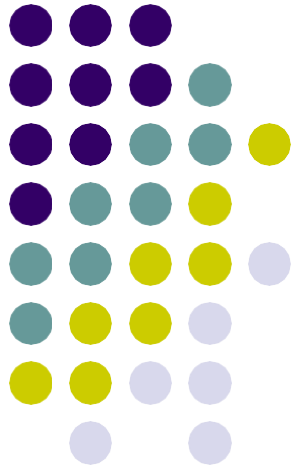
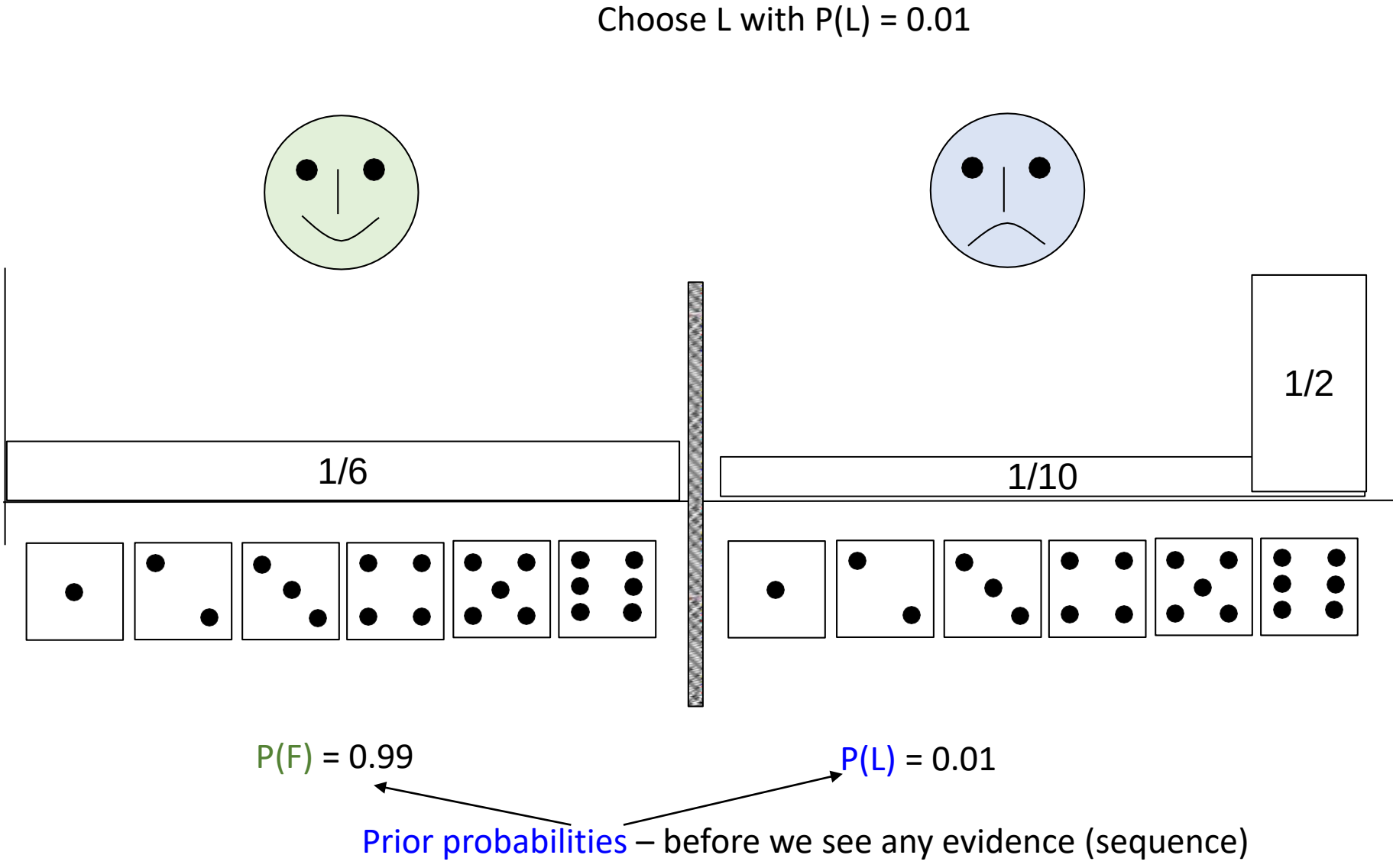
Probabilistic approaches. Hidden Markov Models

Lecture 11



The honest and the dishonest casino



Bayes theorem and the model comparison

- Pick a die at random - and roll
- We get 3 consecutive sixes
- Is the die loaded? What is the probability?

- We want to know P(L|3 sixes)
- From Bayes theorem:

P(L|3 sixes) = P(3 sixes|L)*P(L)/P(3 sixes)

P(F|3 sixes) = P(3 sixes|F)*P(F)/P(3 sixes)

The sequence was generated either by fair or by loaded die

P(3 sixes) = P(3 sixes|F)*P(F) + P(3 sixes|L) *P(L) = 0.0058

- P (L|3 sixes) = (0.5*0.5*0.5 * 0.01) /0.0058 = 0.215
- P(F|3 sixes) = (1/6)*(1/6)*(1/6)*0.99 / 0.0058 = 0.785

What are the odds?

- P (W1|evidence) = P(evidence|W1)*P(W1)/P(evidence)
- P (W2|evidence) = P(evidence|W2)*P(W2)/P(evidence)

- To compare P (W1|evidence) vs P (W2|evidence) :

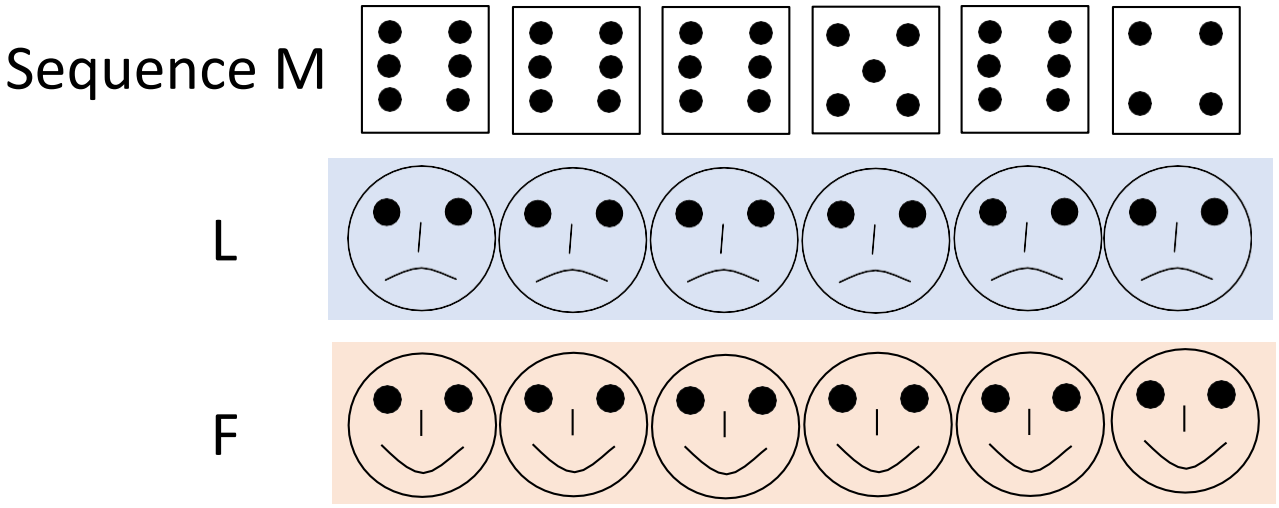
P (W1|evidence) / P (W2|evidence)

- Or to avoid underflow:

log (P (W1|evidence) / P (W2|evidence))

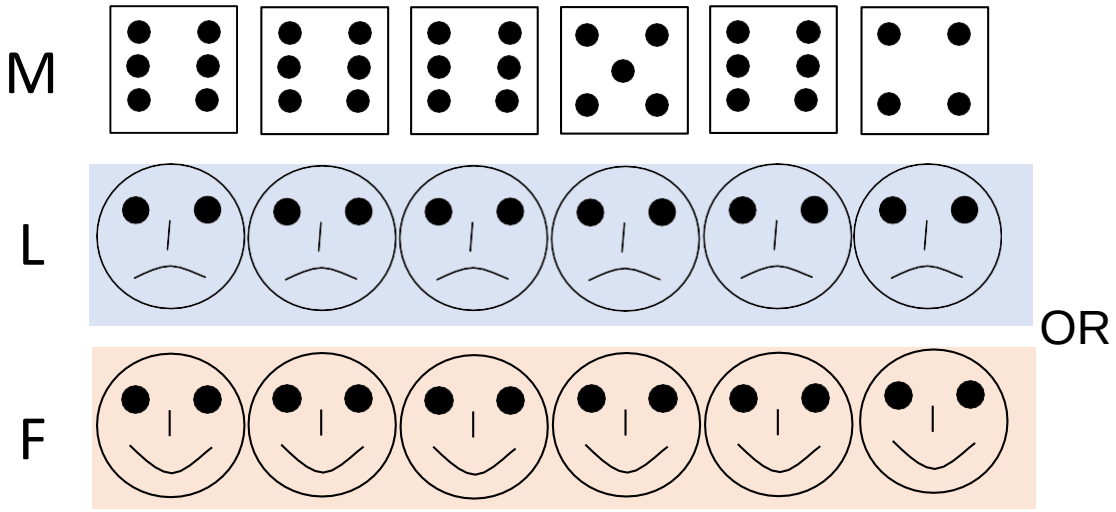
- Log odds ratio = log [P(evidence|W1)*P(W1)/ P(evidence|W2)*P(W2)]
- If > 0 – first is more likely, if < 0 – second is more likely

If two models **are <u>equally likely</u>**, we can use the conditional probabilities for discrimination



We can just compare P(M | L) and P(M | F)

We can use conditional probabilities for discrimination



	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

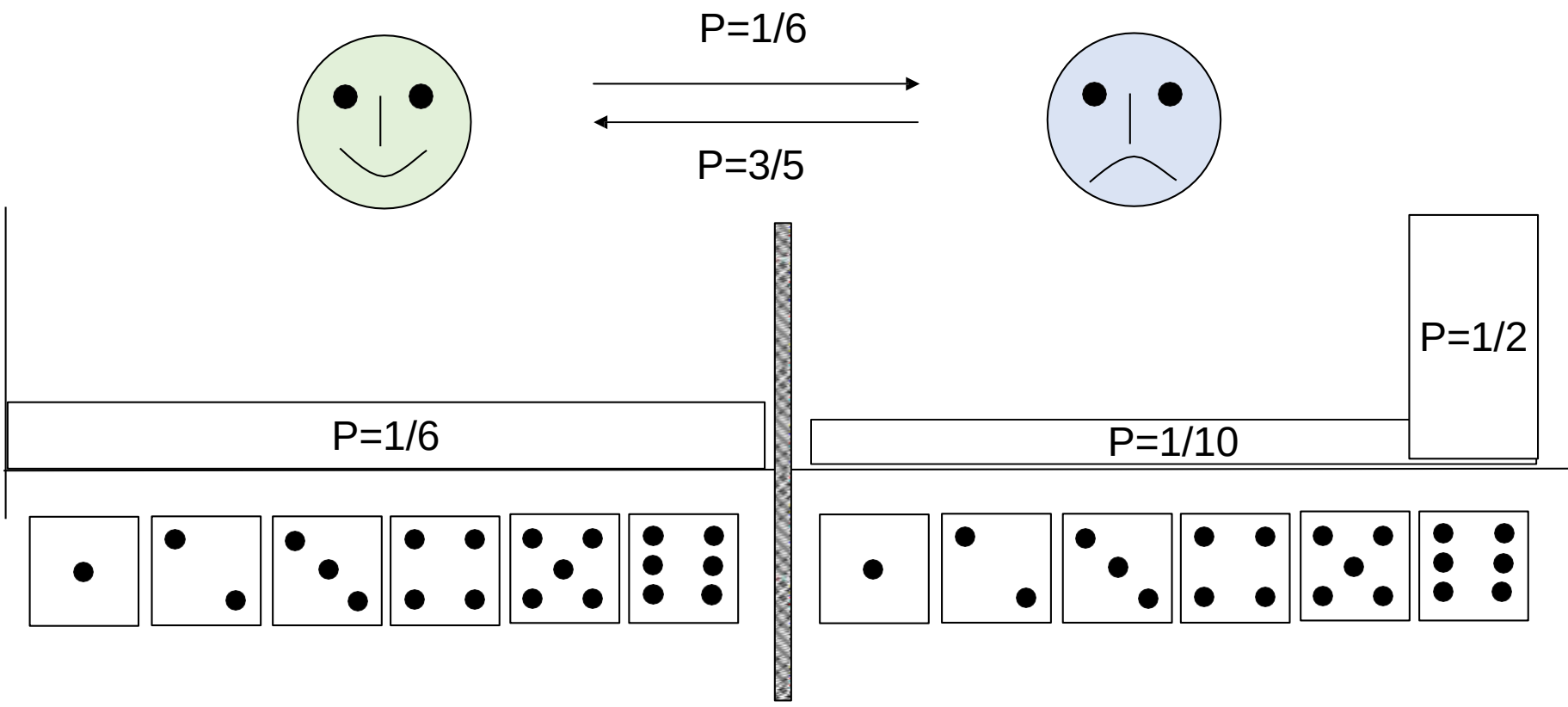
$P(M \mid L)=0.5*0.5*0.5*0.1*0.5*0.1=0.000625 = 6.25*10^{-4}$

$P(M \mid F)=0.17*0.17*0.17*0.17*0.17*0.17=0.000024 = 2.4*10^{-5}$

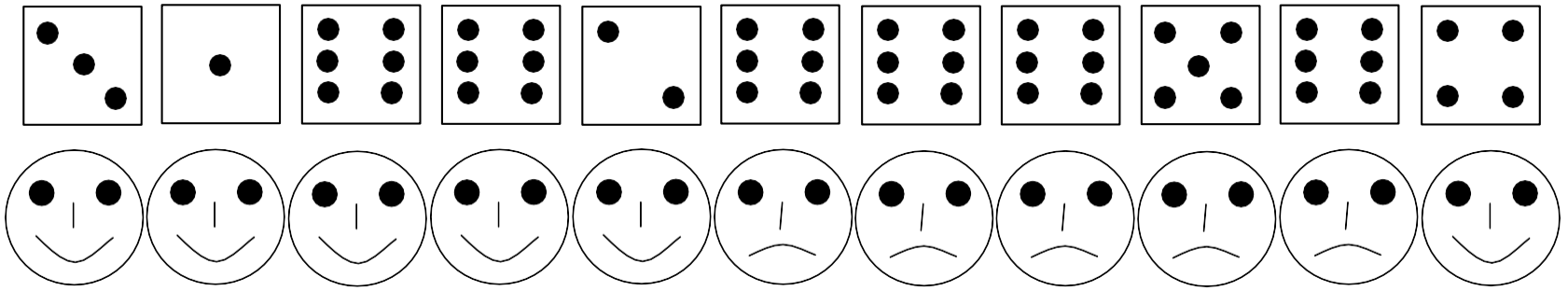
How confident we are that this sequence was produced by a loaded die? P(M and model L)/ P(M and model F)=25.89

Or log [P(M I model L)/ P(M | F)]=1.4 **Log-odds ratio**

The occasionally dishonest casino



Sequence generated by a model of an occasionally dishonest casino



Markov chains: recap

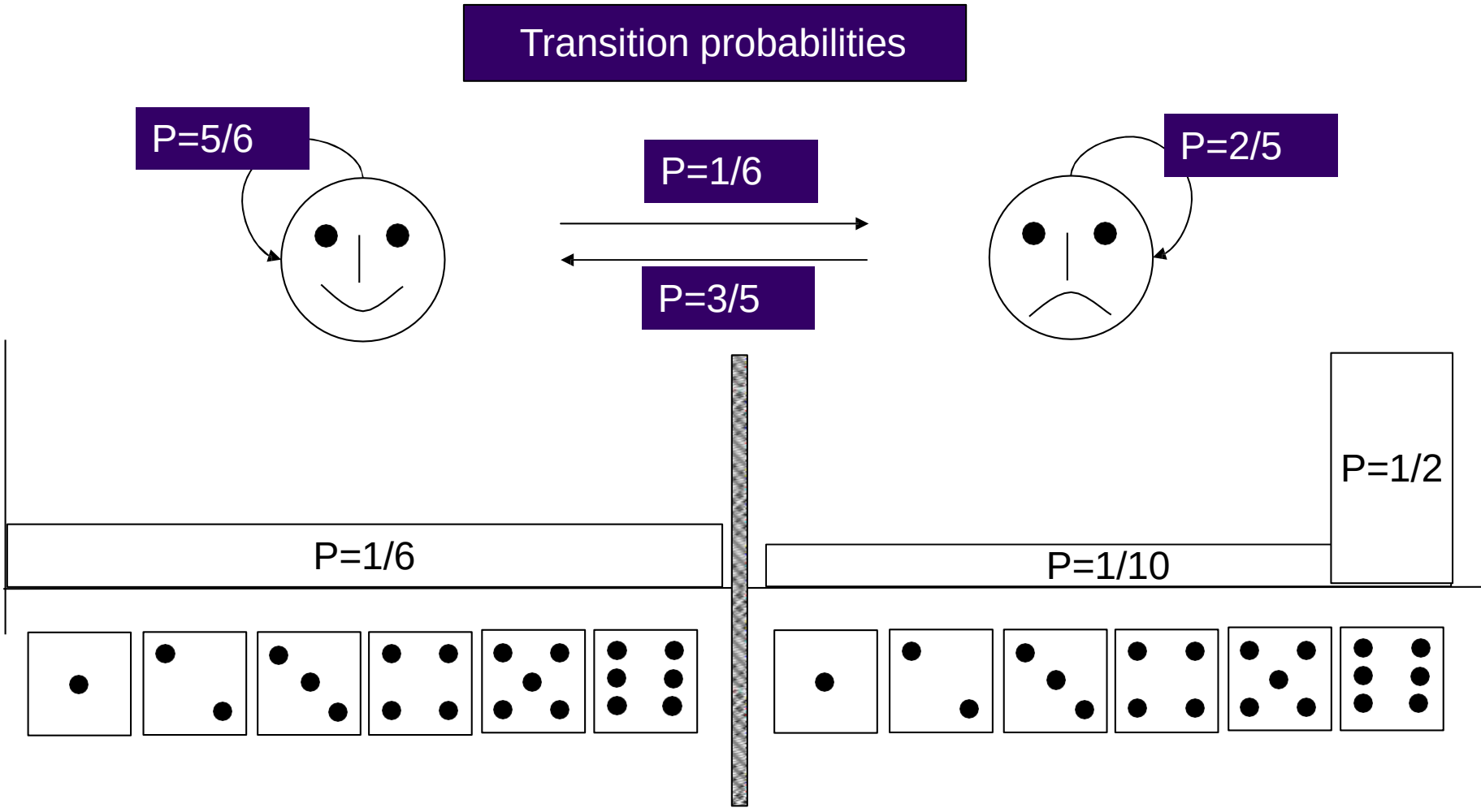
- The system can be in a finite number of states
- Transition from state to state is not predetermined, but rather is specified in terms of *probabilities*
- The transition probabilities depend only on the immediate history

- *The process of transitions from state to state* is called a ***Markov process*** or a ***Markov chain***

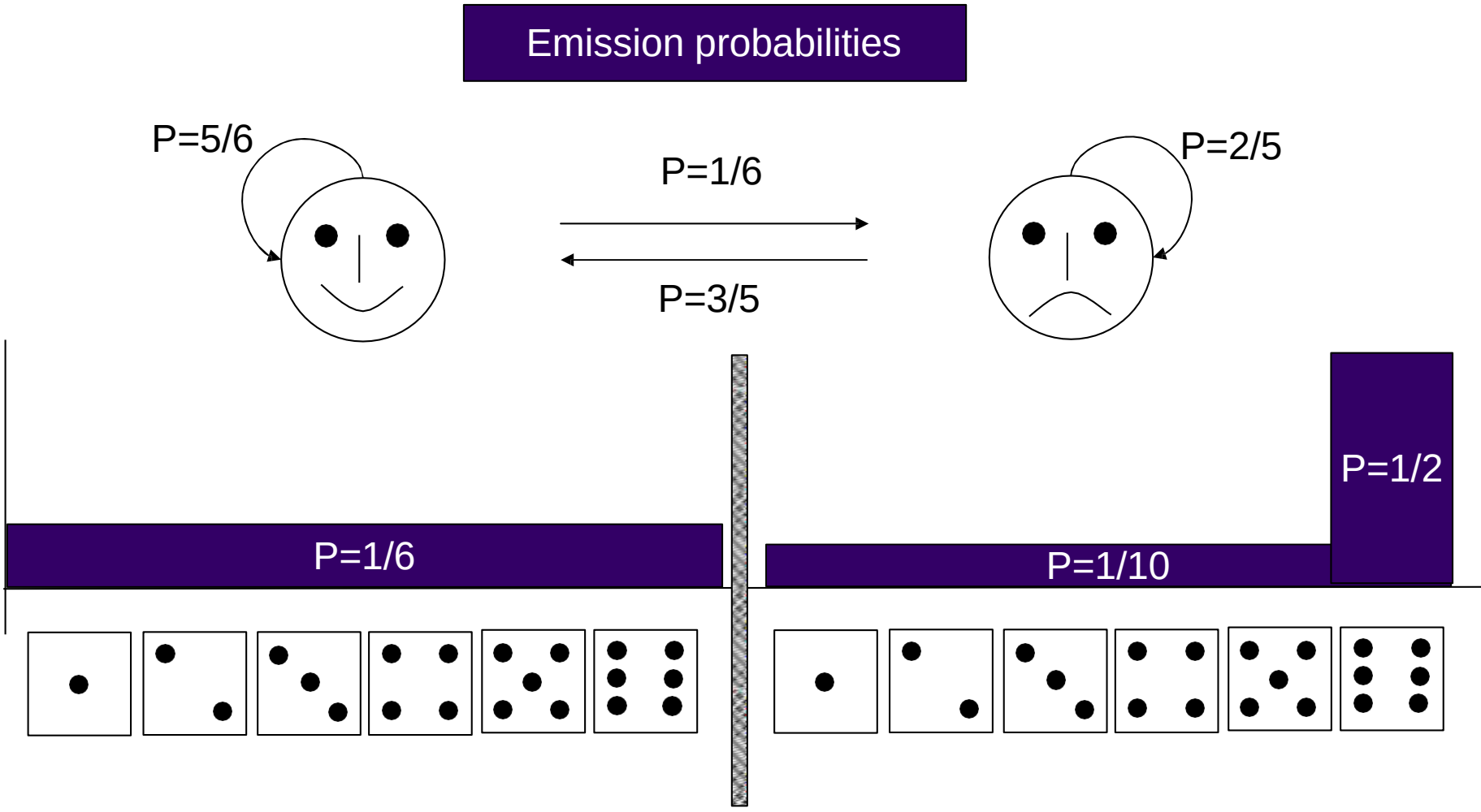
States can also behave probabilistically

- While in a particular state, system emits a symbol m_k from a finite alphabet with the probability $e_i(m_k)$, called *an emission probability* of symbol m_k in state W_i
- If we construct the schedule of observation times, and at each point in time record the symbols emitted by a system along with the state, we obtain 2 sequences:
  - the sequence of emitted symbols which is called *an observed sequence M*
  - the sequence of states π which is called a *path* through system states

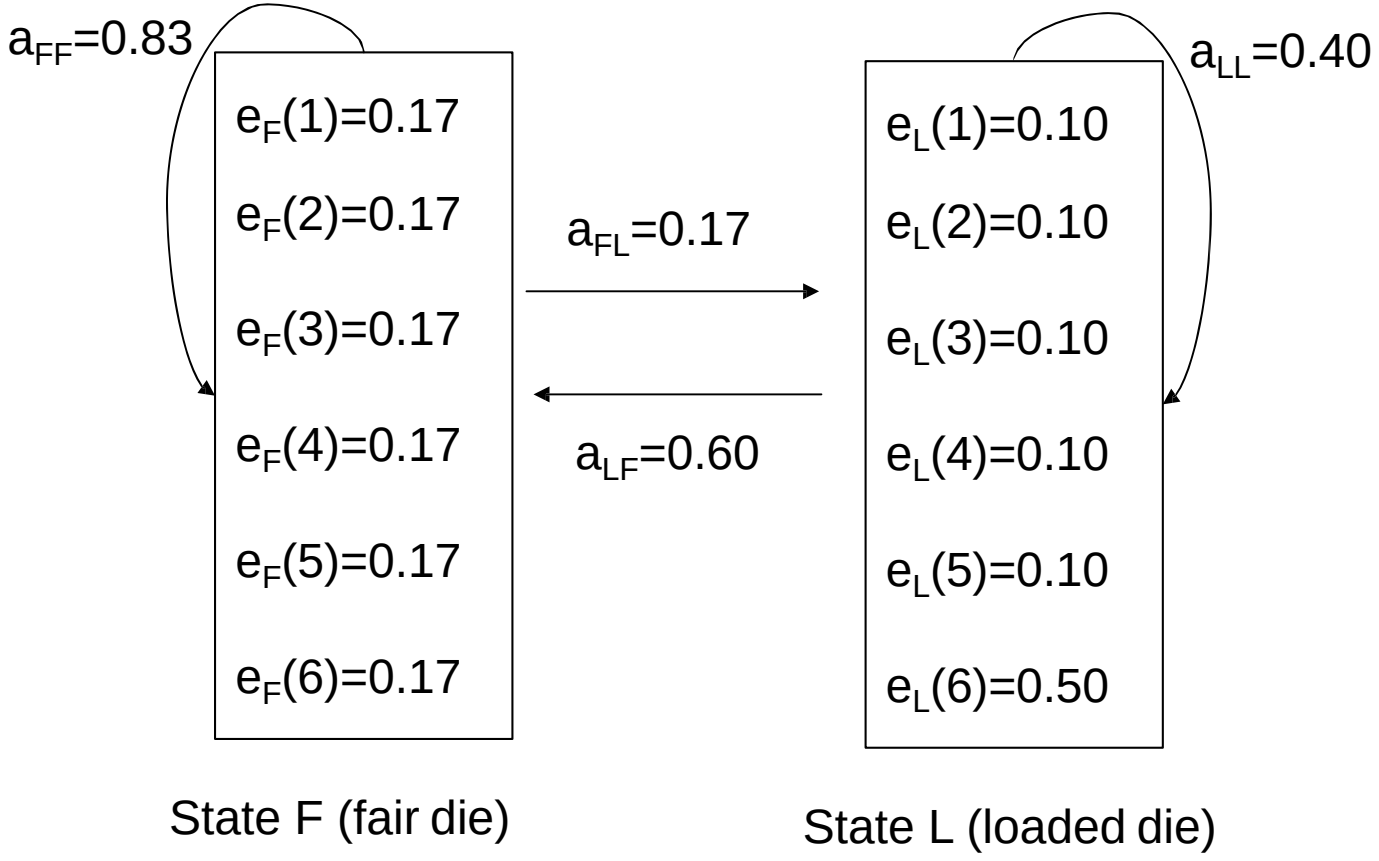
Terminology



Terminology



Transition and emission diagram



Tabular parameters

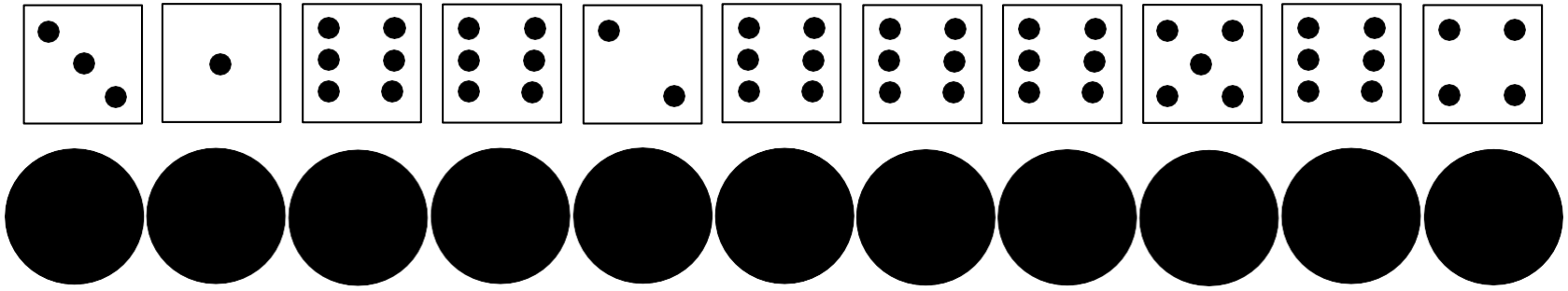
The state transition matrix

	F	L
F	0.83	0.17
L	0.60	0.40

Emission probabilities

	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

Hidden Markov Model (HMM)



States are unknown (hidden)

3 types of questions to HMM

1. Given a sequence of *N* observations, what is the probability of obtaining this sequence given a particular state path (**Sequence probability**)
2. Given a sequence of *N* observations, what is the most probable sequence of the underlying states (**Most probable *path***)
3. Given a sequence of N observations, what is the probability that the i-th observation was produced when the system was in state Wj

Question 1

Given a sequence and a path, what is the sequence probability?

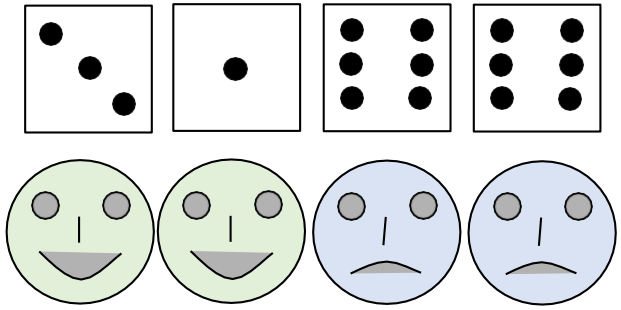
- The probability P(M| π) is the *conditional probability* that sequence M was generated while system was moving from state to state according to π

The probability that the sequence was generated following a path π

- Pick a path π
- Calculate a joint probability of π and M

	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

A suggested path



	F	L
F	0.83	0.17
L	0.60	0.40

P(M and π)=0.17 * 0.83 * 0.17 * 0.17 * 0.50 * 0.60 * 0.50=0.0006

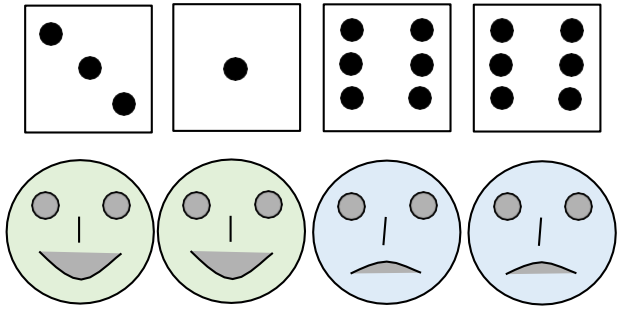
- Note that this is not P(π | M)

The probability that the sequence was generated following a path π when π is unknown (hidden)

- Pick a path π
- Calculate a joint probability of π and M

	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

A suggested path



	F	L
F	0.83	0.17
L	0.60	0.40

P(M and π)=0.17 * 0.83 * 0.17 * 0.17 * 0.50 * 0.60 * 0.50=0.0006

- Repeat **for each possible path** and choose a path which maximizes P(π and M).
- Total 2^N calculations (for 2 states and sequence of length N)

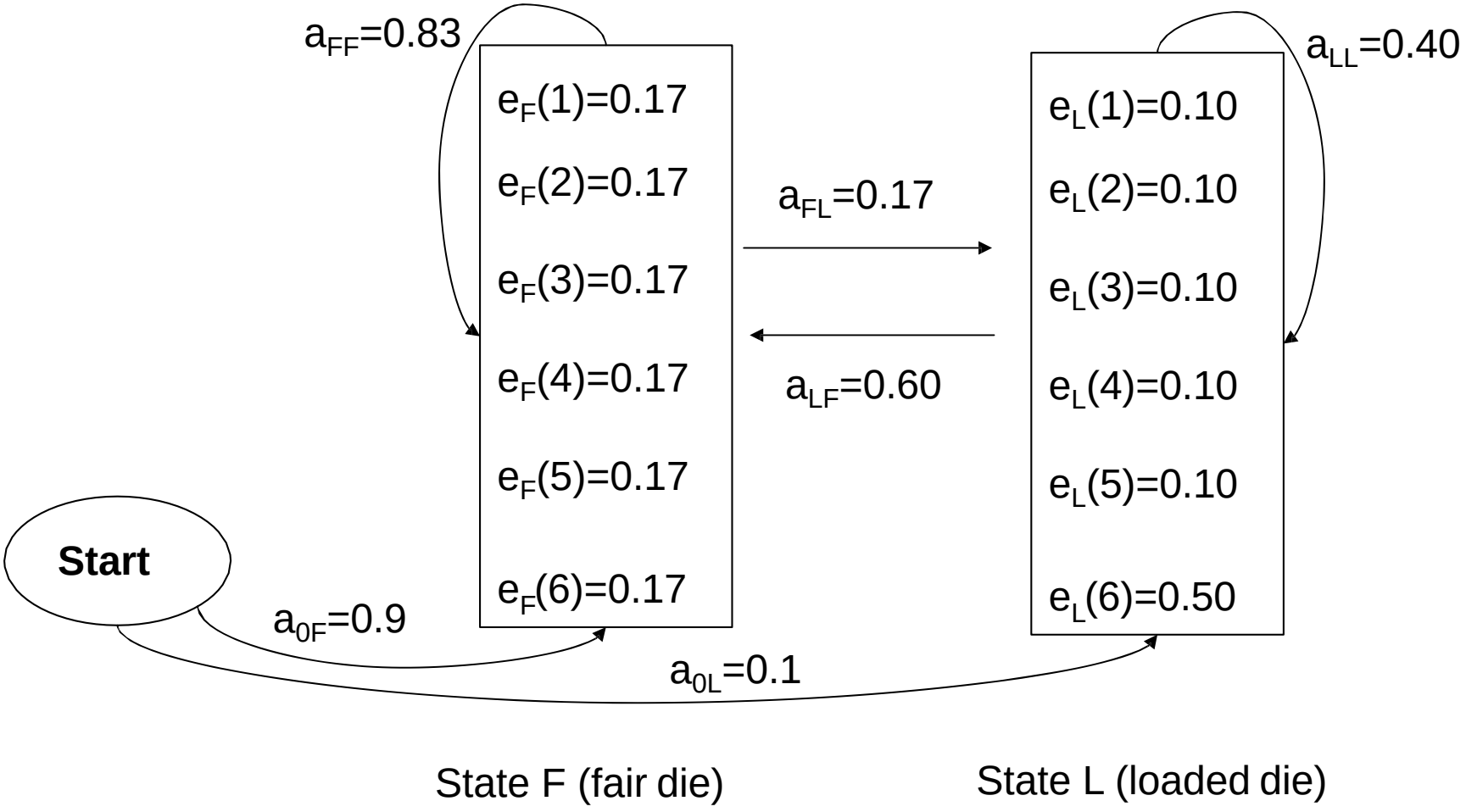
Question 2

Given only a sequence of observations, what is the most probable path?

Viterbi algorithm: dynamic programming

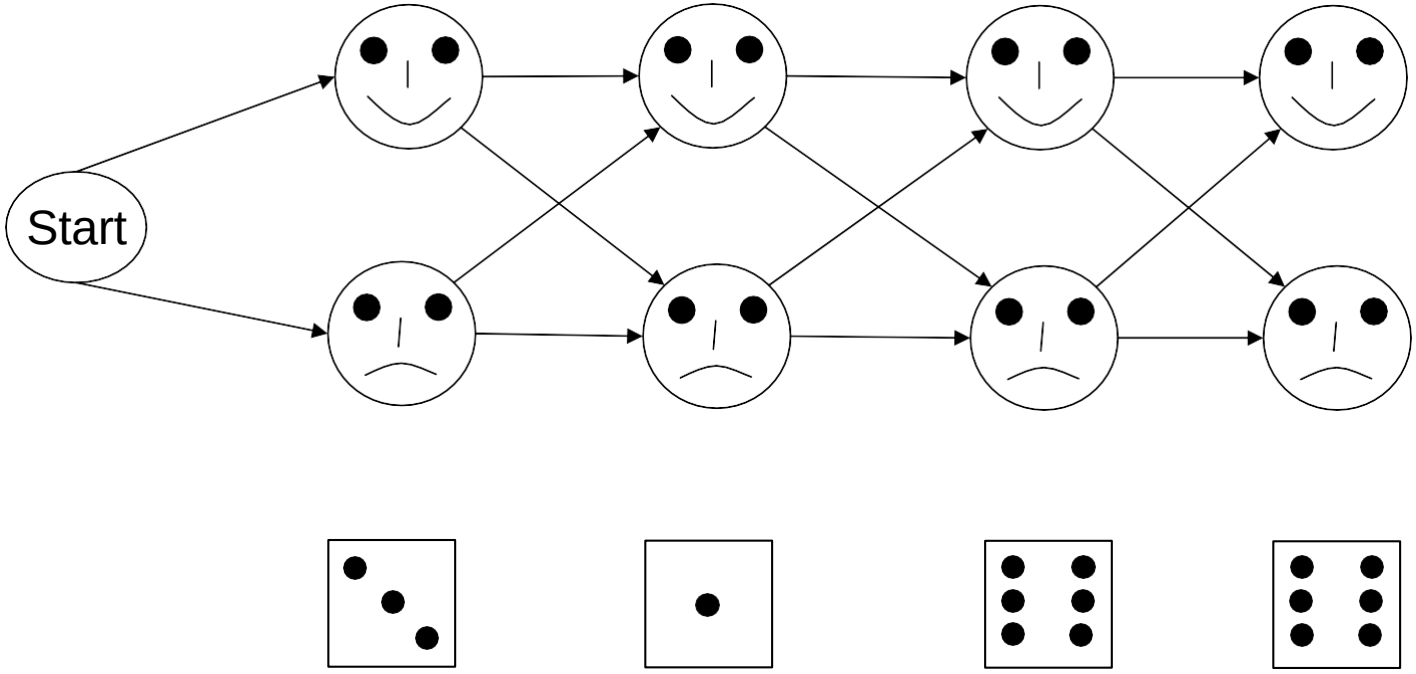
Dynamic programming. Initialization – the probability of choosing a die for the first time

- Add to the system a **start state** and parameters – the probabilities of choosing a fair or a loaded die in the beginning of a game



Dynamic programming. Initialization

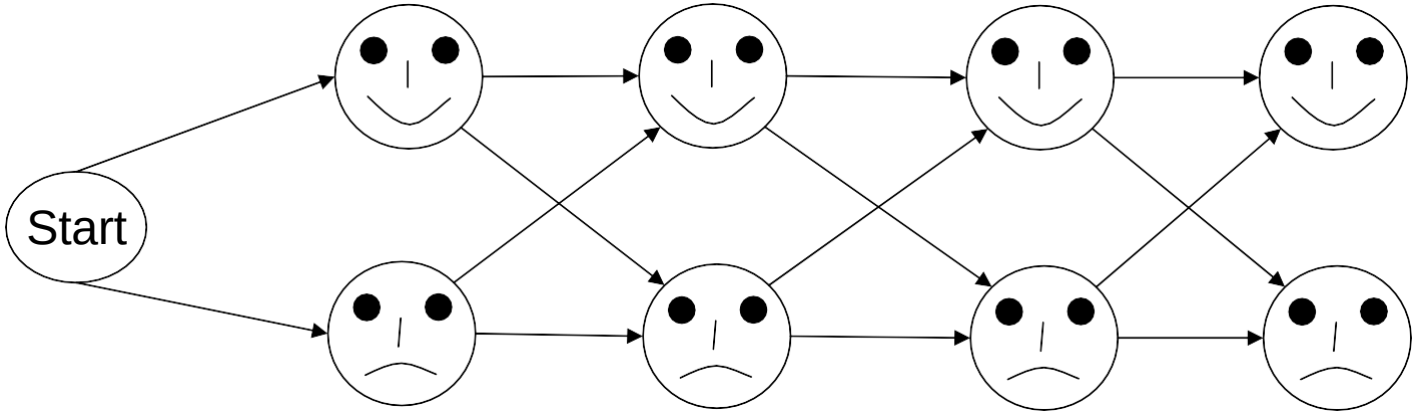
The graph of a process.



$P(\pi_{F,1}) = a_{0F} * e_F(M[1])$

$P(\pi_{L,1}) = a_{0L} * e_L(M[1])$

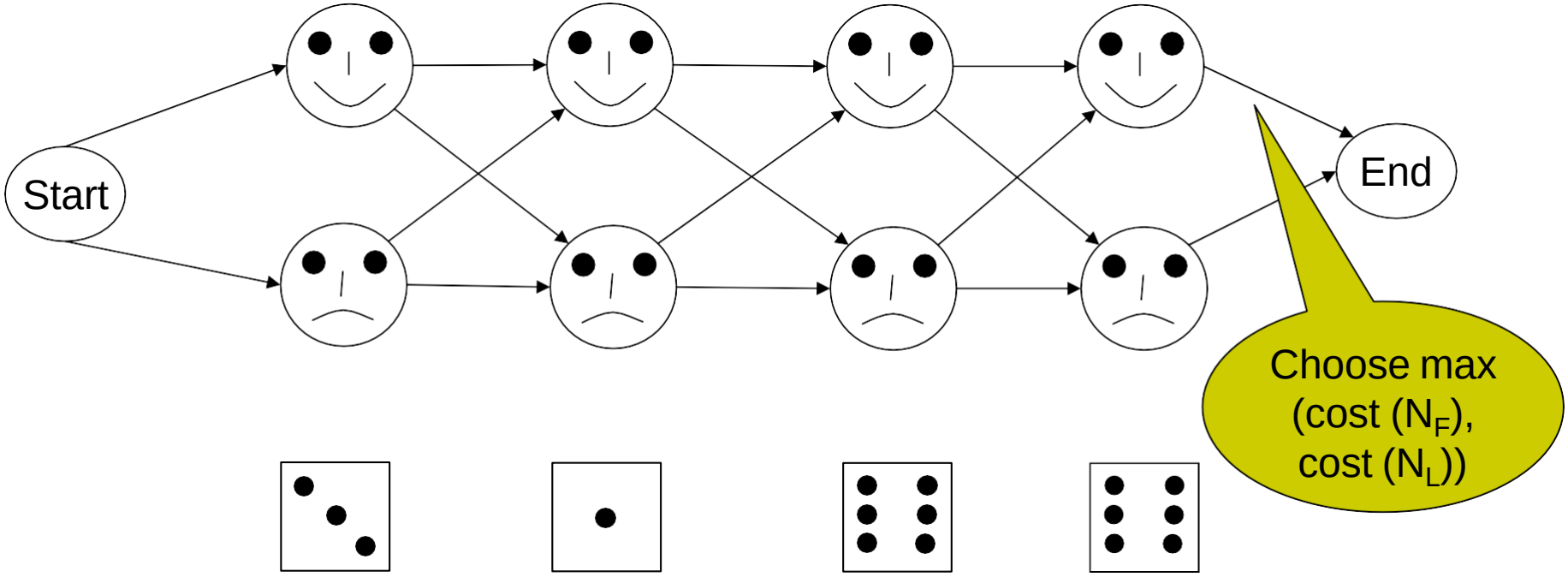
Dynamic programming.
Recursion



We are looking for a path which maximizes the probability of sequence M

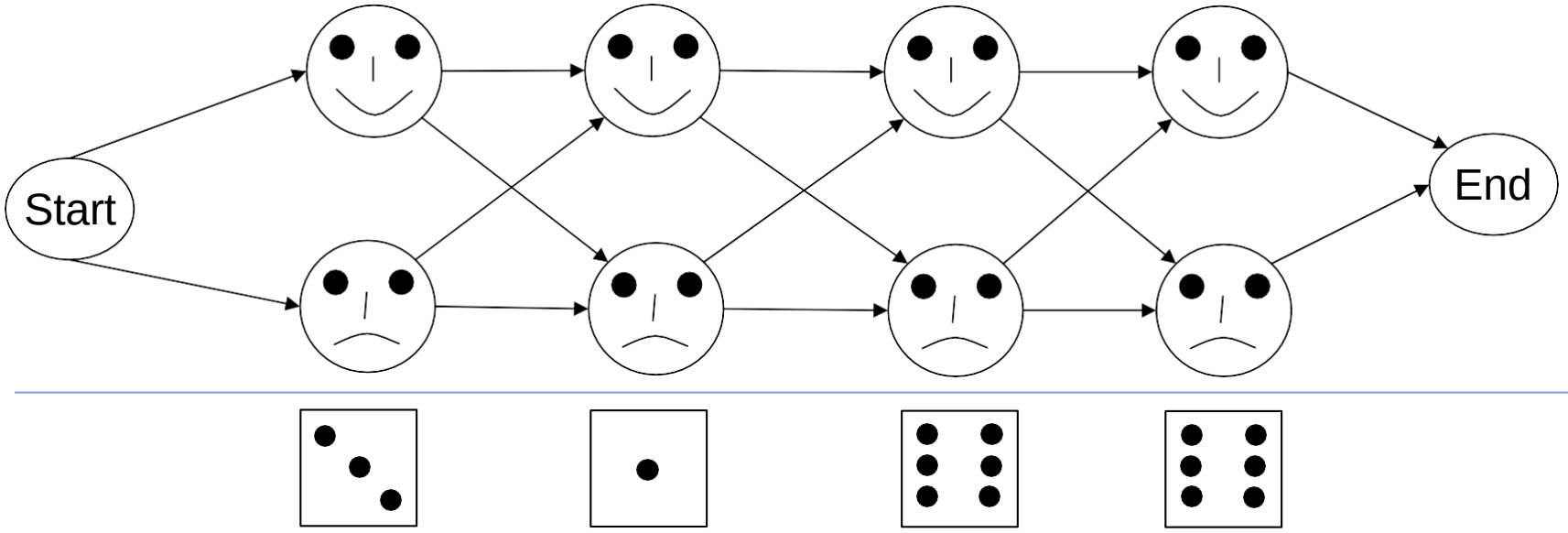
Dynamic programming. Recursion

If we know the best paths ending at states L and F in position 4, we can choose max between them and terminate the program



Dynamic programming. Recursion

This can be repeated for each combination of a position in a sequence of observations and one of 2 states



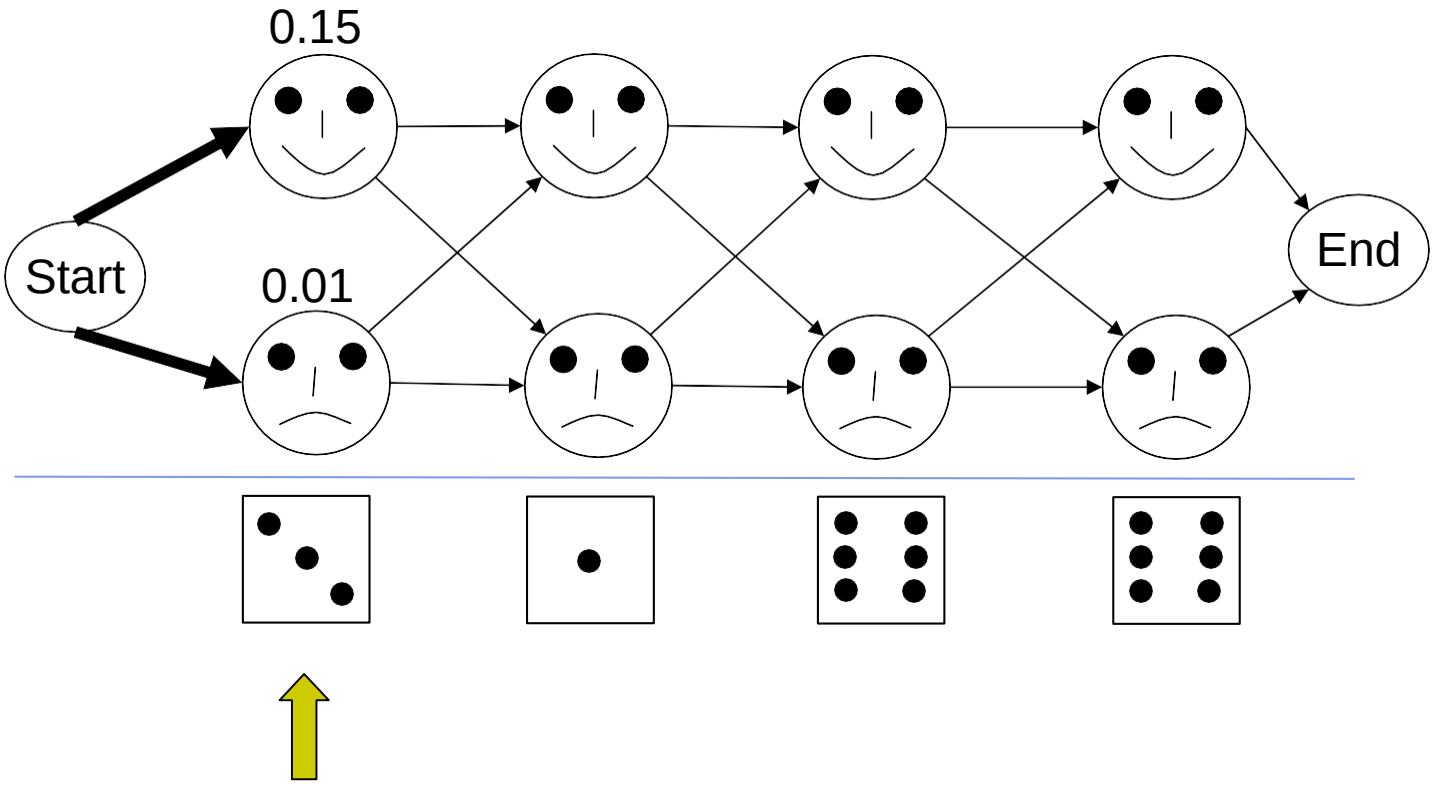
$$P(\pi_{F,i+1}) = \max \{P(\pi_{F,i}) * a_{FF}, P(\pi_{L,i}) * a_{LF}\} * e_F(M[i+1])$$

$$P(\pi_{L,i+1}) = \max \{P(\pi_{L,i}) * a_{LL}, P(\pi_{F,i}) * a_{FL}\} * e_L(M[i+1])$$

$$P(\pi^*) = \max \{P(\pi_{F,N}), P(\pi_{L,N})\}$$

Note: the probabilities are ***multiplied***, not added up

Viterbi algorithm. Demo 1

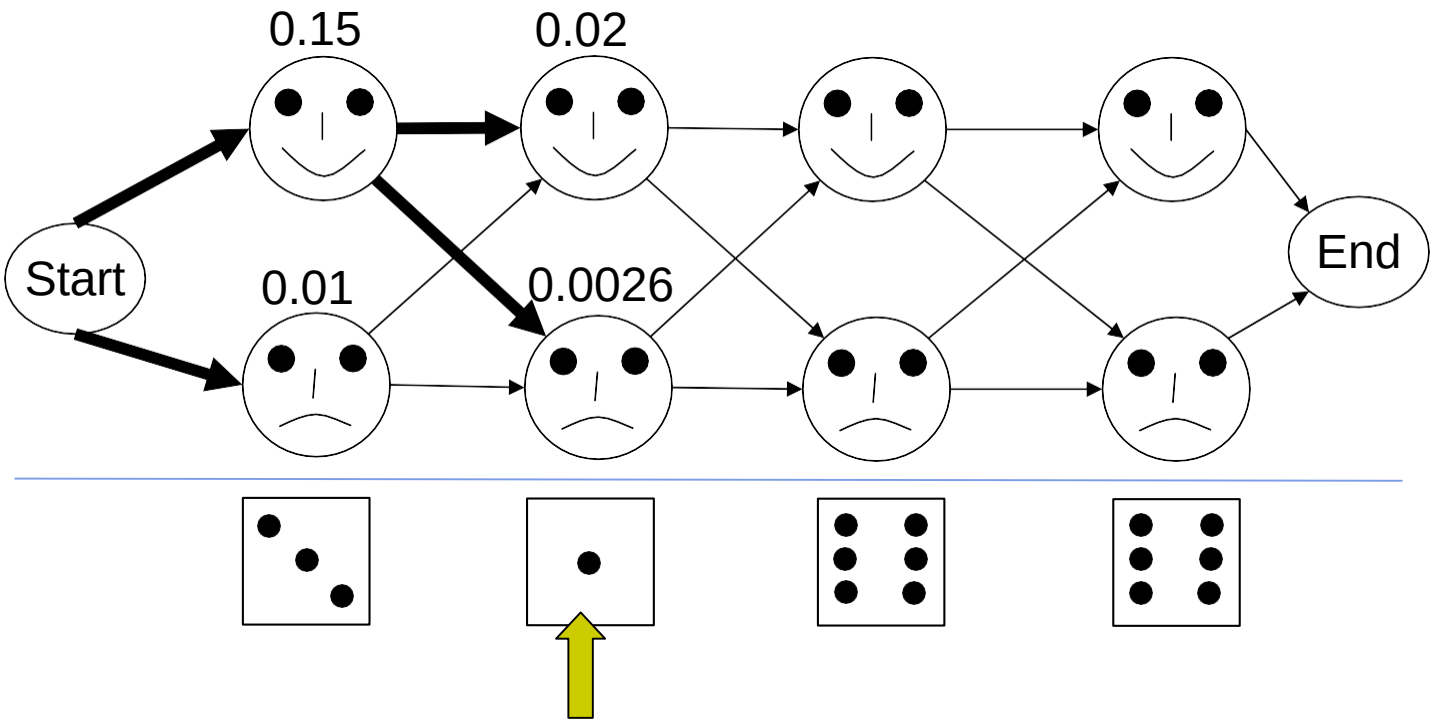


	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

	F	L
F	0.83	0.17
L	0.60	0.40
0	0.90	0.10

We have reached position i=1 with the probability 0.9*0.17 of going to the F state and emitting 3, and with probability 0.1*0.10 of going to the L-state and emitting 3. There are no other possibilities

Viterbi algorithm. Demo 2



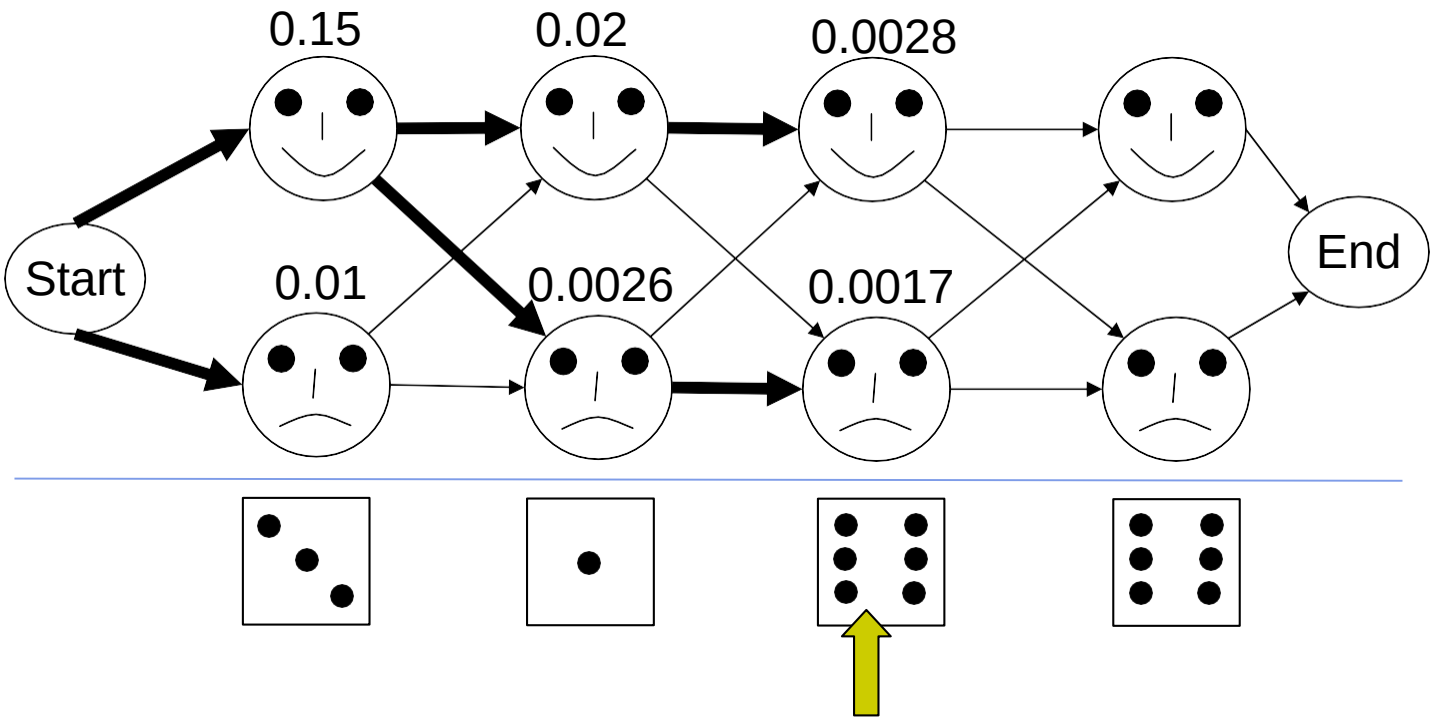
	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

	F	L
F	0.83	0.17
L	0.60	0.40
0	0.90	0.10

We can reach position i=2 (F-state) with the probability 0.15*0.83*0.17 or with probability 0.01*0.6*0.10. We chose the max between these two: 0.15*0.83*0.17=0.002

The L-state in position i=2 can be reached with probability 0.01*0.40*0.10 or 0.15*0.17*0.10=0.0026. The second is larger so we choose it.

Viterbi algorithm. Demo 3



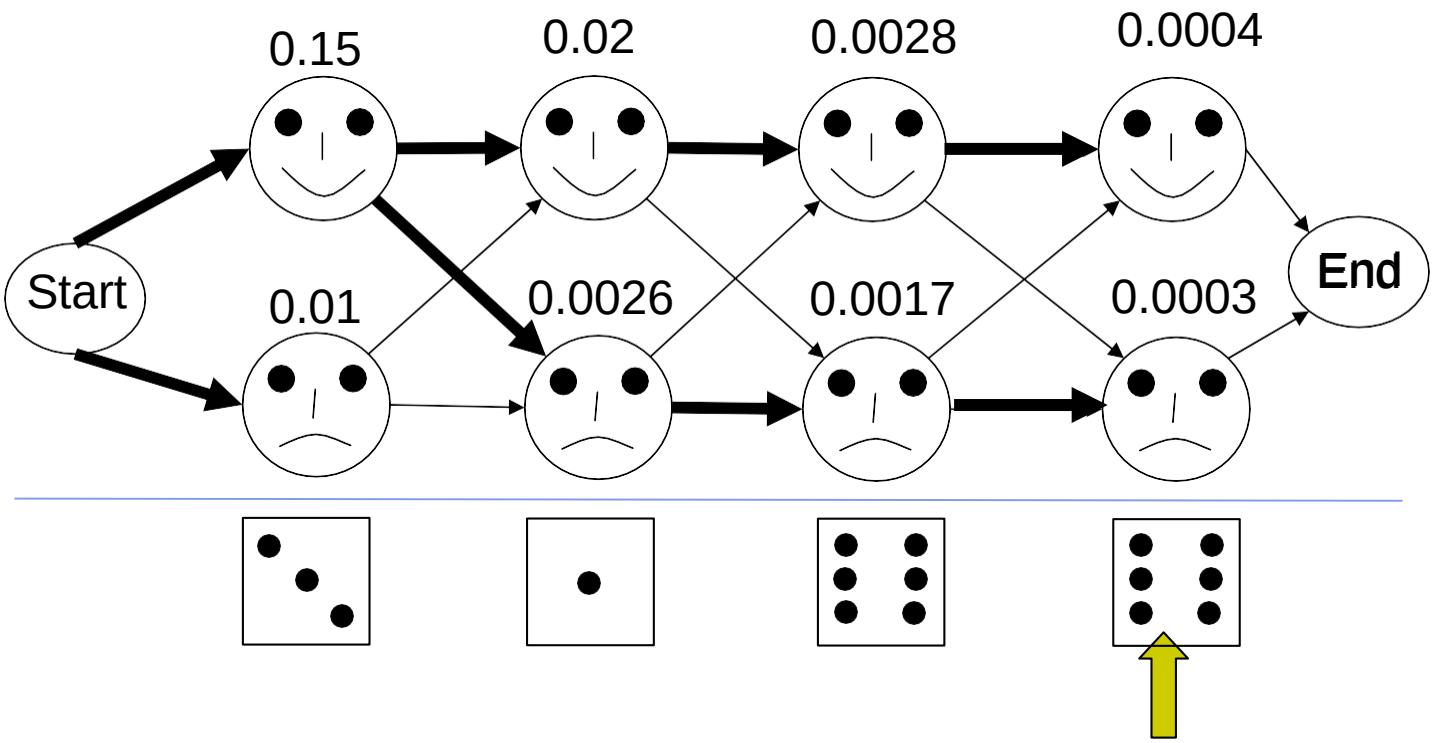
	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

	F	L
F	0.83	0.17
L	0.60	0.40
0	0.90	0.10

We can reach position i=3 (F-state) with the probability 0.02*0.83*0.17=0.0028 or with probability 0.0026*0.4*0.17=0.00018. We chose the max between these two: 0.02*0.83*0.17=0.0028

The L-state in position i=3 can be reached with probability 0.02*0.17*0.50=0.0017 or 0. 0026*0.4*0.5=0.0017. We chose the second - arbitrarily

Viterbi algorithm. Demo 4



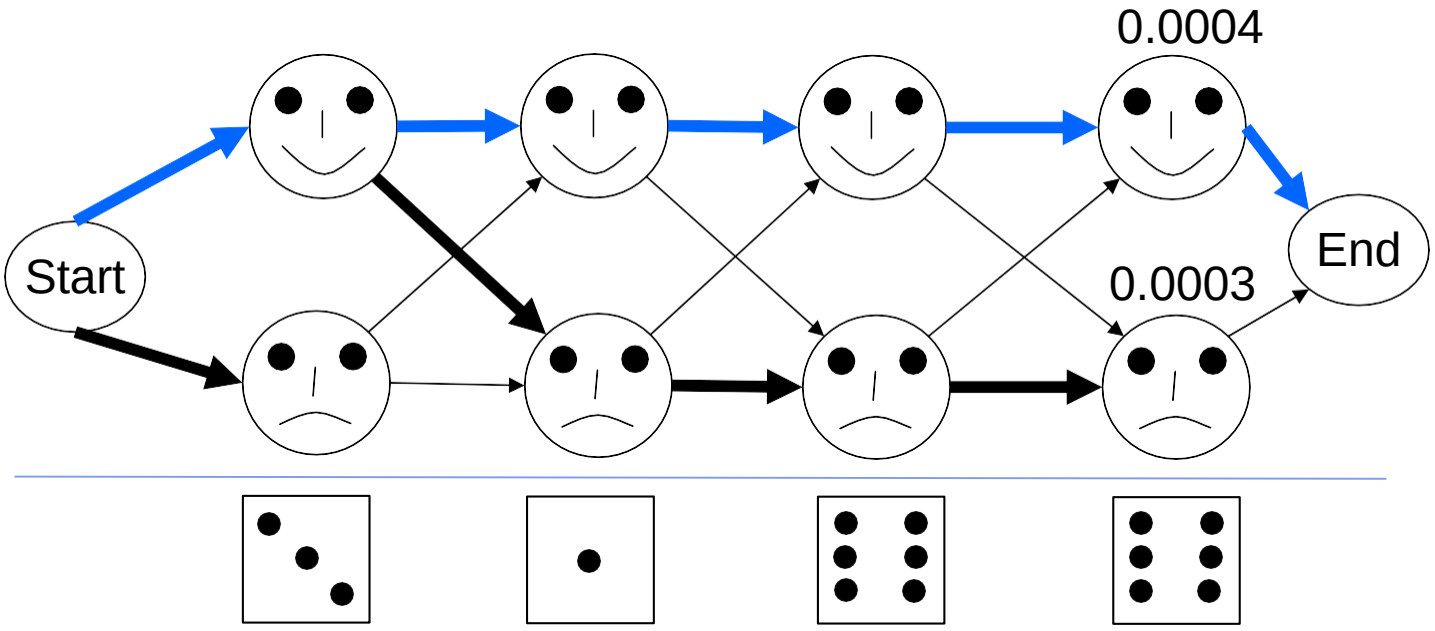
	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

	F	L
F	0.83	0.17
L	0.60	0.40
0	0.90	0.10

We can reach position i=4 (F-state) with the probability 0.0028*0.83*0.17=0.0004 or with probability 0.0017*0.6*0.17=0.00017. We chose the max between these two: 0.0028*0.83*0.17=0.0004

The L-state in position i=4 can be reached with probability 0.0017*0.40*0.50=0.00034 or 0.0028*0.17*0.5 =0.00024. We chose the max: 0.0017*0.40*0.50=0.00034

Viterbi algorithm. Demo - end



	F	L
1	0.17	0.10
2	0.17	0.10
3	0.17	0.10
4	0.17	0.10
5	0.17	0.10
6	0.17	0.50

	F	L
F	0.83	0.17
L	0.60	0.40
0	0.90	0.10

Choose max: 0.0004. So, the most probable sequence of states:

FFFF

Evidently, it is not enough to have 2 sixes in a row in order to be able to spot the loaded die.

Viterbi algorithm. Log-values

$P(\pi_{F,1}) = a_{0F} * e_F(M[1])$ $\quad$ $P(\pi_{L,1}) = a_{0L} * e_L(M[1])$

$P(\pi_{F,i+1}) = \max \{ P(\pi_{F,i}) * a_{FF}, P(\pi_{L,i}) * a_{LF} \} * e_F(M[i+1])$

$P(\pi_{L,i+1}) = \max \{P(\pi_{L,i}) * a_{LL}, P(\pi_{F,i}) * a_{FL}\} * e_L(M[i+1])$

$P(\pi^*) = \max \{P(\pi_{F,N}), P(\pi_{L,N})\}$

In order to avoid the underflow errors, in practice *log* is used instead of the actual probabilities

$P(\pi_{F,1}) = \log a_{0F} + \log e_F(M[1])$ $\quad$ $P(\pi_{L,1}) = \log a_{0L} + \log e_L(M[1])$

$P(\pi_{F,i+1}) = \max \{P(\pi_{F,i}) + \log a_{FF}, P(\pi_{L,i}) + \log a_{LF}\} + \log e_F(M[i+1])$

$P(\pi_{L,i+1}) = \max \{P(\pi_{L,i}) + \log a_{LL}, P(\pi_{F,i}) + \log a_{FL}\} + \log e_L(M[i+1])$

$P(\pi^*) = \max \{P(\pi_{F,N}), P(\pi_{L,N})\}$

How good is the prediction

Overall, an underlying hidden pathway explains the given sequence well – the path explanation obtained with Viterbi is good

Already we can answer:

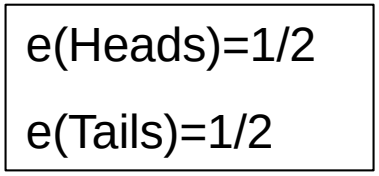
- What is the probability that a given sequence of observations came from a particular HMM
- Where in the sequence the model has probably changed

Exercise 1. Markov models

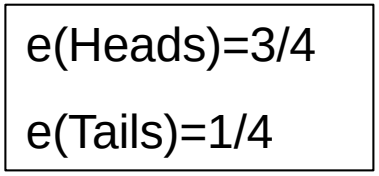
- In Vancouver, if it rains today, then it rains tomorrow 3 times out of 5. If it is sunny today, it is also sunny tomorrow 1 time out of 3. Build a Markov model for the weather in Vancouver.

Exercise 2. Discrimination by probability

- Markov models for the honest and for the dishonest casino are presented below:



Fair coin



Biased coin

Given that is is equally probable to choose F or L, find out which coin has most probably produced the following sequence of observations:

HHHTTHT

Exercise 2. Is the coin biased?

- For sequence M of length N with *k* heads:

$P(M \mid \text{fair coin}) = \Pi_n(1/2) * P(F)/P(M) \sim 1/2^N$

$P(M \mid \text{biased coin}) = \Pi_k(3/4) * \Pi_{N-k}(1/4) * P(B)/P(M) \sim 3^k/4^k * 1/4^{N-k}$

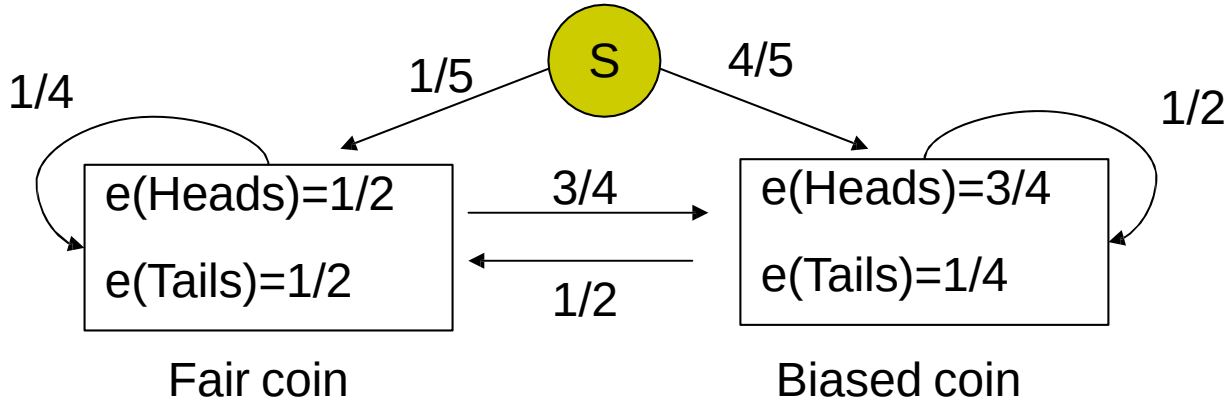
- For this simple example, we can compute how many heads out of N are needed to conclude that the coin is biased:
- when P(M and fair coin) < P (M and biased coin) ?

$1/2^N < 3^k/4^N$
$1 < 3^k/2^N$
$2^N < 3^k$

$N\log 2 < k\log 3$
$k > (\log 2/\log 3) * N$
$k > 0.63\ N$

Exercise 3.

- Using the Viterbi algorithm, find the most probable path of states for the following sequence given the HMM which produced this sequence.



Observed sequence: HTTHHH

Building a Hidden Markov Model

- 2 parts:
  - Model topology: what states there are and how are they connected
  - The assignment of parameter values: the transition and emission probabilities

Parameter estimation

- We are given a set of training sequences
- 2 cases:
  - When the states in the training sequences are known

    $a_{from,to} = count_{from,to} / \Sigma_x count_{from,x}$

    $e_{state\ i}(symbol\ j) = count_{state\ i}(symbol\ j) / \Sigma_y (symbol\ y \mid state_i)$
  - When the states are unknown
    - Viterbi training

Parameter estimation when the states are known - example

X	1	2	6	6	1	1	2
π	F	L	F	F	L	L	L

$e_F(3)=0$?

To avoid this, use *pseudocounts*

$e_F(1)=(1+1)/(3+6)$, 1 is a pseudocount, 6 is the number of different symbols

eF(1)=2/9

$e_F(2)=1/(3+6)=1/9$

$e_F(3)=1/(3+6)=1/9$

$e_F(4)=1/(3+6)=1/9$

$e_F(5)=1/(3+6)=1/9$

$e_F(6)=(2+1)/(3+6)=3/9$

$a_{F,L}=2/3$

$a_{F,F}=1/3$

$a_{L,F}=1/3$

$a_{L,L}=2/3$

Or with pseudocounts

$a_{F,L}=(2+1)/(3+2)=3/5$

$a_{F,F}=(1+1)/(3+2)=2/5$

$a_{L,F}=(1+1)/(3+2)=2/5$

$a_{L,L}=(2+1)/(3+2)=3/5$

Viterbi training for parameter estimation

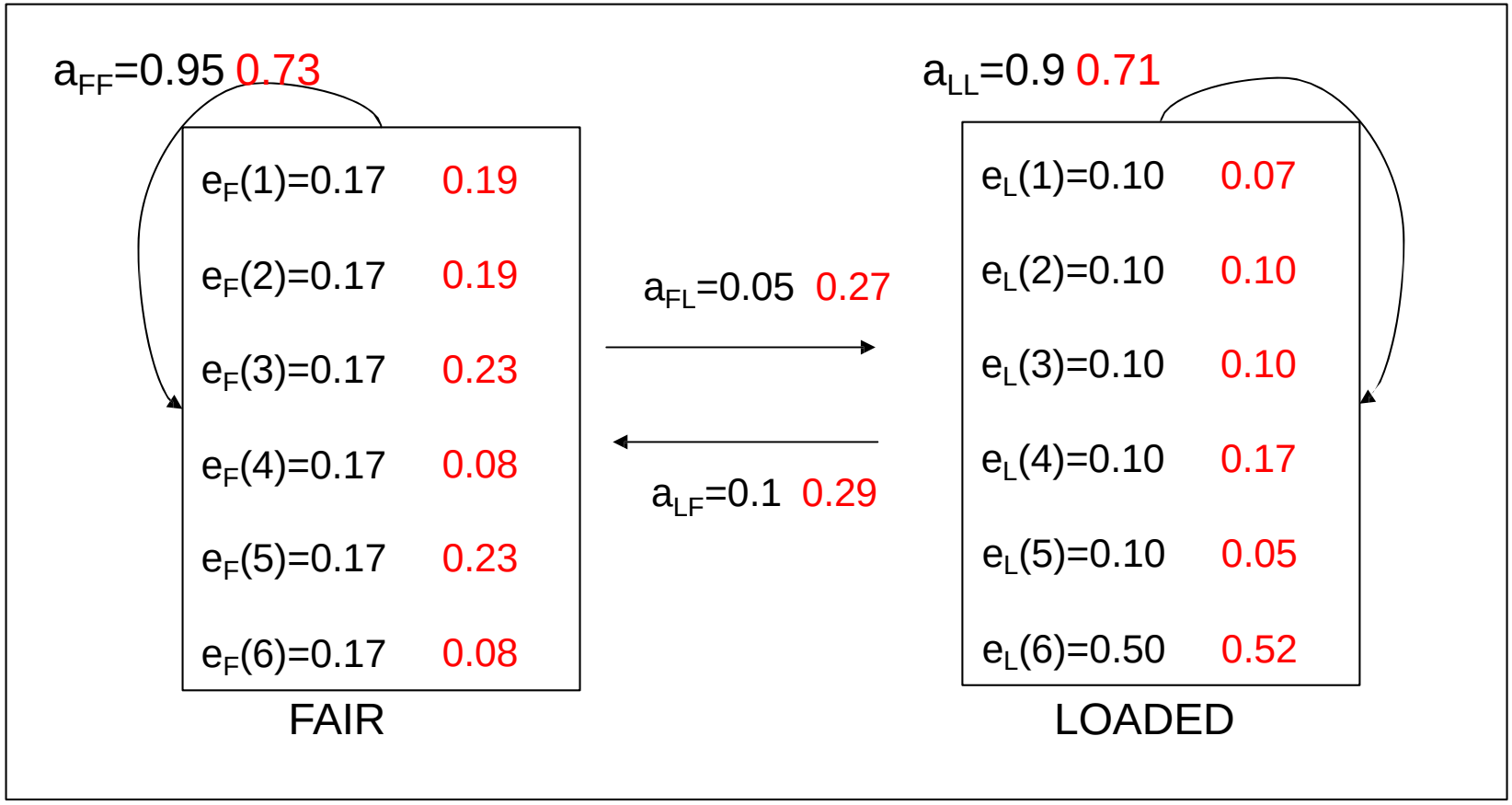
- Pick a set of random parameters
- Repeat
  - Find the most probable path of states according to this set of parameters
  - This path partitions the sequences into partitions according to the states
  - Calculate new set of parameters, now from the known states
- Until the path does not change anymore

Viterbi training

- The assignment of paths is a discrete process, thus the algorithm converges precisely
- When there is no path change, the parameters will not change either, because they are determined completely by the paths
- The algorithm maximizes the probability P(observed data| Θ, π^*)

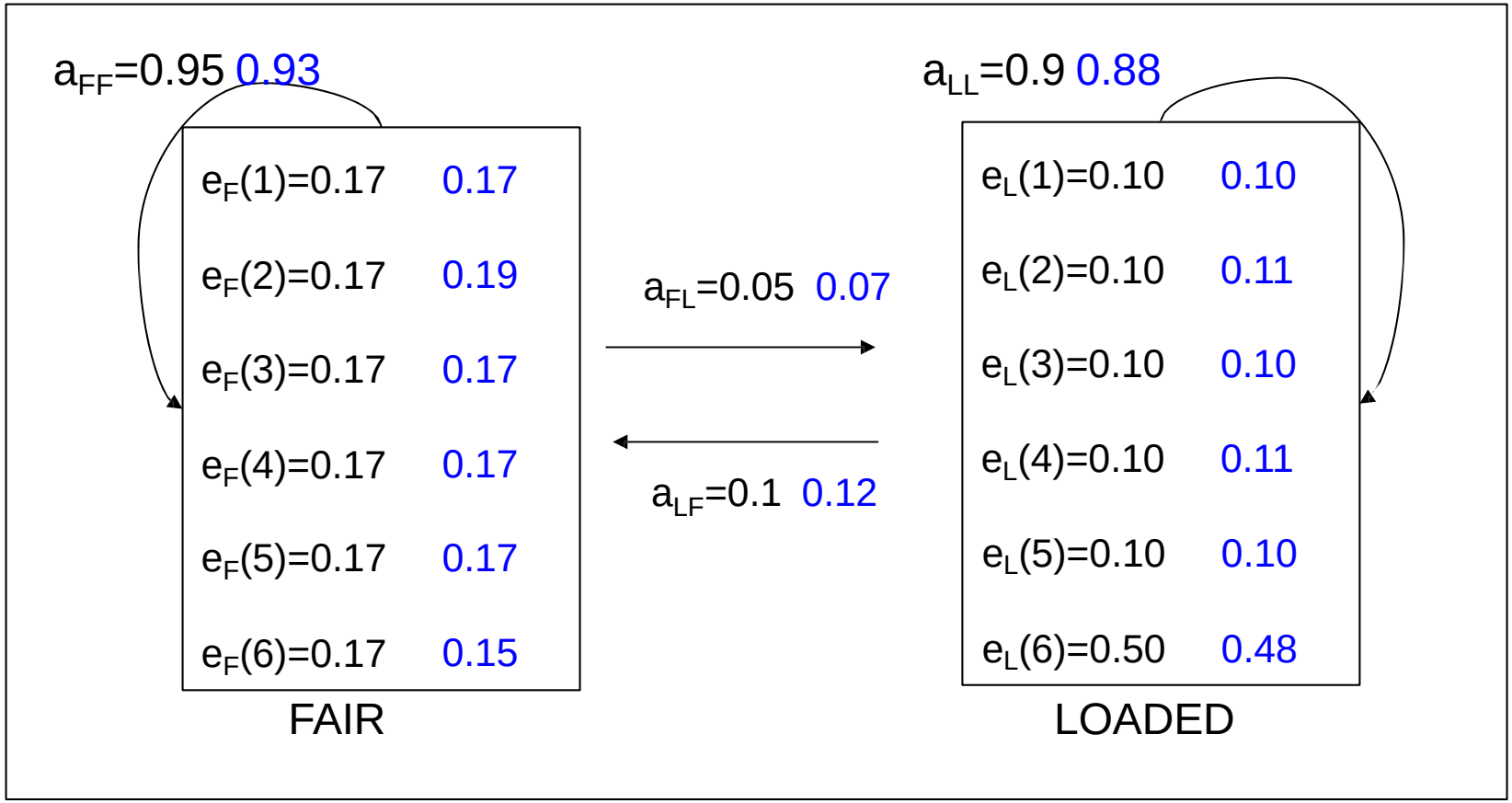
  and not P(observed data | Θ) which we ideally want

Parameter estimation – illustration 1



The parameters estimated for 300 random rolls and an iterative process started from randomly selected parameters

Parameter estimation – illustration 2



The parameters estimated for 30 000 random rolls and an iterative process started from randomly selected parameters

HMM applications

- Robot planning + sensing when there's uncertainty
- Speech Recognition/Understanding
- Consumer decision modeling
- Economics & Finance
- Human Genome Project
- …

Classic example: Speech recognition

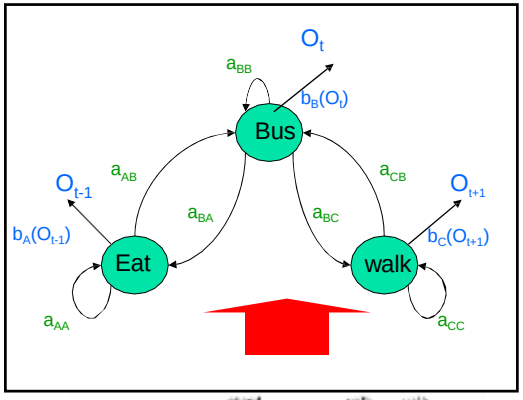
- Signal → words
  - Observable is signal
  - Hidden state is part of word

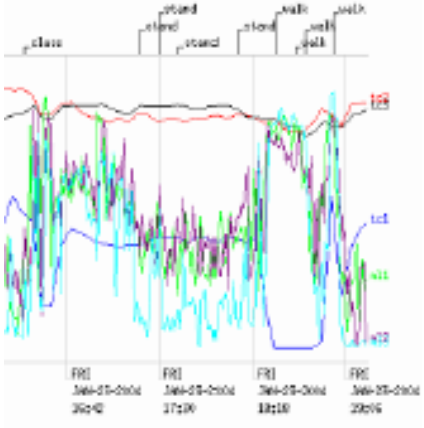
- Formulation:
  - What is the most probable word given this signal?

UTTERLY GROSS SIMPLIFICATION

In practice: many levels of inference; not only HMM

Human daily activities recognition from wearable sensor signals





Bio-application 1. Gene finding

CpG islands

- C nucleotide followed by G is easily methylated
- Methylated C easily becomes T
- The methylation is suppressed in important regulatory regions – around promoters (starting sites of transcription)
- Thus, an overall low frequency of C->G di-nucleotide is significantly increased in the gene promoter regions

Biological questions

- Given a short stretch of DNA sequence, determine whether it came from a CpG island or not
- Given a long un-annotated DNA sequence, find CpG islands in it

Transition probability estimation: from real DNA sequences

From 48 **known** CpG islands of a total length 60,000 nucleotides, and from regular DNA stretches:

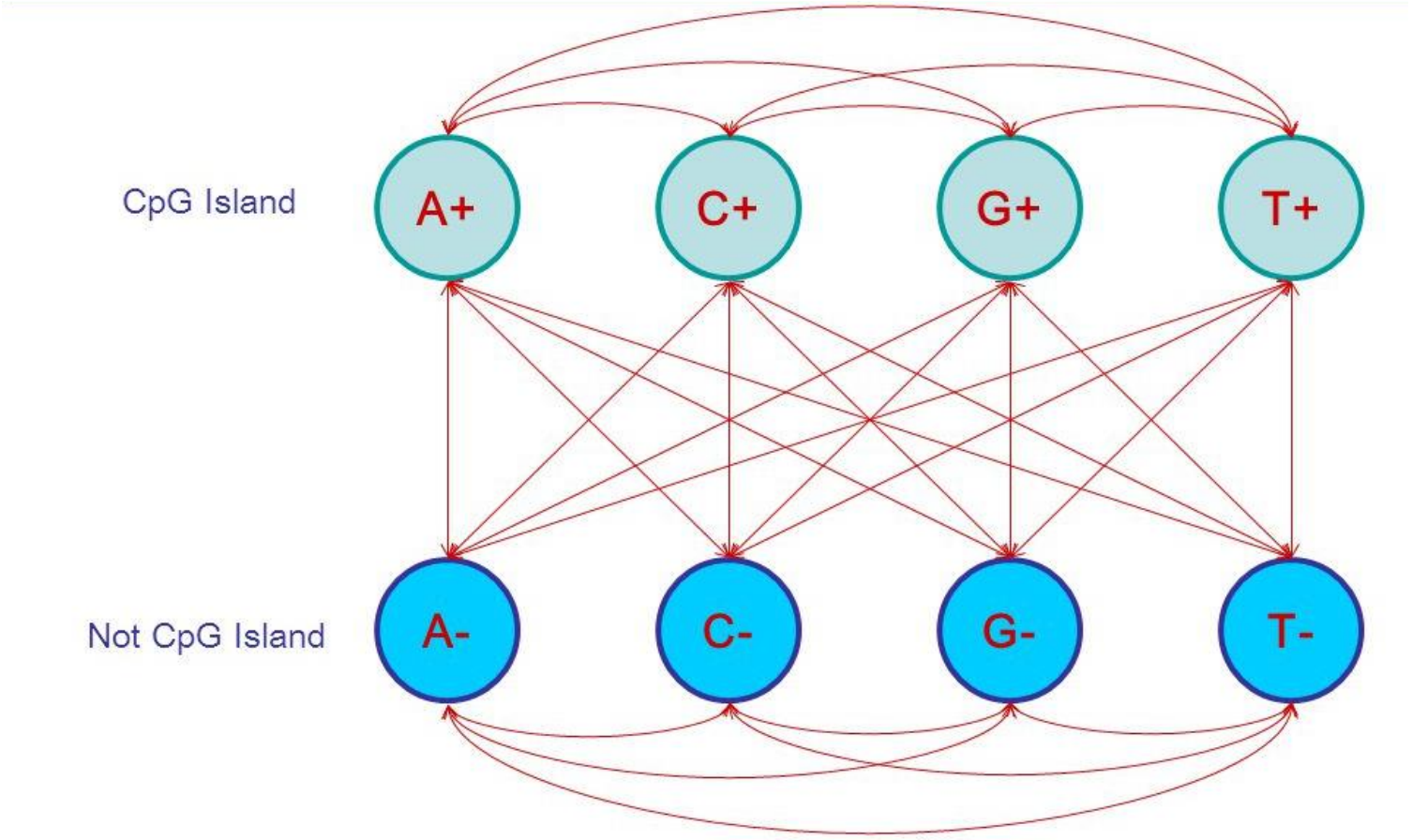
the transition probabilities for each pair of nucleotides were estimated (expected 0.25 if at random)

+	A	C	G	T
A	0.18	0.27	0.43	0.12
C	0.17	0.37	0.27	0.19
G	0.16	0.34	0.38	0.12
T	0.08	0.36	0.38	0.18

-	A	C	G	T
A	0.30	0.20	0.29	0.21
C	0.32	0.30	0.08	0.30
G	0.25	0.25	0.30	0.20
T	0.18	0.24	0.29	0.29

$$a_{from,to} = count_{from,to} / \Sigma_x count_{from,x}$$

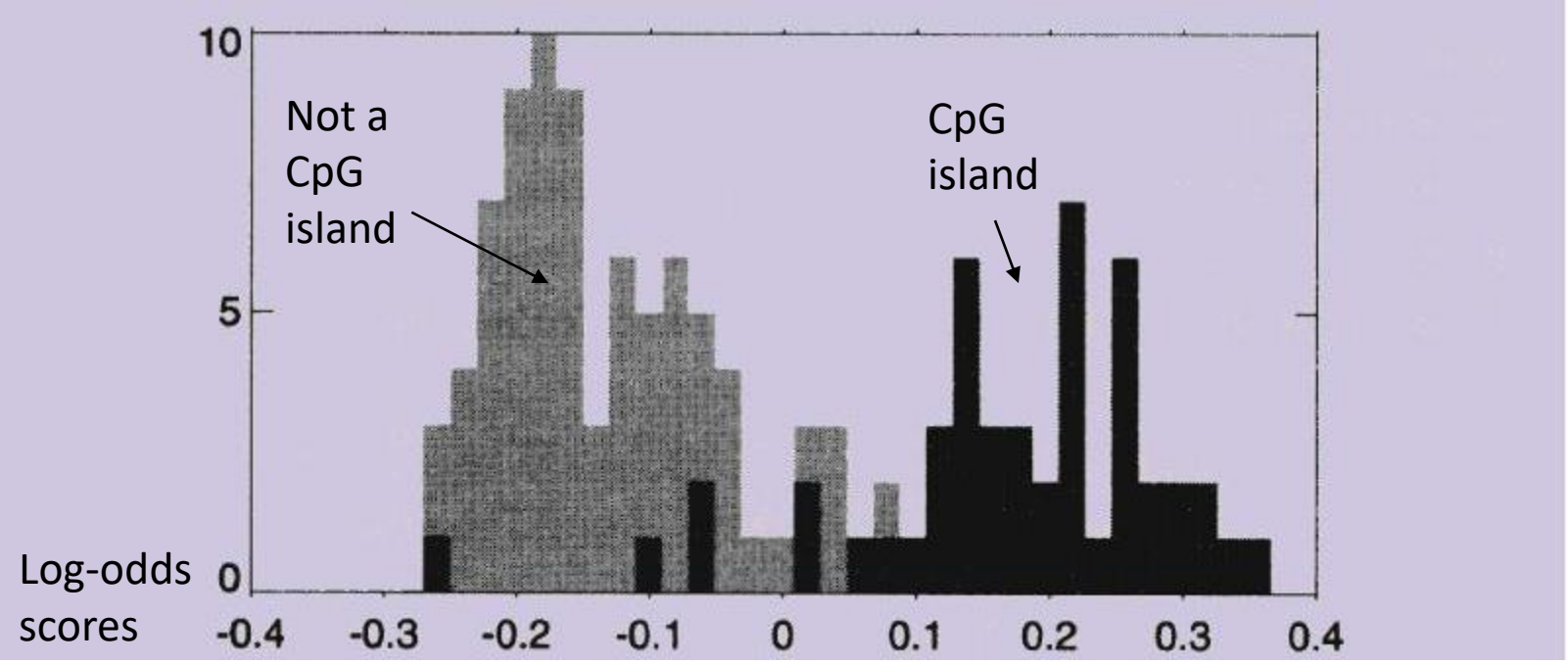
Markov model for DNA sequence



Am I in the CpG island?

To use these (+) and (-) models for discrimination for a given sequence we calculate the log-odds ratio:

- **Score(M)=log [P(M|given model +)/P(M|given model -)]**
- If this value is positive, we are in the CpG island, if not, we are not



Model efficiency: results of tests on another set of labeled DNA sequences

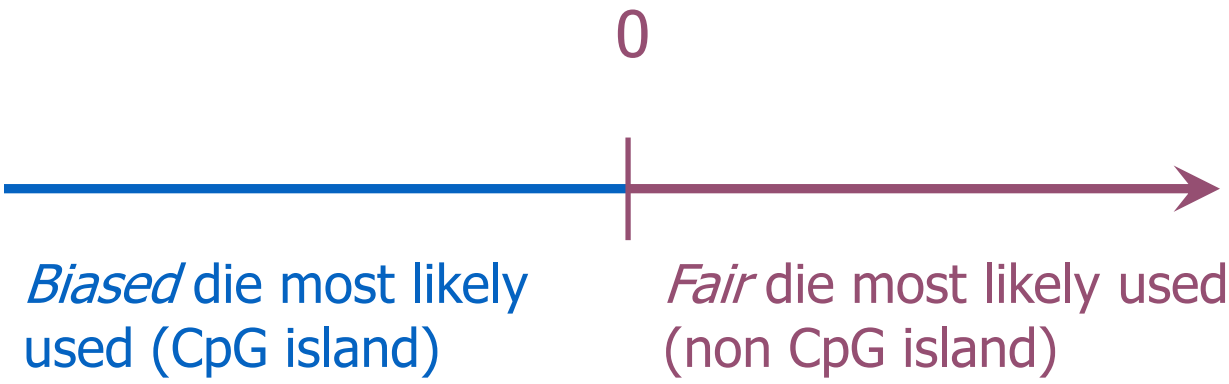
Finding CpG islands - HMM

- HMM: the essential difference from a simple Markov chain is that there is no one-to-one correspondence between the states and the symbols
- By looking at a *single symbol*, there is no way to tell whether it came from state C+ or C-

Computing Log-odds Ratios in a sliding window

$$x_1 x_2 \boxed{x_3 x_4 x_5 x_6 x_7} x_8 \ldots x_n$$

- Consider a *sliding window* of the outcome sequence
- Find the log-odds for this short window



Disadvantages:

- the length of CpG-island is not known in advance
- different windows may classify the same position differently

The most probable path through the sequence of states

The most probable path for sequence ***CGCG***

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

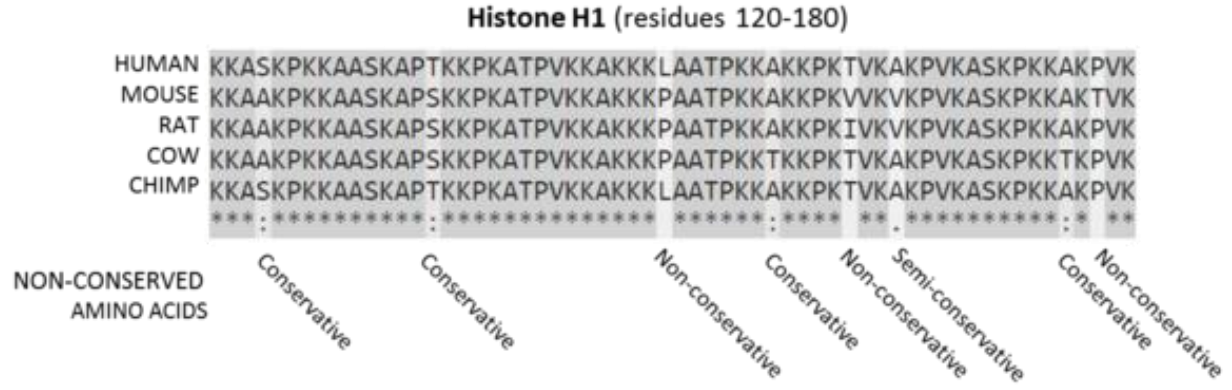
When we apply the Viterbi algorithm to a long un-annotated DNA sequence, the states will switch between + and -, giving suggested boundaries for CpG islands

Bio-application 2. Aligning a given sequence to a family of sequences

Profile HMM

Multiple Alignments and Protein Family Classification

- Multiple alignment of a protein family shows variations in conservation along the length of a protein
- Example: after aligning many globin proteins, the biologists recognized that the helices region in globins are more conserved than others.



Finding Distant Members of a Protein Family

- A distant cousin of functionally related sequences in a protein family may have weak pairwise similarities with each member of the family and thus fail significance test
- However, they may have weak similarities with many members of the family

- The goal is to align a sequence to all members of the family at once.
- Family of related proteins can be represented by their multiple alignment and the corresponding profile.

Profile Representation of Protein Families

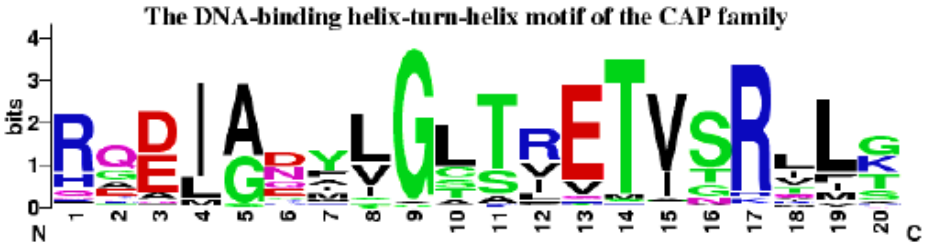
For example, aligned **DNA sequences** can be represented by a $4 \cdot n$ profile matrix reflecting the frequencies of nucleotides in every aligned position.

A	.72	.14	0	0	.72	.72	0	0
T	.14	.72	0	0	0	.14	.14	.86
G	.14	.14	.86	.44	0	.14	0	0
C	0	0	.14	.56	.28	0	.86	.14

Protein family can be represented by a $20 \cdot n$ profile representing frequencies of amino acids.

Multiple alignment and symbol probabilities

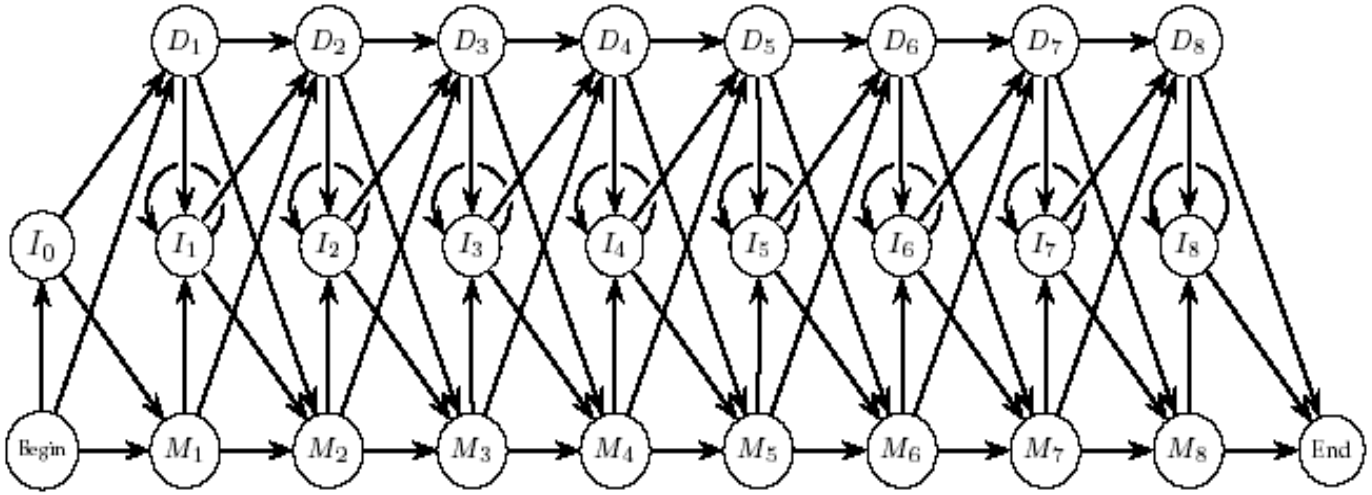
```
Helix                AAAAAAAAAAAAAAA   BBBBBBBBBBBBBBBCCCCCCCCCCC
HBA_HUMAN  ---------VLSPADKTNVKAAWGKVGA--HAGEYGAEALERMFLSFPTTKTYFPHF
HBB_HUMAN  --------VHLTPEEKSAVTALWGKV----NVDEVGGEALGRLLVVYPWTQRFFESF
MYG_PHYCA  ---------VLSEGEWQLVLHVWAKVEA--DVAGHGQDILIRLFKSHPETLEKFDRF
GLB3_CHITP ----------LSADQISTVQASFDKVKG------DPVGILYAVFKADPSIMAKFTQF
GLB5_PETMA PIVDTGSVAPLSAAEKTKIRSAWAPVYS--TYETSGVDILVKFFTSTPAAQEFFPKF
LGB2_LUPLU --------GALTESQAALVKSSWEEFNA--NIPKHTHRFFILVLEIAPAAKDLFS-P
GLB1_GLYDI ---------GLSAAQRQVIAATWKDIAGADNGAGVGKDCLIKFLSAHPQMAAVFG-F
Consensus            Ls....  v a W kv .  .    g . L.. f . P .   F  P

Helix          DDDDDDDEEEEEEEEEEEEEEEEEEE                FFFFFFFFFFF
HBA_HUMAN  -DLS-----HGSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL-
HBB_HUMAN  GDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHL---D--NLKGTFATLSELHCDKL-
MYG_PHYCA  KHLKTEAEMKASEDLKKHGVTVLTALGAILKK----K-GHHEAELKPLAQSHATKH-
GLB3_CHITP AG-KDLESIKGTAPFETHANRIVGFFSKIIGEL--P---NIEADVNTFVASHKPRG-
GLB5_PETMA KGLTTADQLKKSADVRWHAERIINAVNDAVASM--DDTEKMSMKLRDLSGKHAKSF-
LGB2_LUPLU LK-GTSEVPQNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG-
GLB1_GLYDI SG----AS---DPGVAALGAKVLAQIGVAVSHL--GDEGKMVAQMKAVGVRHKGYGN
Consensus   .  t    .. . v..Hg kv. a   a...l   d   . a l. l   H  .

Helix       FFGGGGGGGGGGGGGGGGGG      HHHHHHHHHHHHHHHHHHHHHHHHHH
HBA_HUMAN  -RVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR------
HBB_HUMAN  -HVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH------
MYG_PHYCA  -KIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNKALELFRKDIAAKYKELGYQG
GLB3_CHITP --VTHDQLNNFRAGFVSYMKAHT--DFA-GAEAAWGATLDTFFGMIFSKM-------
GLB5_PETMA -QVDPQYFKVLAAVIADTVAAG---------DAGFEKLMSMICILLRSAY-------
LGB2_LUPLU --VADAHFPVVKEAILKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMNDAA---
GLB1_GLYDI KHIKAQYFEPLGASLLSAMEHRIGGKMNAAAKDAWAAAYADISGALISGLQS-----
Consensus    v.   f  l . ..  ....    f   . aa. k. .     l sky
```



What are Profile HMMs?

- A Profile HMM is a probabilistic representation of a multiple alignment
- A given multiple alignment (of a protein family) is used to build a profile HMM
- This model then may be used to find and score less obvious potential matches of new protein sequences

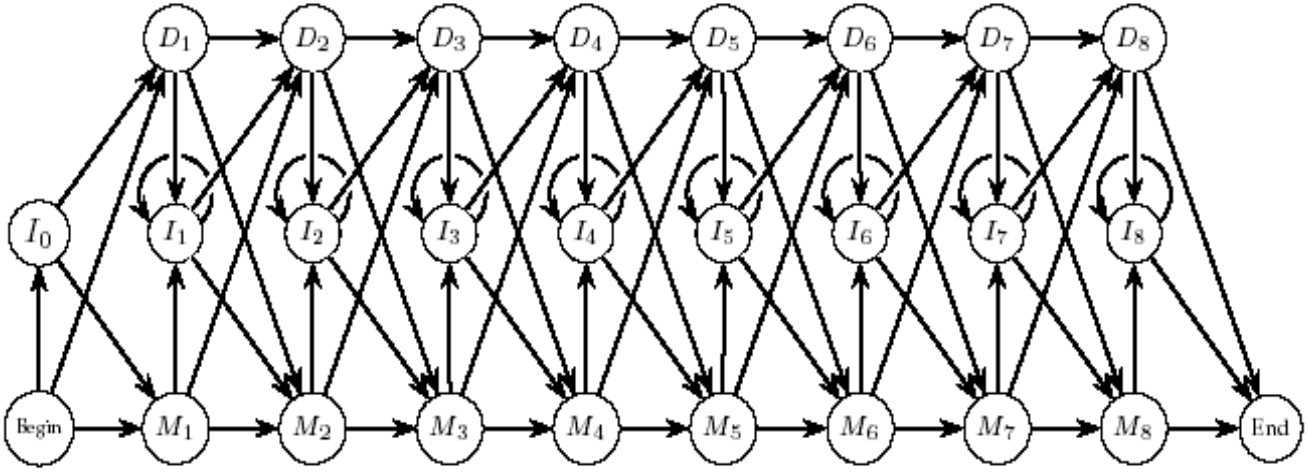
Building a profile HMM



- Assign each column (sequence position) to a *Match* state in HMM. Add I*nsertion* and *Deletion* state.
- Estimate the emission probabilities according to amino acid counts in column from the multiple alignment. Different positions in the protein will have different emission probabilities.
- Estimate the transition probabilities between *Match, Deletion* and *Insertion* states
- The HMM model gets **trained** to derive the optimal parameters

States of Profile HMM

- Match states $M_1 \ldots M_n$ (plus *begin/end* states)
- Insertion states $I_0 I_1 \ldots I_n$
- Deletion states $D_1 \ldots D_n$



Aligning new sequence to a profile

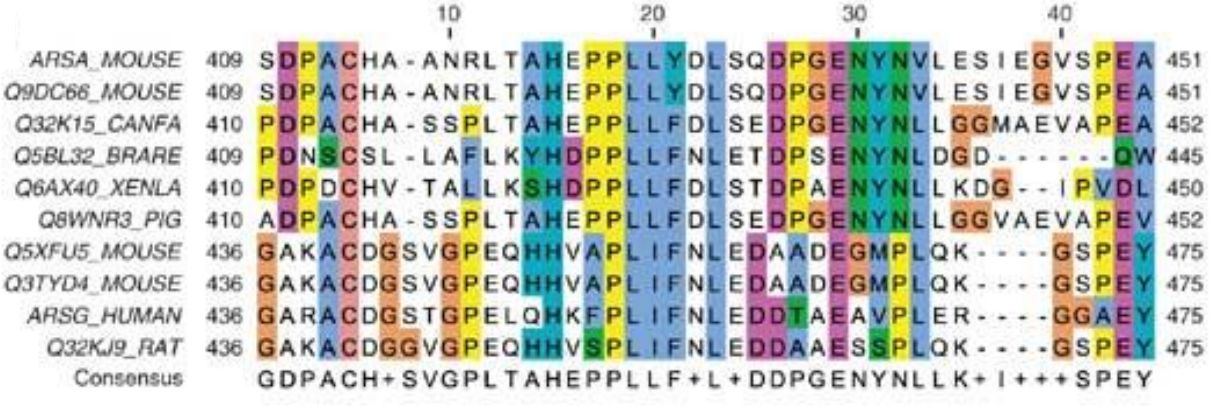
- HMMs can be used for aligning a sequence against a profile representing protein family
- A *20·n* profile *P* corresponds to *n* sequentially linked *match* states $M_1,\ldots,M_n$ in the profile HMM of *P*

Emission Probabilities in Profile HMM

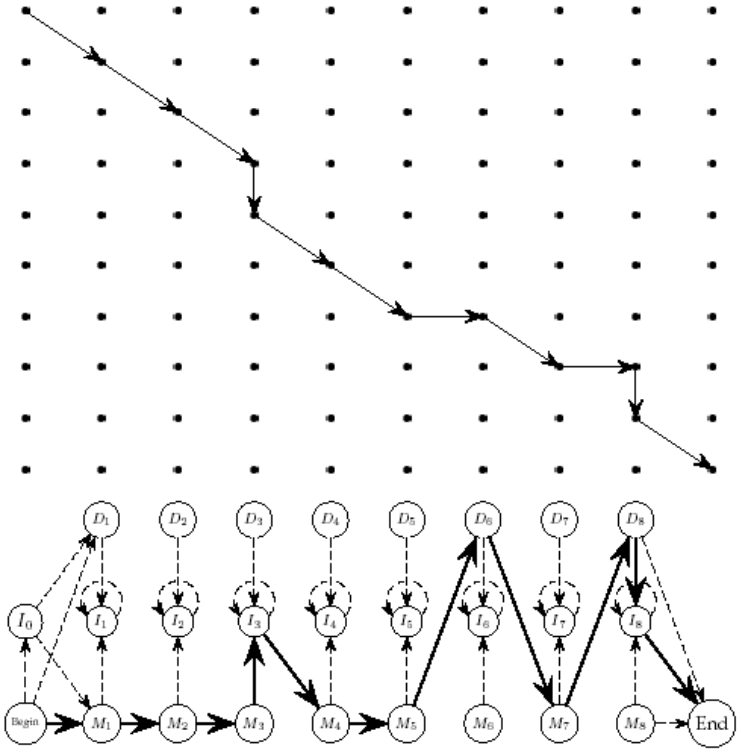
- Probability of emitting a symbol *a* at an insertion state I_j:

$$e_{Ij}(a) = p(a)$$

where *p(a)* is the frequency of the occurrence of the symbol *a* in all the sequences.



Paths in Edit Graph and Profile HMM



A path through an edit graph and the corresponding path through a profile HMM

Most used tool: *PFAM*

- Pfam decribes ***protein domains***
- Each protein domain family in Pfam has:

  - S*eed alignment*: manually verified multiple alignment of a representative set of sequences.

  - *HMM* built from the seed alignment for further database searches.

  - *Full alignment* generated automatically from the HMM
- The distinction between seed and full alignments facilitates Pfam updates.

  - Seed alignments are stable resources.

  - HMM profiles and full alignments can be updated with newly found amino acid sequences