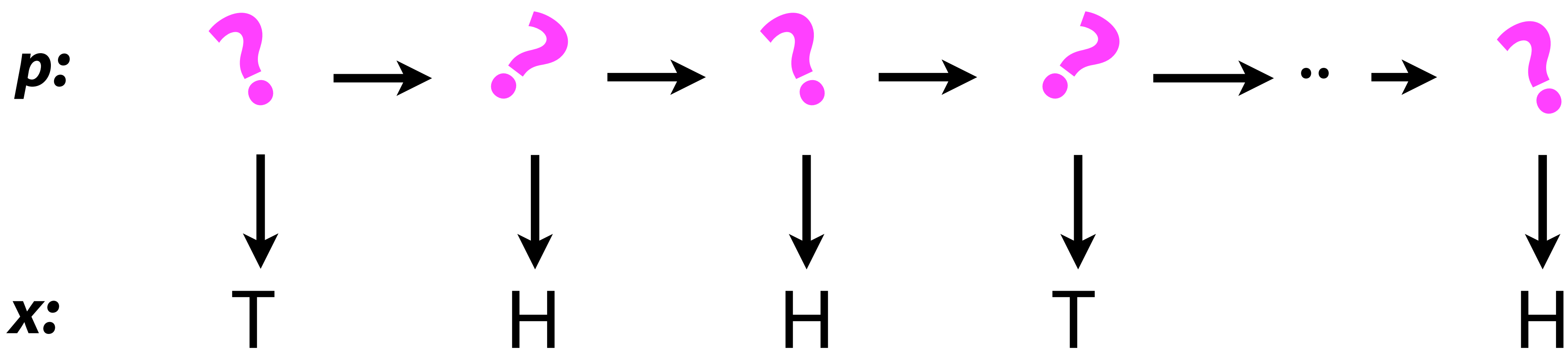
Many possible ***paths p***; fair-or-loaded sequences

p* denotes the *most likely* given the emission sequence ***x***

$$\boldsymbol{p}^* = \underset{p}{\operatorname{argmax}}\ P(\, p \mid x \,) = \underset{p}{\operatorname{argmax}}$$

Hidden Markov Model

When was the dealer using the loaded coin?



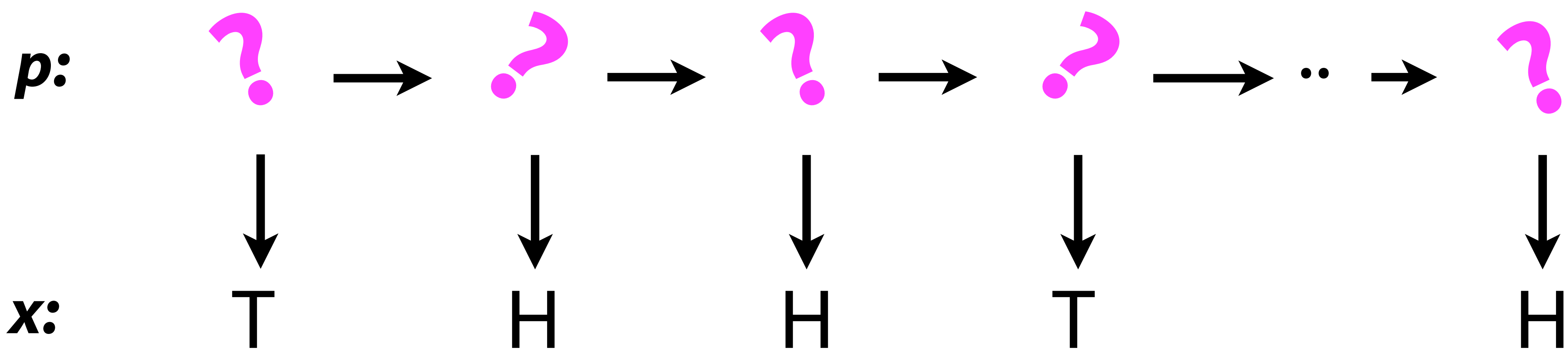
Many possible ***paths p***; fair-or-loaded sequences

p* denotes the *most likely* given the emission sequence ***x***

$$\boldsymbol{p}^* = \underset{p}{\operatorname{argmax}}\, \mathrm{P}(\,p \mid x\,) = \underset{p}{\operatorname{argmax}}\, \frac{\mathrm{P}(\,p, x\,)}{\mathrm{P}(x\,)}$$

Hidden Markov Model

When was the dealer using the loaded coin?



Many possible ***paths p***; fair-or-loaded sequences

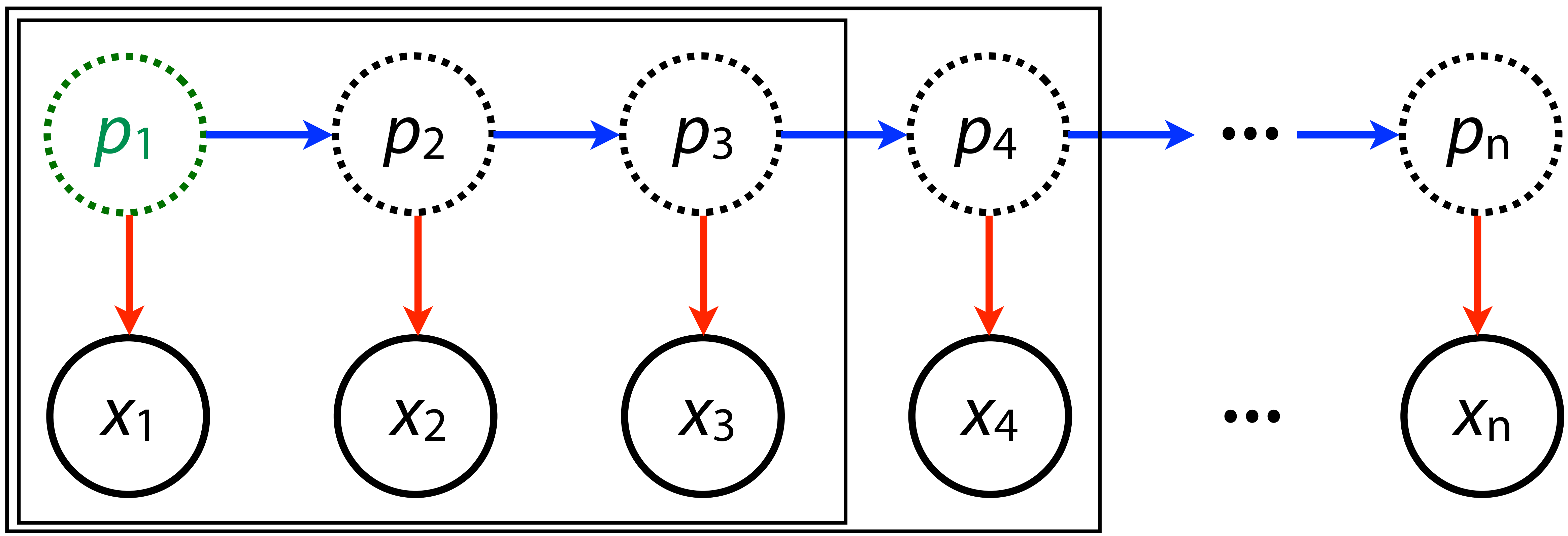
p* denotes the *most likely* given the emission sequence ***x***

$$\boldsymbol{p}^* = \underset{p}{\operatorname{argmax}}\ \mathrm{P}(\, p \mid x \,) = \underset{p}{\operatorname{argmax}}\ \mathrm{P}(\, p, x \,)$$

P(x) denominator doesn't affect argmax

Finding ***p**** given **x** is *decoding*

Hidden Markov Model



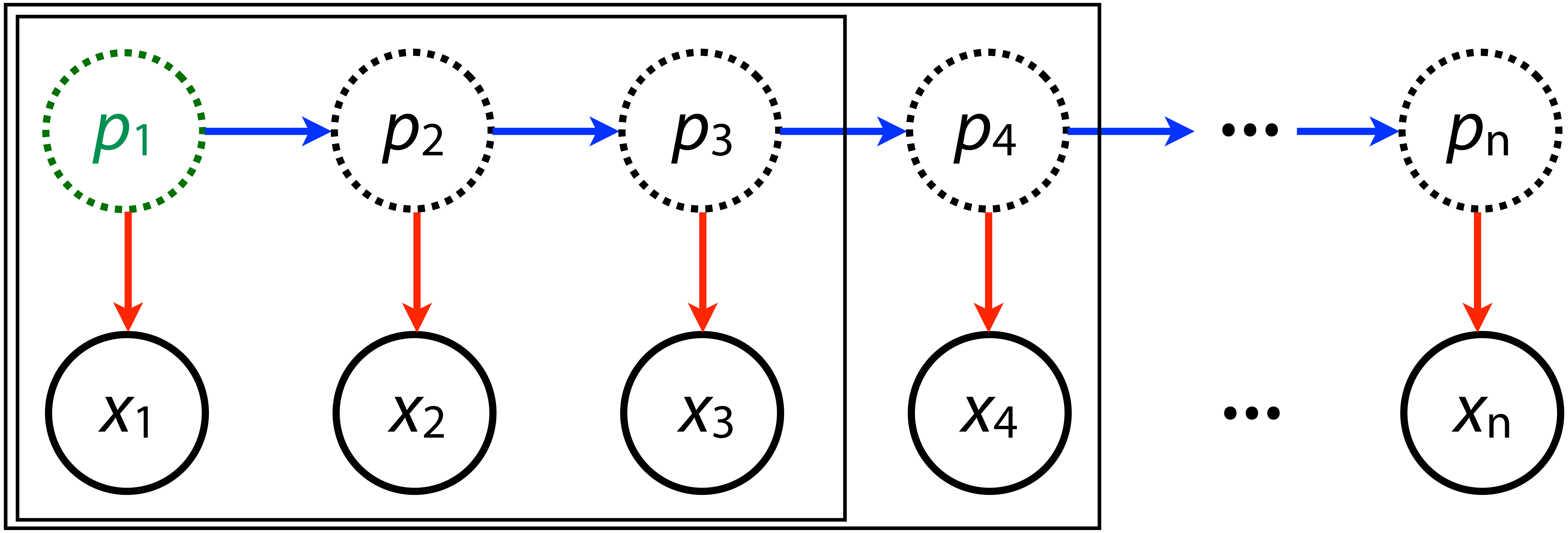
$$P(p_1, p_2, ..., p_n, x_1, x_2, ..., x_n) = \prod_{k=1}^{n} P(x_k \mid p_k) \cdot \prod_{k=2}^{n} P(p_k \mid p_{k-1}) \cdot P(p_1)$$

Say we know $P(p_1, p_2, p_3, x_1, x_2, x_3)$,

Computing $P(p_1, p_2, p_3, p_4, x_1, x_2, x_3, x_4)$ is easy…

…just multiply in $P(x_4 \mid p_4)$ and $P(p_4 \mid p_3)$

Hidden Markov Model



$$P(p_1, p_2, \ldots, p_n, x_1, x_2, \ldots, x_n) = \prod_{k=1}^{n} P(x_k \mid p_k) \cdot \prod_{k=2}^{n} P(p_k \mid p_{k-1}) \cdot P(p_1)$$

Say we know $\underset{p_1, p_2, p_3}{\operatorname{argmax}} P(p_1, p_2, p_3, x_1, x_2, x_3)$,

Is computing $\underset{p_1, p_2, p_3, p_4}{\operatorname{argmax}} P(p_1, p_2, p_3, p_4, x_1, x_2, x_3, x_4)$ easy?

Sort of!

Decoding example

Dealer repeatedly flips a coin. Coin is sometimes *fair*, P(heads) = 0.5, sometimes *loaded*, P(heads) = 0.8. Between flips, dealer switches coins with prob. 0.2.

50/50 prob of starting with fair/loaded

A	F	L
F	0.8	0.2
L	0.2	0.8

E	H	T
F	0.5	0.5
L	0.8	0.2

	F	L
I	0.5	0.5

For X = HTTHH , argmax_p = FFFFF

For X = HTTHHH, argmax_p = LLLLLL

Decoding example

A	F	L
F	0.8	0.2
L	0.2	0.8

E	H	T
F	0.5	0.5
L	0.8	0.2

	F	L
I	0.5	0.5

For X = HTTHH

argmax = FFFFF

$p_1, p_2, p_3, p_4, p_{5=F}$

argmax = LLLLL

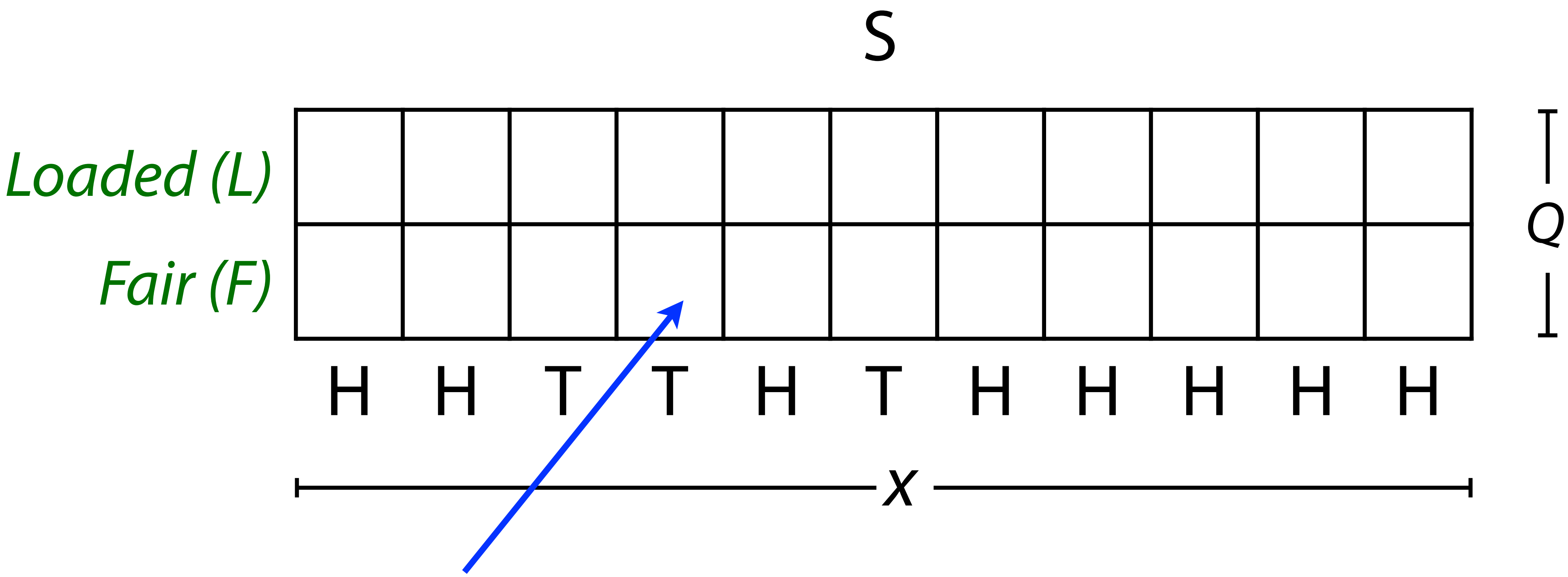
$p_1, p_2, p_3, p_4, p_{5=L}$

For X = HTTHHH

Answer will be an extension of one of these

Hidden Markov Model: Viterbi algorithm

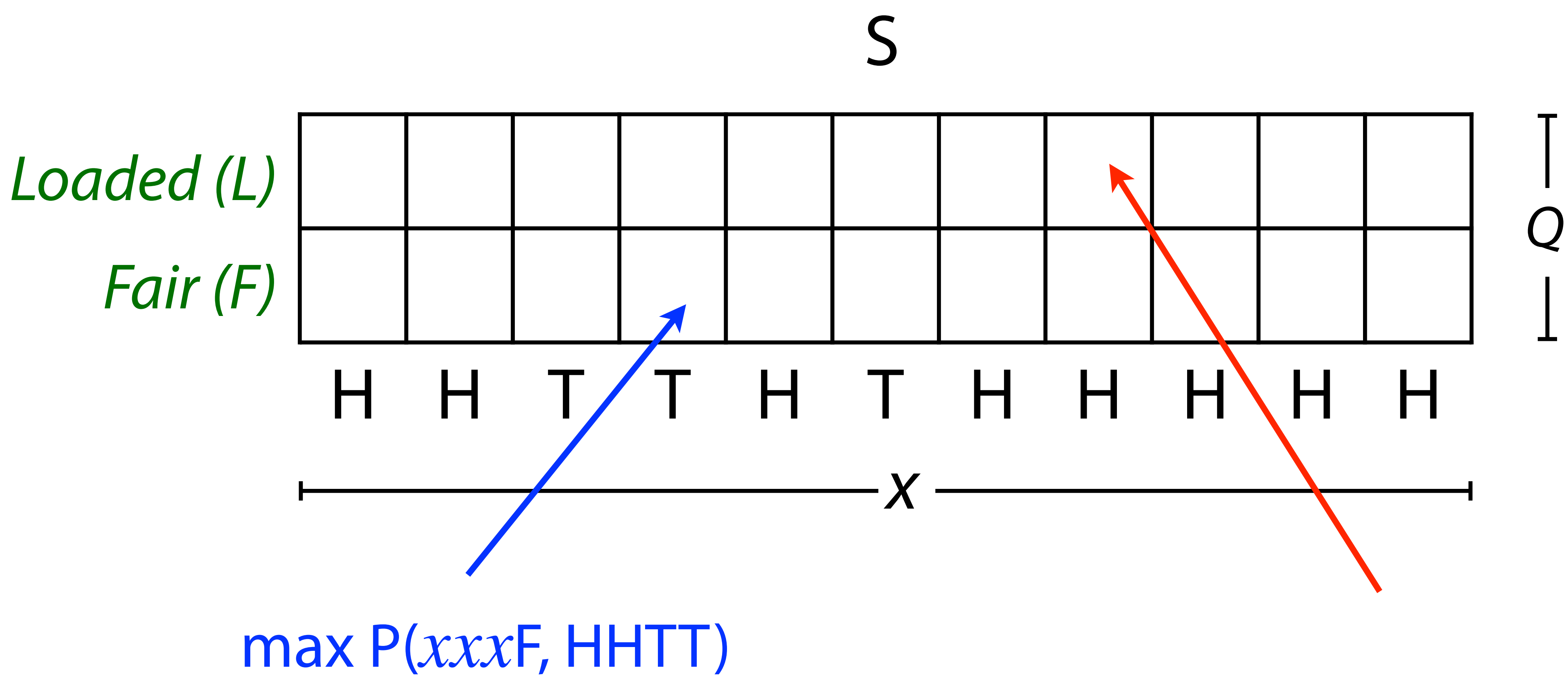
Fill in a ***dynamic programming*** matrix *S*



$S_{k,\,i}$ is the *max joint probability* of observing the length-*i*+*1* prefix of *x* and any sequence of states ending in state *k*

Hidden Markov Model: Viterbi algorithm

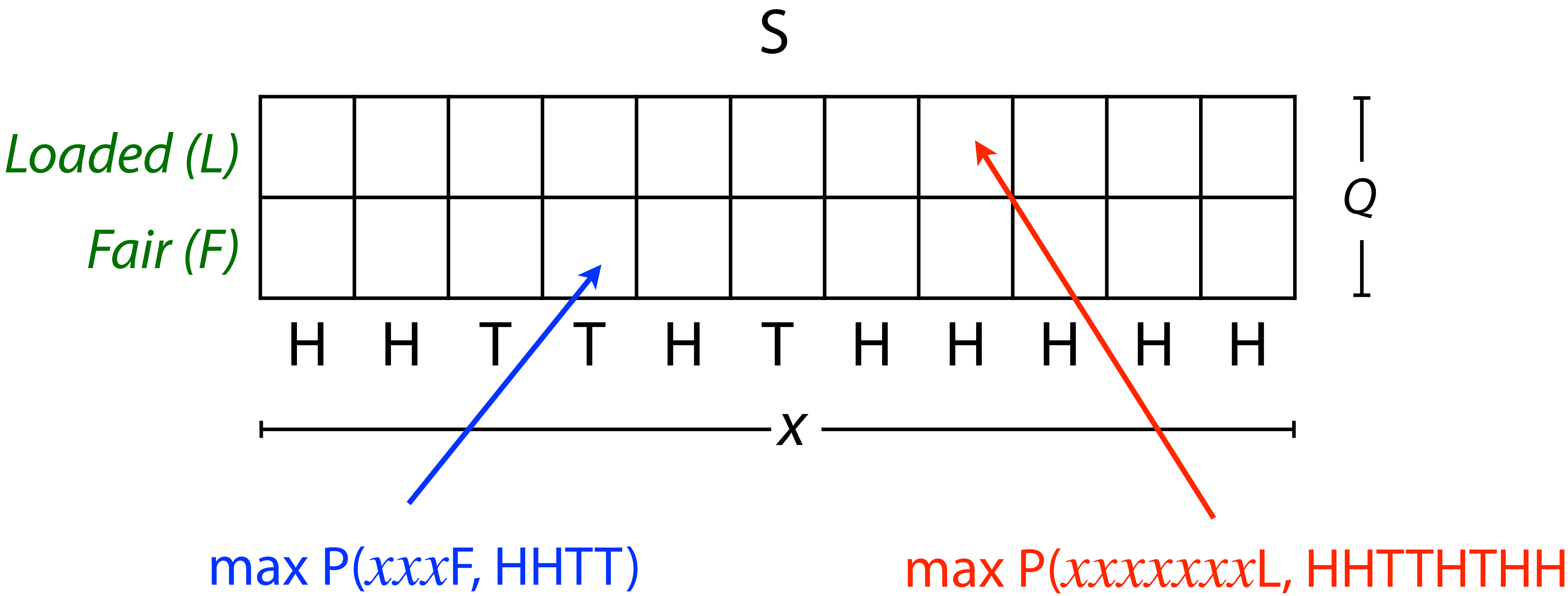
Fill in a ***dynamic programming*** matrix *S*



$S_{k,i}$ is the *max joint probability* of observing the length-*i+1* prefix of *x* and any sequence of states ending in state *k*

Hidden Markov Model: Viterbi algorithm

Fill in a ***dynamic programming*** matrix *S*



$S_{k,i}$ is the *max joint probability* of observing the length-i+1 prefix of x and any sequence of states ending in state k

Hidden Markov Model: Viterbi algorithm

Say x_i is Heads

$\boxed{\boldsymbol{s}_{\text{Loaded}, i}} =$

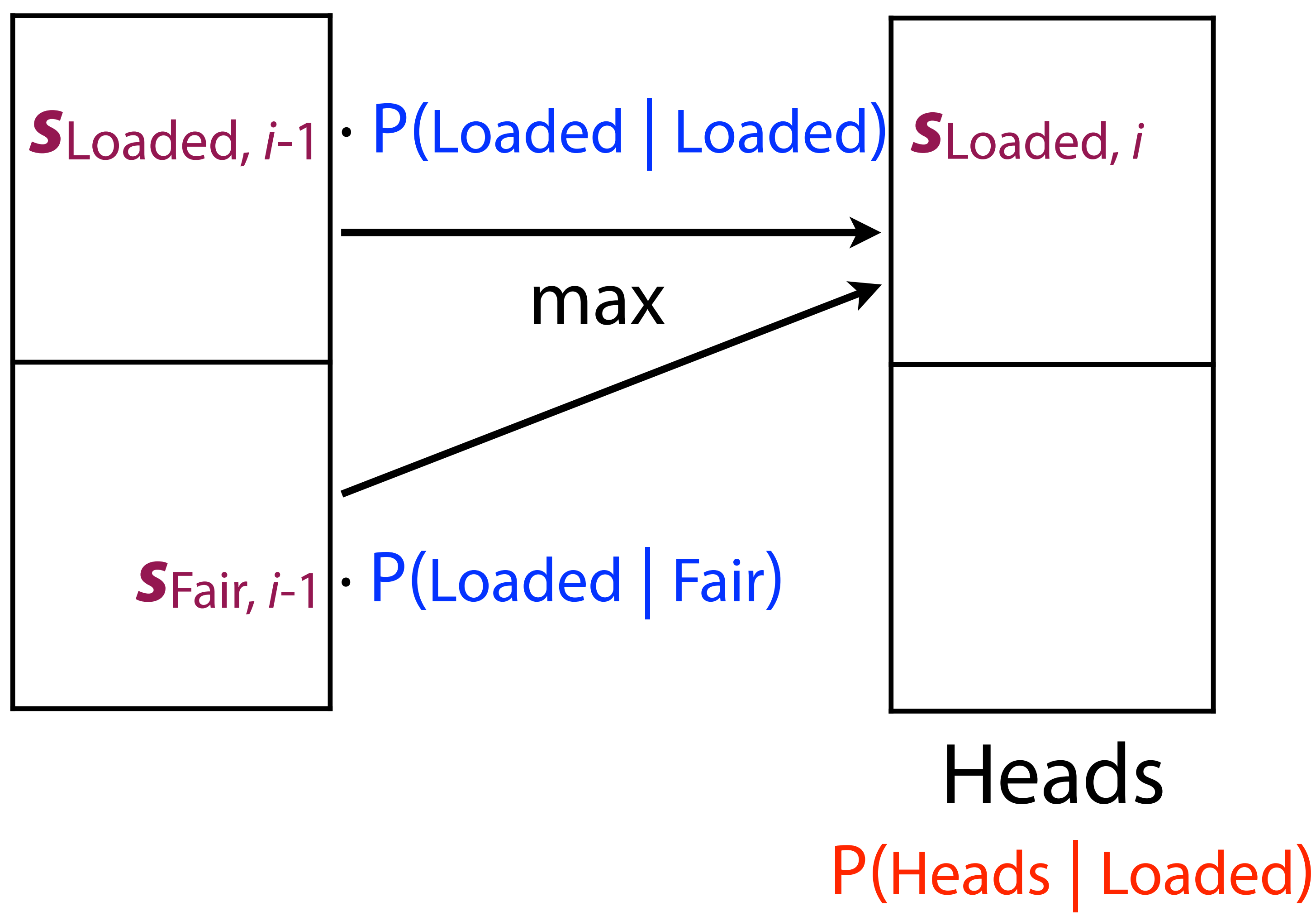
Hidden Markov Model: Viterbi algorithm

Say x_i is Heads

$$\boxed{s_{\text{Loaded},\,i}} = P(\text{Heads} \mid \text{Loaded}) \cdot \max_{k \in \{\text{Fair, Loaded}\}} \{\, s_{k,i-1} \cdot P(\text{Loaded} \mid k) \,\}$$

Emission prob

Transition prob



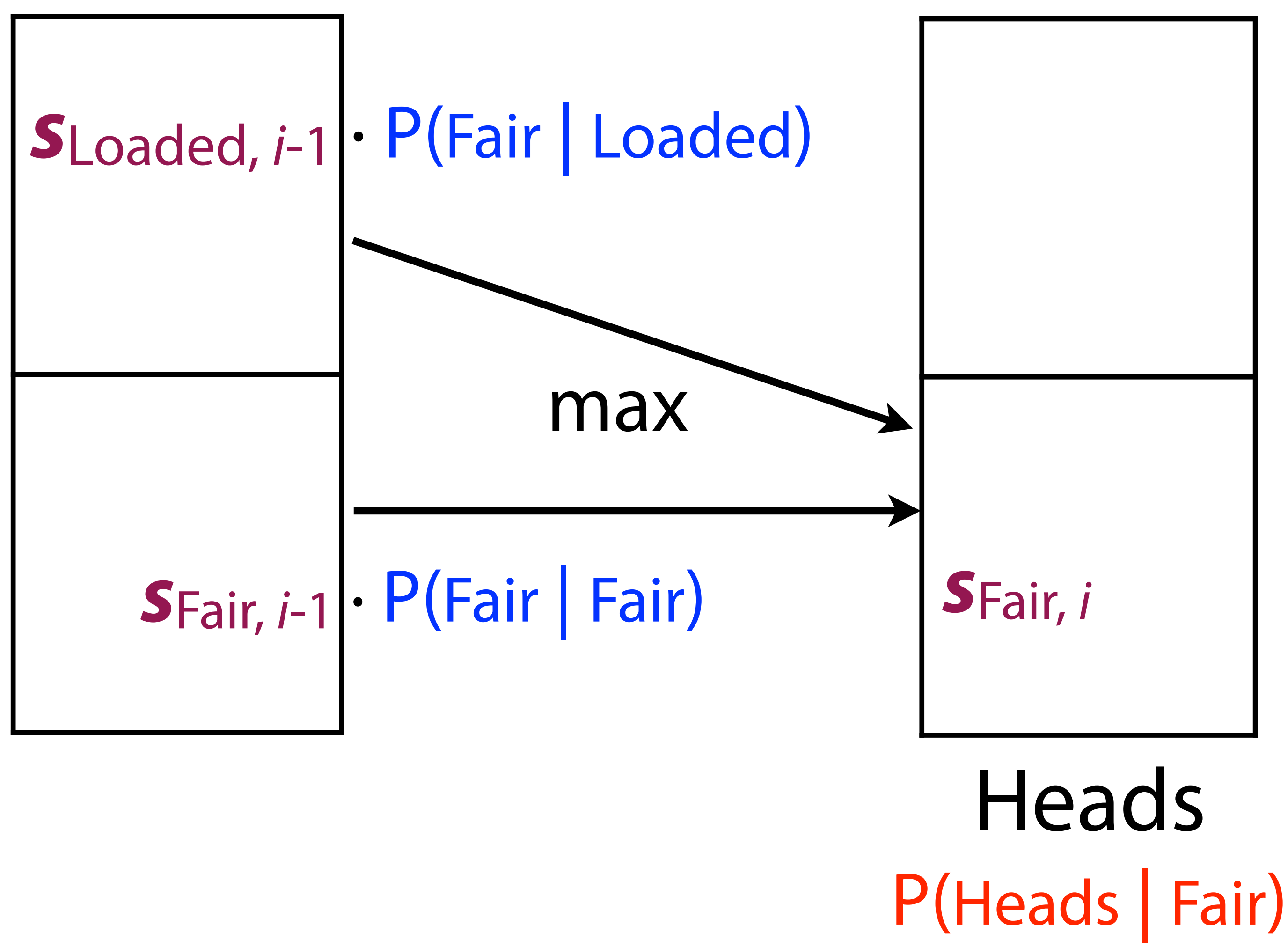
Hidden Markov Model: Viterbi algorithm

Say x_i is Heads

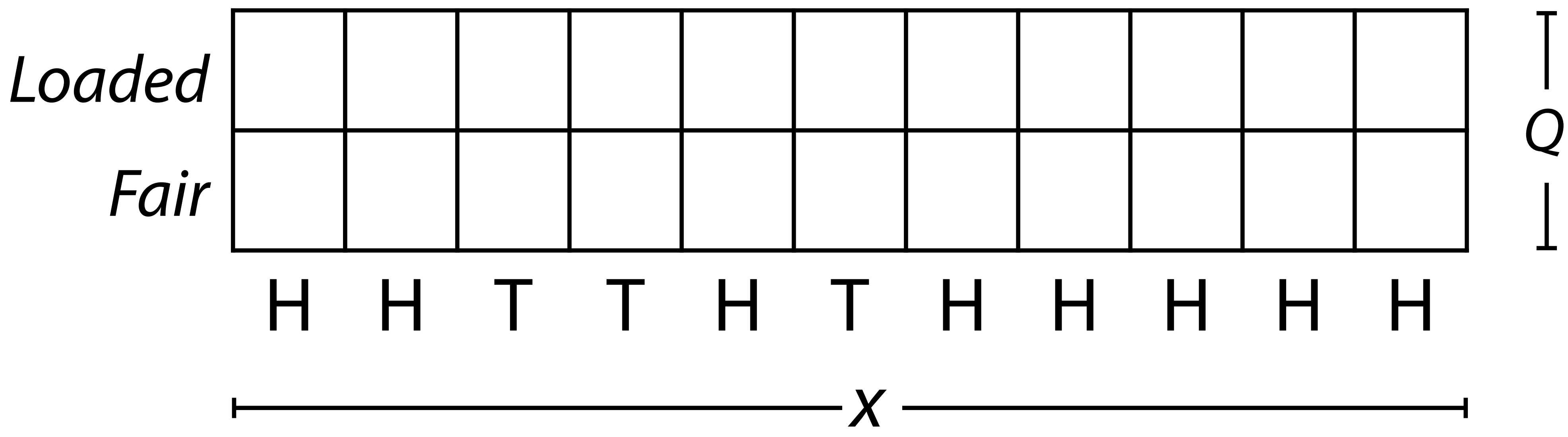
$$\boxed{s_{\text{Fair},\,i}} = P(\text{Heads} \mid \text{Fair}) \cdot \max_{k \,\in\, \{\text{Fair, Loaded}\}} \{\, s_{k,i-1} \cdot P(\text{Fair} \mid k) \,\}$$

Emission prob

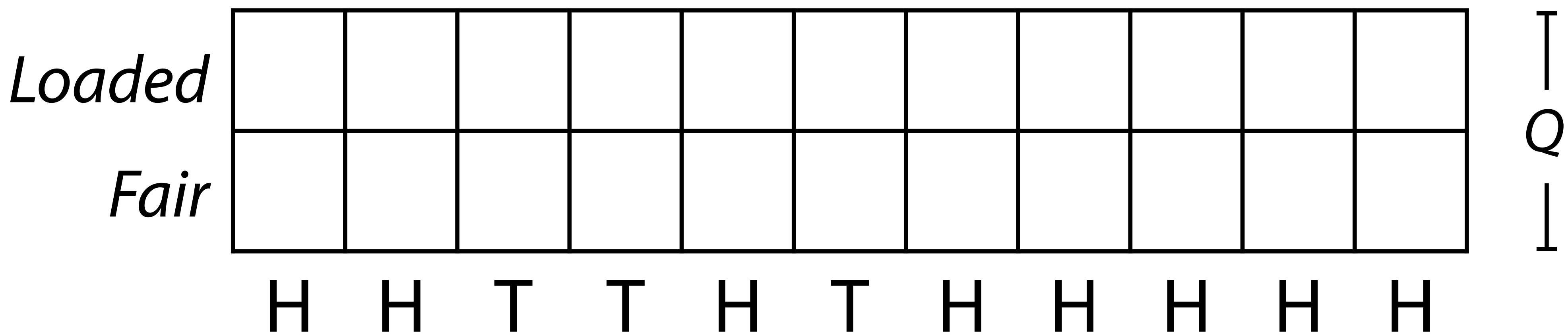
Transition prob



Hidden Markov Model: Viterbi algorithm



Hidden Markov Model: Viterbi algorithm



Dealer repeatedly flips a coin. Coin is sometimes *fair*, P(heads) = 0.5, sometimes *loaded*, P(heads) = 0.8. Between flips, dealer switches coins with prob. 0.4.

50/50 prob of starting with fair/loaded

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

	F	L
I	0.5	0.5

Hidden Markov Model: Viterbi algorithm

Loaded											
Fair											
	H	H	T	T	H	T	H	H	H	H	H

Q

$S_{L,0} =$

$S_{F,0} =$

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

	F	L
I	0.5	0.5

Hidden Markov Model: Viterbi algorithm

Loaded	0.4										
Fair	0.25										
	H	H	T	T	H	T	H	H	H	H	H

Q

$$S_{L,\,0} = P(L) \cdot P(H \mid L)$$
$$= 0.5 \cdot 0.8$$

$$S_{F,\,0} = P(H) \cdot P(H \mid F)$$
$$= 0.5 \cdot 0.5$$

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

	F	L
I	0.5	0.5

Hidden Markov Model: Viterbi algorithm

	H	H	T	T	H	T	H	H	H	H	H
Loaded	0.4										
Fair	0.25										

Q

$S_{L,1} =$

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

Hidden Markov Model: Viterbi algorithm

Loaded	0.4										
Fair	0.25										
	H	H	T	T	H	T	H	H	H	H	H

Q

$$S_{L,1} = P(H \mid L) \cdot \max \begin{cases} 0.4 \cdot \mathrm{P}(\mathrm{L} \mid \mathrm{L}) \\ 0.25 \cdot \mathrm{P}(\mathrm{L} \mid \mathrm{F}) \end{cases}$$

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

Hidden Markov Model: Viterbi algorithm

Loaded	0.4	0.19									
Fair	0.25										
	H	H	T	T	H	T	H	H	H	H	H

Q

$$S_{L,1} = 0.8 \cdot \max \begin{cases} 0.4 \cdot 0.6 \\ 0.25 \cdot 0.4 \end{cases}$$

$$= 0.8 \cdot 0.4 \cdot 0.6 \approx 0.19$$

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

Hidden Markov Model: Viterbi algorithm

Loaded	0.4	0.19									
Fair	0.25										
	H	H	T	T	H	T	H	H	H	H	H

Q

$$S_{F,1} = P(H \mid F) \cdot \max \begin{cases} 0.4 \cdot P(F \mid L) \\ 0.25 \cdot P(F \mid F) \end{cases}$$

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

Hidden Markov Model: Viterbi algorithm

Loaded	0.4	0.19										
Fair	0.25	0.08										Q
	H	H	T	T	H	T	H	H	H	H	H	

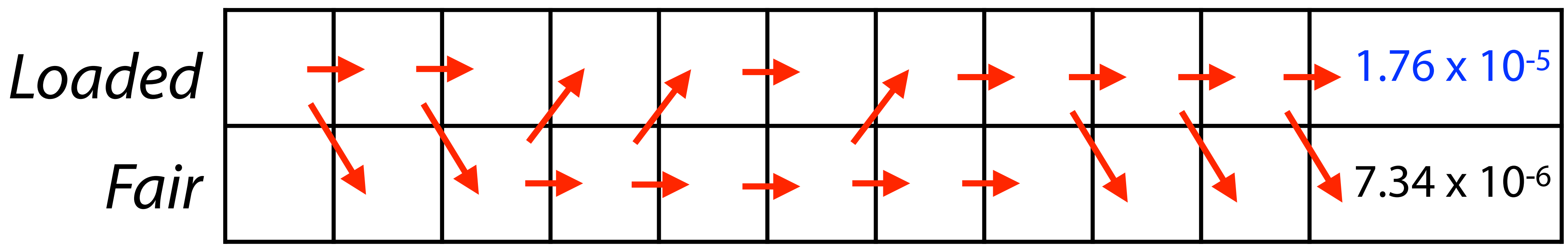
$$S_{F,1} = 0.5 \cdot \max \begin{cases} 0.4 \cdot 0.4 \\ 0.25 \cdot 0.6 \end{cases}$$

$$= 0.5 \cdot 0.4 \cdot 0.4 = 0.08$$

A	F	L
F	0.6	0.4
L	0.4	0.6

E	H	T
F	0.5	0.5
L	0.8	0.2

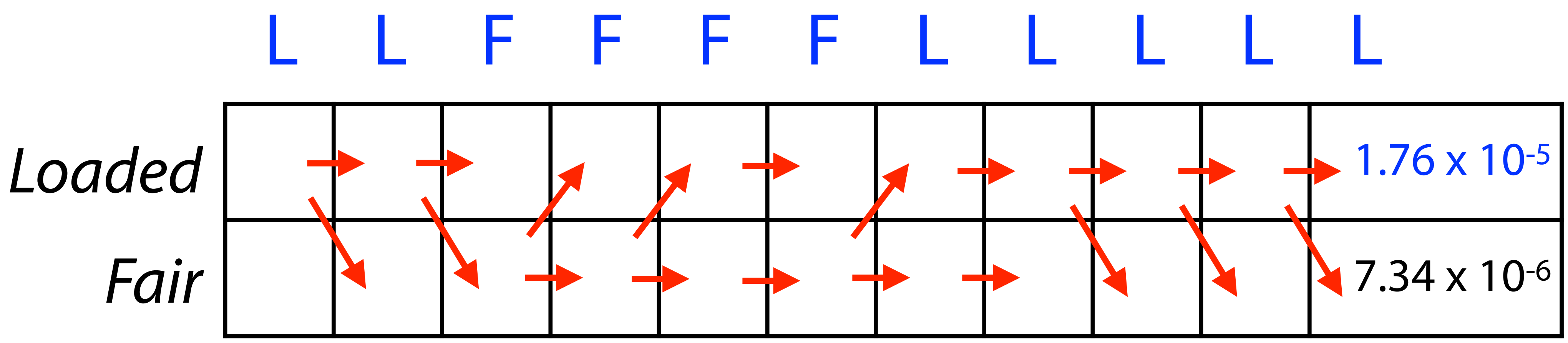
Hidden Markov Model: Viterbi algorithm



Traceback arrows track which term “wins” the maximum

Traceback procedure:

Hidden Markov Model: Viterbi algorithm



Traceback arrows track which term “wins” the maximum

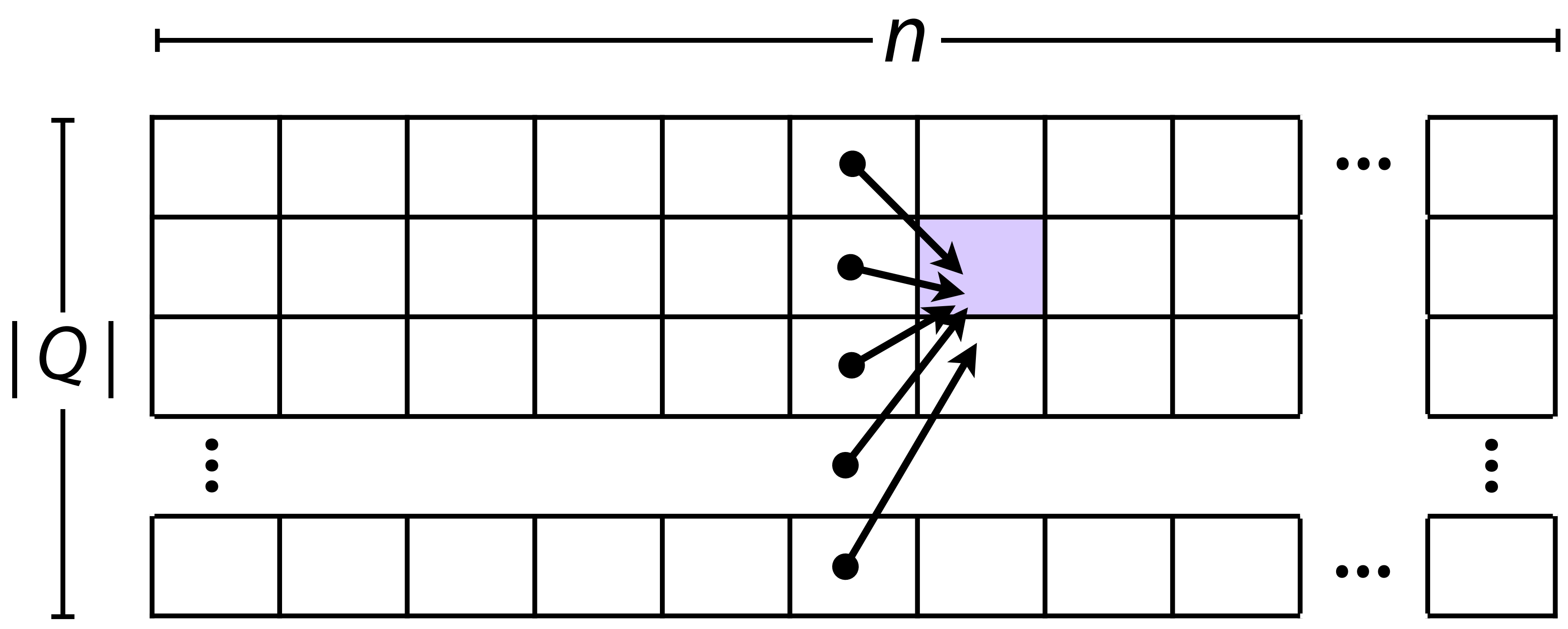
Traceback procedure:

Start from greatest score in final column

Keep asking "how did I get here?" (which predecessor "won" the maximum) until we reach 1st column

Hidden Markov Model: Viterbi algorithm

How much work is this? Q = set of states, n = length of emission string



Time: # $s_{k,i}$ values to calculate = $n \cdot |Q|$, each involves max over $|Q|$ products

$O(n \cdot |Q|^2)$

Space: Matrix A has $|Q|^2$ elements, E has $|Q||\Sigma|$ elements, I has $|Q|$ elements