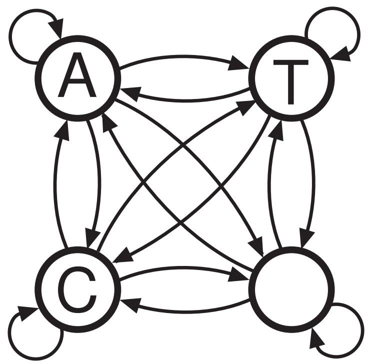
Markov Chains



Sequence: $x_1, x_2, \ldots, x_L$

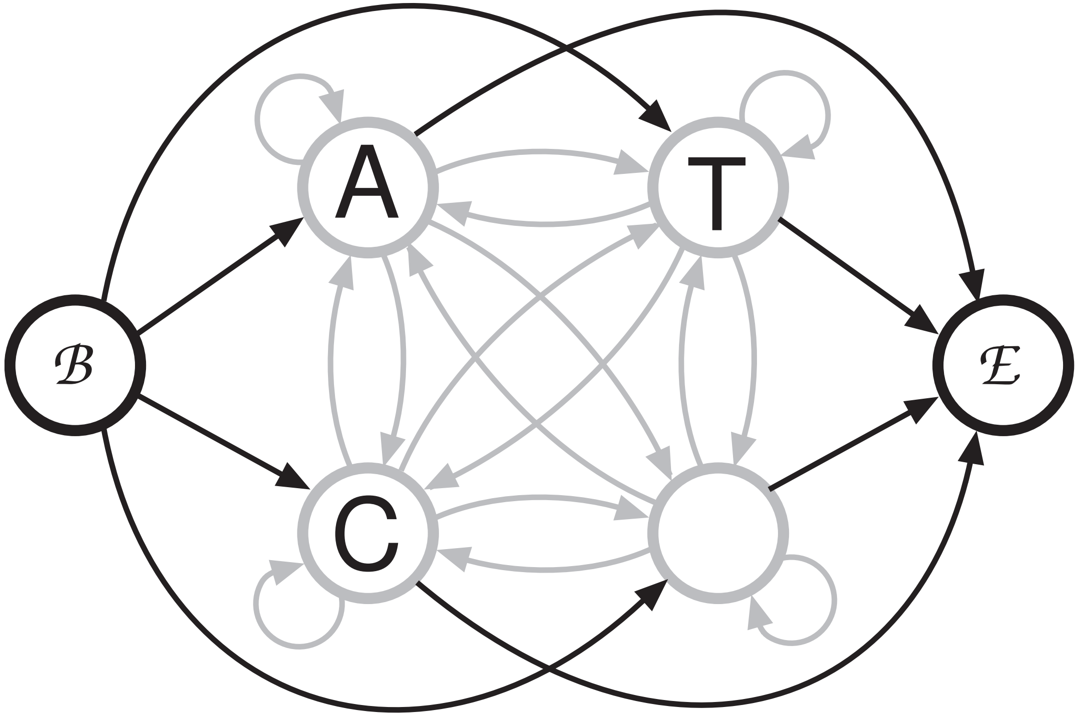
Transition probabilities: $a_{st} = P(x_i = t | x_{i-1} = s)$

Probability of sequence

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

Markov property: probability of x_i depends only on x_{i-1}

$$P(x) = P(x_L | x_{L-1}) P(x_{L-1} | x_{L-2}) \cdots P(x_1) = P(x_1) \prod_{i=2}^{L} a_{x_{i-1} x_i}.$$



hmm052

CpG Islands

Estimate of transition probabilities: $a^+_{st} = \dfrac{c^+_{st}}{\sum_{t'} c^+_{st'}}$,

+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

−	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

$$S(x) = \log \frac{P(x|\text{model } +)}{P(x|\text{model } -)} = \sum_{i=1}^{L} \log \frac{a^+_{x_{i-1}x_i}}{a^-_{x_{i-1}x_i}} = \sum_{i=1}^{L} \beta_{x_{i-1}x_i}$$

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

Hidden Markov Models

HMM Theory

Two stochastic processes:

A Markov chain over (hidden) states

Emissions of letters in each state

State process:

transition from state k to l with probability a_{kl}

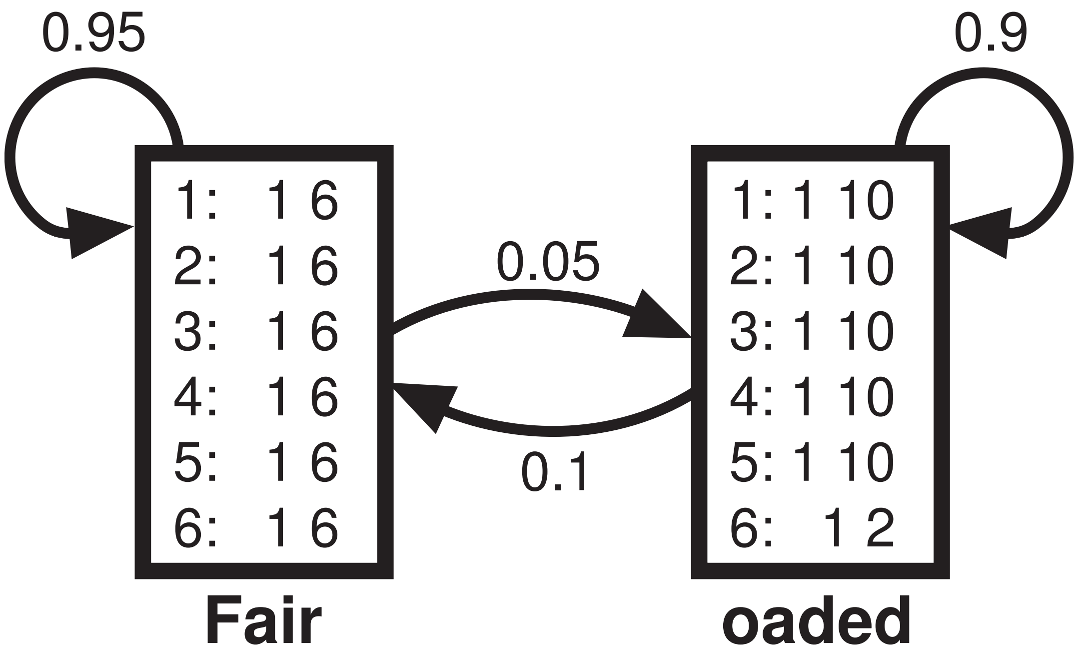
Emission process:

emit letter b in state k with probability $e_k(b)$

Probability of a path and a sequence:

$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

The occasionally dishonest casino, part 1



Viterbi algorithm

Most probable state path: $\pi^* = \underset{\pi}{\operatorname{argmax}}\, P(x, \pi)$

Calculate recursively: $v_l(i+1) = e_l(x_{i+1}) \max_k (v_k(i) a_{kl})$

Viterbi algorithm

Most probable state path: $\pi^* = \underset{\pi}{\text{argmax}}\, P(x, \pi)$

Calculate recursively: $v_l(i+1) = e_l(x_{i+1}) \max_k (v_k(i) a_{kl})$

Algorithm:

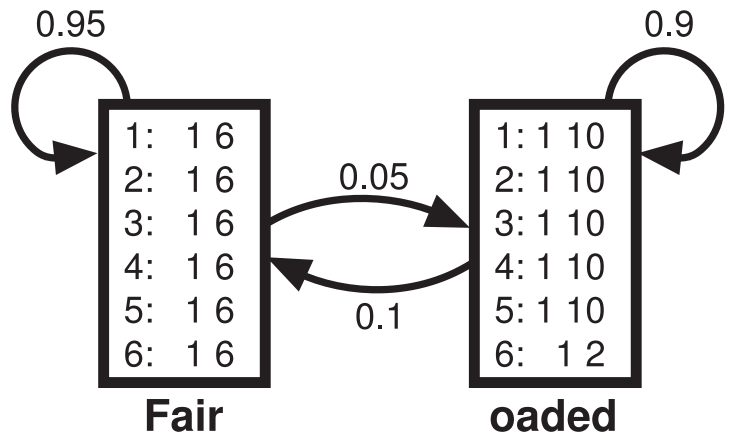
Initialisation ($i = 0$): $v_0(0) = 1$, $v_k(0) = 0$ for $k > 0$.

Recursion ($i = 1 \ldots L$): $v_l(i) = e_l(x_i) \max_k (v_k(i-1) a_{kl})$;
$\text{ptr}_i(l) = \text{argmax}_k (v_k(i-1) a_{kl})$.

Termination: $P(x, \pi^*) = \max_k (v_k(L) a_{k0})$;
$\pi^*_L = \text{argmax}_k (v_k(L) a_{k0})$.

Traceback ($i = L \ldots 1$): $\pi^*_{i-1} = \text{ptr}_i(\pi^*_i)$.

The occasionally dishonest casino, part 2




```
Rolls   315116246446644245311321631164152133625144543631656626566666
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLL

Rolls   651166453132651245636664631636663162326455236266666625151631
Die     LLLLLFFFFFFFFFFFFFLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFFFF
Viterbi LLLLLFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF

Rolls   222555441666566563564324364131513465146353411126414626253356
Die     FFFFFFFFLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls   366163666466232534413661661163252562462255265252266435353336
Die     LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   233121625364414432335163243633665562466662632666612355245242
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLFFFFFFFFFFFF
```

Forward algorithm

Calculate the total probability of x:

$$P(x) = \sum_{\pi} P(x, \pi)$$

Define $f_k(i) = P(x_1 \dots x_i, \pi_i = k)$ and do recursion like Viterbi algorithm.

Forward algorithm

Calculate the total probability of x:

$$P(x) = \sum_{\pi} P(x, \pi)$$

Define $f_k(i) = P(x_1 \ldots x_i, \pi_i = k)$ and do recursion like Viterbi algorithm.

Algorithm:

Initialisation ($i = 0$): $f_0(0) = 1$, $f_k(0) = 0$ for $k > 0$.

Recursion ($i = 1 \ldots L$): $f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}$.

Termination: $P(x) = \sum_k f_k(L) a_{k0}$.

Backward algorithm

One can do exactly the same **backwards** with:

$$b_k(i) = P(x_{i+1} \ldots x_L | \pi_i = k).$$

Backward algorithm

One can do exactly the same **backwards** with:

$$b_k(i) = P(x_{i+1} \ldots x_L | \pi_i = k).$$

Algorithm:

Initialisation ($i = L$): $b_k(L) = a_{k0}$ for all k.

Recursion ($i = L - 1, \ldots, 1$):

$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1).$$

Termination:

$$P(x) = \sum_l a_{0l} e_l(x_1) b_l(1).$$

Posterior probability of a state

Probabiliity that x_i is generated in state k:

$$P(\pi_i = k|x) = \frac{P(x, \pi_i = k)}{P(x)}$$

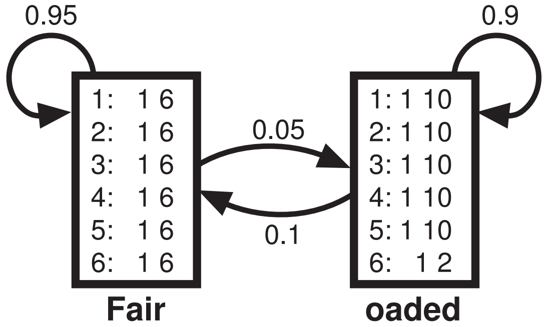
Calculate numerator:

$$\begin{aligned} P(x, \pi_i = k) &= P(x_1 \dots x_i, \pi_i = k)\ P(x_{i+1} \dots x_L | x_1 \dots x_i, \pi_i = k) \\ &= P(x_1 \dots x_i, \pi_i = k)\ P(x_{i+1} \dots x_L | \pi_i = k) \\ &= f_k(i) b_k(i) \end{aligned}$$

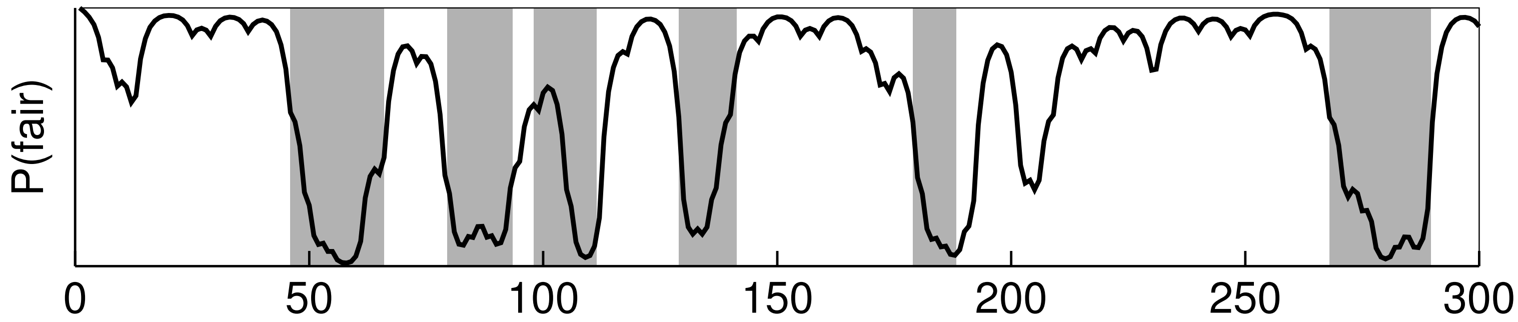
Result:

$$P(\pi_i = k|x) = \frac{f_k(i) b_k(i)}{P(x)}$$

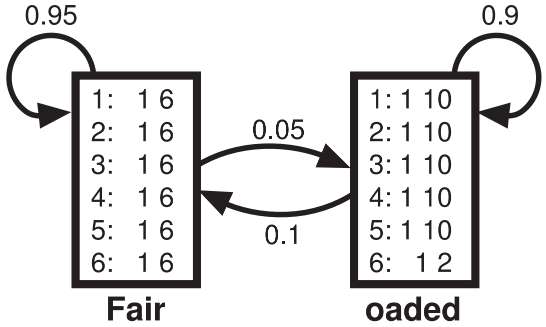
The occasionally dishonest casino, part 3–4



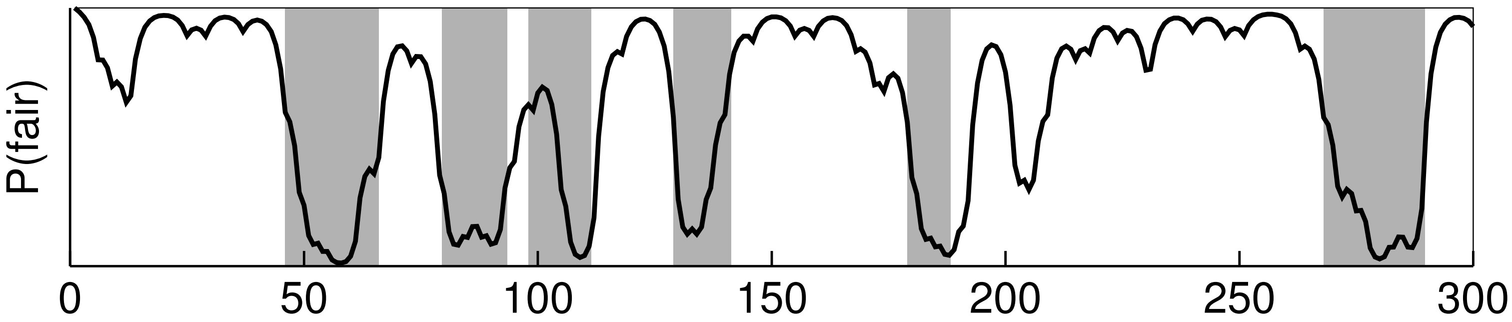
Posterior probability of the two states for 300 random rolls of a die:



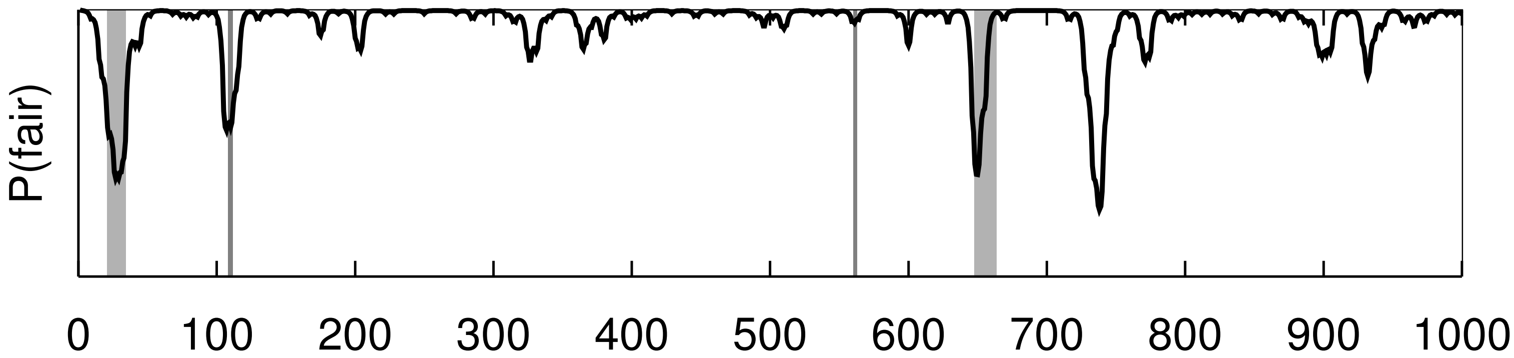
The occasionally dishonest casino, part 3–4



Posterior probability of the two states for 300 random rolls of a die:



Changing the probability of switching to the loaded die to 0.01:



Parameter estimation

Maximize the likelihood:

$$
\begin{aligned}
l(x^1, \ldots, x^n|\theta) &= \log P(x^1, \ldots, x^n|\theta) \\
&= \sum_{j=1}^{n} \log P(x^j|\theta)
\end{aligned}
$$

When state sequences are known:

$$
a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \quad \text{and} \quad e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}
$$

where A_{kl} and $E_k(b)$ are counts

```
Rolls   315116246446644245311321631164152133625144543631656626566666
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL

Rolls   651166453132651245636664631636663162326455236266666625151631
Die     LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFFFF

Rolls   222555441666566563564324364131513465146353411126414626253356
Die     FFFFFFFFLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLL

Rolls   366163666466232534413661661163252562462255265252266435353336
Die     LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   233121625364414432335163243633665562466662632666612355245242
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
```

When the paths are unknown

Estimate A_{kl} and $E_k(b)$ from current model:

$$A_{kl} = \sum_j \frac{1}{P(x^j)} \sum_i f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1)$$

$$E_k(b) = \sum_j \frac{1}{P(x^j)} \sum_{\{i|x_i^j=b\}} f_k^j(i) b_k^j(i),$$

Baum-Welch algorithm

Initialisation: Pick arbitrary model parameters.

Recurrence:

Set all A and E to zero.

For each sequence $j = 1 \ldots n$:

Calculate $f_k(i)$ by the forward algorithm.

Calculate $b_k(i)$ by the backward algorithm.

Add contribution to A and E.

Update parameters.

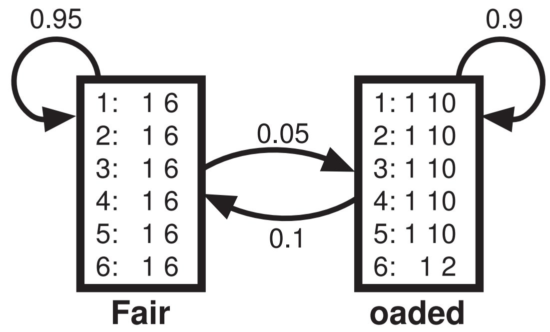
Calculate new log likelihood of model.

Termination:

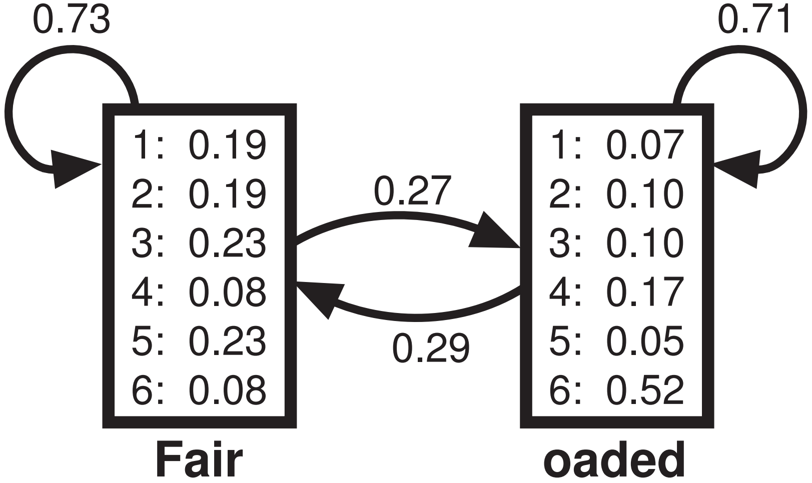
Stop if log likelihood change is small

or the maximum number of iterations is exceeded.

The occasionally dishonest casino, part 5



Estimation from
300 random rolls



Estimation from
30000 random rolls