Example: Whack-the-mole *)

A mole has burrowed a network of underground tunnels, with N openings at ground level. We are interested in modeling the sequence of openings at which the mole will poke its head out of the ground. The probability distribution of the "next" opening only depends on the present location of the mole.

Three holes:

$$X = \{x_1, x_2, x_3\}$$

modeling the movements of the mole probabilistically as a Markov chain

*) Thanks for the example to Emilio Frazolli

Whack-the-mole as Markov chain

Transition probabilities:

$$T = \begin{bmatrix} P(x_1|x_1) & P(x_2|x_1) & P(x_3|x_1) \\ P(x_1|x_2) & P(x_2|x_2) & P(x_3|x_2) \\ P(x_1|x_3) & P(x_2|x_3) & P(x_3|x_3) \end{bmatrix}$$

$$= \begin{bmatrix} T_{1,1} & T_{1,2} & T_{1,3} \\ T_{2,1} & T_{2,2} & T_{2,3} \\ T_{3,1} & T_{3,2} & T_{3,3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.4 & 0 & 0.6 \\ 0 & 0.6 & 0.4 \end{bmatrix}$$

Initial probabilities:

$$\pi = (\pi(1), \pi(2), \pi(3)) = \quad ?$$

Whack-the-mole as Markov chain

Let us assume that we know, e.g., with certainty, that the mole was at hole x_1 at time step 1 (i.e., $P(X_1 = x_1) = 1$. It takes d time units to go get the mallet. Where should I wait for the mole if I want to maximize the probability of whacking it the next time it surfaces?

$$\pi =$$

Whack-the-mole as Markov chain

Let us assume that we know, e.g., with certainty, that the mole was at hole x_1 at time step 1 (i.e., $P(X_1 = x_1) = 1$. It takes d time units to go get the mallet. Where should I wait for the mole if I want to maximize the probability of whacking it the next time it surfaces?

$$\pi = (1, 0, 0)$$

Whack-the-mole as Markov chain

to calculate:

$$p_d = (p_d(x_1), p_d(x_2), p_d(x_3))$$

in general holds:

$$p_d(s) = \sum_{\forall s', s=succ(s')} p_{d-1}(s') * T_{s,s'}$$

in particular we can compute the individual probabilities:

$$\begin{aligned} p_2(x_1) &= \sum_{s \in \{x_1, x_2, x_3\}} p_1(s) \cdot T_{s,x_1} \\ &= p_1(x_1) \cdot T_{1,1} + p_1(x_2) \cdot T_{2,1} + p_1(x_3) \cdot T_{3,1} \\ &= 1 \cdot 0.1 + 0 \cdot 0.4 + 0 \cdot 0 \\ &= 0.1 \end{aligned}$$

Example: Whack-the-mole

for the other cases:

$$
\begin{aligned}
p_2(x_2) &= \sum_{s \in \{x_1, x_2, x_3\}} p_1(s) \cdot T_{s,x_2} \\
&= 1 \cdot 0.4 + 0 \cdot 0 + 0 \cdot 0.6 \\
&= 0.4
\end{aligned}
$$

$$
\begin{aligned}
p_2(x_3) &= \sum_{s \in \{x_1, x_2, x_3\}} p_1(s) \cdot T_{s,x_3} \\
&= 1 \cdot 0.5 + 0 \cdot 0.6 + 0 \cdot 0.4 \\
&= 0.5
\end{aligned}
$$

which results in the probability distribution at timestep 2

$$p_2 = (0.1, 0.4, 0.5)$$

Whack-the-mole as Markov chain

the computation on a more abstract level

$$
\begin{aligned}
p_2 &= p_1 \cdot T \\
p_3 &= p_2 \cdot T \\
p_4 &= p_3 \cdot T \\
&\ldots
\end{aligned}
$$

the distribution at the next timesteps

$$
\begin{aligned}
p_3 &= p_2 \cdot T = (0.17, 0.34, 0.49) \\
p_4 &= p_3 \cdot T = (0.153, 0.362, 0.485) \\
&\ldots
\end{aligned}
$$

Whack-the-mole as Markov chain

Alternative query with little relevance for mole hunting, but high impact for practical applications (speech recognition, signal processing, machine translation, ...):

Assuming the mole surfaces every time it reaches a hole, so we can see it. What's the probability of a particular sequence of appearances $O = (x_1 x_2 ... x_n)$, i.e. $p(x_1 x_2 ... x_n | \mathcal{M})$?

$$
\begin{aligned}
p(x_1 x_2 ... x_n | \mathcal{M}) &= \pi(x_1) \cdot p(x_2 | x_1) \cdot \ldots \cdot p(x_{n-1} | p(x_n)) \\
&= \pi(x_1) \cdot \prod_{i=2}^{n} p(x_i | x_{n-1})
\end{aligned}
$$

which follows from the chain rule together with the Markov (independence) assumption

Whack-the-mole as Markov chain

How likely it is that we are able to observe the mole doing a (anticyclic) round trip $O = (3, 2, 1, 3)$?

$$
\begin{aligned}
p(x_3, x_2, x_1, x_3|\mathcal{M}) &= \pi(x_3) \cdot p(x_2|x_3) \cdot p(x_1|x_2) \cdot p(x_3|x_1) \\
&= 0.485 \cdot 0.6 \cdot 0.4 \cdot 0.5 \\
&= 0.05496
\end{aligned}
$$

using the approximation of the long term probability distribution $p_4 = (0.153, 0.362, 0.485)$ from above as initial probabilities

Whack-the-mole as a Hidden-Markov-Model

Let us assume that every time the mole surfaces, we can hear it, but not see it (its dark outside). Our hearing is not very precise.

uncertainty of sensing $\rightarrow$ separation of state and observation

Markov chain $\rightarrow$ Hidden Markov model

Whack-the-mole as a Hidden-Markov-Model

modeling the uncertainty of sensing probabilistically using additional emission/observation probabilities:

$$E = \begin{bmatrix} P(o_1|x_1) & P(o_2|x_1) & P(o_3|x_1) \\ P(o_1|x_2) & P(o_2|x_2) & P(o_3|x_2) \\ P(o_1|x_3) & P(o_2|x_1) & P(o_3|x_3) \end{bmatrix}$$

$$= \begin{bmatrix} E_{1,1} & E_{1,2} & E_{1,3} \\ E_{2,1} & E_{2,2} & E_{2,3} \\ E_{3,1} & E_{3,2} & E_{3,3} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

Whack-the-mole as a Hidden-Markov-Model

Let us assume that over three times the mole surfaces, we make the following sequence of observations: $O = (1, 3, 3)$

Compute the distribution of the states of the mole at the end of the observation, as well as its most likely state trajectory.

state distribution:

Whack-the-mole as a Hidden-Markov-Model

Let us assume that over three times the mole surfaces, we make the following sequence of observations: $O = (1, 3, 3)$

Compute the distribution of the states of the mole at the end of the observation, as well as its most likely state trajectory.

state distribution: forward algorithm

Whack-the-mole as a Hidden-Markov-Model

Let us assume that over three times the mole surfaces, we make the following sequence of observations: $O = (1, 3, 3)$

Compute the distribution of the states of the mole at the end of the observation, as well as its most likely state trajectory.

state distribution: forward algorithm

most likely state sequence:

Whack-the-mole as a Hidden-Markov-Model

Let us assume that over three times the mole surfaces, we make the following sequence of observations: $O = (1, 3, 3)$

Compute the distribution of the states of the mole at the end of the observation, as well as its most likely state trajectory.

state distribution: forward algorithm

most likely state sequence: $\textsc{Viterbi}$ algorithm

Whack-the-mole: Forward Algorithm

Forward algorithm:

$$p_d(s) = \alpha_d(s) = E_{s,o_d} \sum_{\forall s', s = succ(s')} p_{d-1}(s') * T_{s',s}$$

Initially:

$$p_1(s) = \alpha_1(s) = \pi(s) E_{s,o_d}$$

Whack-the-mole: Forward Algorithm

assuming an initial probability distribution, e.g. the long-term approximation of the Markov-chain (rounded values)

$$\pi = (0.16, 0.36, 0.48)$$

we can compute the individual probabilities:
for timestep one: $o_1 = 1$

$$\begin{aligned} p_1(x_1) &= \pi(s) \cdot E_{s,o_1} \\ &= \pi(x_1) \cdot E_{1,1} \\ &= 0.16 \cdot 0.6 \\ &= 0.096 \end{aligned}$$

Whack-the-mole: Forward Algorithm

for the other states at timestep one: $o_1 = 1$

$$
\begin{aligned}
p_1(x_2) &= \pi(s) \cdot E_{s,o_1} \\
&= \pi(x_2) \cdot E_{2,1} \\
&= 0.36 \cdot 0.2 \\
&= 0.072
\end{aligned}
$$

$$
\begin{aligned}
p_1(x_3) &= \pi(s) \cdot E_{s,o_1} \\
&= \pi(x_3) \cdot E_{3,1} \\
&= 0.48 \cdot 0.2 \\
&= 0.096
\end{aligned}
$$

Whack-the-mole: Forward Algorithm

for timestep two: $o_2 = 3$

$$
\begin{aligned}
p_2(x_1) &= E_{x_1,o_2} \sum_{s \in \{x_1,x_2,x_3\}} p_1(s) \cdot T_{s,x_1} \\
&= E_{1,3} \cdot (p_1(x_1) \cdot T_{1,1} + p_1(x_2) \cdot T_{2,1} + p_1(x_3) \cdot T_{3,1} \\
&= 0.2 \cdot (0.096 \cdot 0.1 + 0.072 \cdot 0.4 + 0.096 \cdot 0) \\
&= 0.00768
\end{aligned}
$$

$$
\begin{aligned}
p_2(x_2) &= E_{x_2,o_2} \sum_{s \in \{x_1,x_2,x_3\}} p_1(s) \cdot T_{s,x_2} \\
&= E_{2,3} \cdot (p_1(x_1) \cdot T_{1,2} + p_1(x_2) \cdot T_{2,2} + p_1(x_3) \cdot T_{3,2} \\
&= 0.2 \cdot (0.096 \cdot 0.4 + 0.072 \cdot 0 + 0.096 \cdot 0.6) \\
&= 0.0192
\end{aligned}
$$

Whack-the-mole: Forward Algorithm

for timestep two (cont.): $o_2 = 3$

$$\begin{aligned} p_2(x_3) &= E_{x_3,o_2} \sum_{s \in \{x_1,x_2,x_3\}} p_1(s) \cdot T_{s,x_3} \\ &= E_{3,3} \cdot (p_1(x_1) \cdot T_{1,3} + p_1(x_2) \cdot T_{2,3} + p_1(x_3) \cdot T_{3,3} \\ &= 0.6 \cdot (0.096 \cdot 0.5 + 0.072 \cdot 0.6 + 0.096 \cdot 0.4) \\ &= 0.07776 \end{aligned}$$

Whack-the-mole: Forward Algorithm

for timestep three: $o_3 = 3$

$$
\begin{aligned}
p_3(x_1) &= E_{x_1,o_3} \sum_{s \in \{x_1,x_2,x_3\}} p_2(s) \cdot T_{s,x_1} \\
&= E_{1,3} \cdot (p_2(x_1) \cdot T_{1,1} + p_2(x_2) \cdot T_{2,1} + p_2(x_3) \cdot T_{3,1} \\
&= 0.2 \cdot (0.00768 \cdot 0.1 + 0.00192 \cdot 0.4 + 0.07776 \cdot 0) \\
&= 0.0003072
\end{aligned}
$$

$$
\begin{aligned}
p_3(x_2) &= E_{x_2,o_3} \sum_{s \in \{x_1,x_2,x_3\}} p_2(s) \cdot T_{s,x_2} \\
&= E_{2,3} \cdot (p_2(x_1) \cdot T_{1,2} + p_2(x_2) \cdot T_{2,2} + p_2(x_3) \cdot T_{3,2} \\
&= 0.2 \cdot (0.00768 \cdot 0.4 + 0.00192 \cdot 0 + 0.007776 \cdot 0.6) \\
&= 0.00154752
\end{aligned}
$$

Whack-the-mole: Forward Algorithm

for timestep three (cont.): $o_3 = 3$

$$
\begin{aligned}
p_3(x_3) &= E_{x_3,o_3} \sum_{s \in \{x_1,x_2,x_3\}} p_2(s) \cdot T_{s,x_3} \\
&= E_{3,3} \cdot (p_2(x_1) \cdot T_{1,3} + p_2(x_2) \cdot T_{2,3} + p_2(x_3) \cdot T_{3,3} \\
&= 0.6 \cdot (0.00768 \cdot 0.5 + 0.0192 \cdot 0.6 + 0.07776 \cdot 0.4) \\
&= 0.0278784
\end{aligned}
$$

Whack-the-mole: Forward Algorithm

Evolution of the probability distribution
given the observation sequence $O = (1, 3, 3)$

$$\pi(s) = (0.16, 0.36, 0.48)$$

$$p_1(s) = (0.096, 0.072, 0.096)$$

$$p_2(s) = (0.00768, 0.0192, 0.07776)$$

$$p_3(s) = (0.0003072, 0.00154752, 0.0278784)$$

Whack-the-mole: Forward Algorithm

computation on a more abstract level representing observations as sequences of one hot vectors

$$O = (o_1, o_2, ..., o_n) = (\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix})$$

Whack-the-mole: Forward Algorithm

$$p_1 = (E \cdot o_1)^T \circ \pi \cdot T$$
$$p_2 = (E \cdot o_2)^T \circ p_1 \cdot T$$
$$p_3 = (E \cdot o_3)^T \circ p_2 \cdot T$$
$$\dots$$
$$p_n = \underbrace{(E \cdot o_n)^T}_{} \circ \underbrace{p_{n-1} \cdot T}_{}$$

- $(E \cdot o_n)^T$: probability distribution of a state having produced the observation $p(o_i|x_j)$
- $\circ$: HADAMARD product
- $p_{n-1} \cdot T$: probability distribution for the current state $p_{n-1}(x_i) \cdot p(x_j|x_i))$

Whack-the-mole: Forward Algorithm

HADAMARD or entrywise product (here for vectors)

$$A \circ B = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix} = \begin{pmatrix} a_1 \cdot b_1 \\ a_2 \cdot b_2 \\ \dots \\ a_n \cdot b_n \end{pmatrix}$$

Whack-the-mole: VITERBI Algorithm

Determining the most likely state sequence
VITERBI coefficients: maximum probability to reach a state from the start point given the observation sequence up to that point in time
coefficients have to be computed for each state at each time point
observation: $O = (1, 2, 3, 1)$

Whack-the-mole: VITERBI Algorithm

intialization:

$$\delta_1(s) = \pi_s \; E_{s,o_1}$$

$$pred_1(s) = null$$

1	
2	
3	

Whack-the-mole: VITERBI Algorithm

intialization:

$$\delta_1(s) = \pi_s \; E_{s,o_1}$$

$$pred_1(s) = null$$

1	0.0912
2	
3	

$$\begin{aligned} \delta_1(1) &= \pi_1 \cdot E_{1,1} \\ &= 0.152 \cdot 0.6 \\ &= 0.0912 \\ pred_1(1) &= null \end{aligned}$$

Whack-the-mole: VITERBI Algorithm

intialization:

$$\delta_1(s) = \pi_s \; E_{s,o_1}$$

$$pred_1(s) = null$$

1	0.0912
2	0.0724
3	

$$\begin{aligned} \delta_1(2) &= \pi_2 \cdot E_{2,1} \\ &= 0.362 \cdot 0.2 \\ &= 0.0724 \\ pred_1(2) &= null \end{aligned}$$

Whack-the-mole: VITERBI Algorithm

intialization:

$$\delta_1(s) = \pi_s \; E_{s,o_1}$$

$$pred_1(s) = null$$

1	0.0912
2	0.0724
3	0.097

$$\begin{aligned}
\delta_1(3) &= \pi_3 \cdot E_{3,1} \\
&= 0.485 \cdot 0.2 \\
&= 0.097 \\
pred_1(3) &= null
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

recursive computation:

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\ T_{q,s})$$

1	0.0912	
2	0.0724	
3	0.097	

Whack-the-mole: VITERBI Algorithm

recursive computation:

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\, T_{q,s})$$

1	0.0912	0.005712 / 2
2	0.0724	
3	0.097	

$$\begin{aligned}
\delta_2(1) &= E_{1,2} \cdot \max_q(\delta_1(q) \cdot T_{q,1}) \\
&= 0.2 \cdot \max\{0.0912 \cdot 0.1, 0.0724 \cdot 0.4, 0.097 \cdot 0\} \\
&= 0.2 \cdot \max\{0.00912, 0.02896, 0\} \\
&= 0.005712 \\
pred_2(1) &= 2
\end{aligned}$$

Whack-the-mole: Viterbi Algorithm

recursive computation:

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\ T_{q,s})$$

1	0.0912	0.005712 / 2
2	0.0724	0.003492 / 3
3	0.097	

$$\begin{aligned}
\delta_2(2) &= E_{2,2} \cdot \max_q(\delta_1(q) \cdot T_{q,2}) \\
&= 0.6 \cdot \max\{0.0912 \cdot 0.4, 0.0724 \cdot 0, 0.097 \cdot 0.6\} \\
&= 0.6 \cdot \max\{0.03648, 0, 0.0582\} \\
&= 0.03492 \\
pred_2(2) &= 3
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

recursive computation:

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\ T_{q,s})$$

1	0.0912	0.005712 / 2
2	0.0724	0.003492 / 3
3	0.097	0.00912 / 1

$$\begin{aligned}
\delta_2(3) &= E_{3,2} \cdot \max_q(\delta_1(q) \cdot T_{q,3}) \\
&= 0.2 \cdot \max\{0.0912 \cdot 0.5, 0.0724 \cdot 0.6, 0.097 \cdot 0.4\} \\
&= 0.2 \cdot \max\{0.0456, 0.04344, 0.0388\} \\
&= 0.00912 \\
pred_2(3) &= 1
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

recursive computation (2):

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\; T_{q,s})$$

1	0.0912	0.005712 / 2	
2	0.0714	0.003492 / 3	
3	0.097	0.00912 / 1	

Whack-the-mole: Viterbi Algorithm

recursive computation (2):

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\ T_{q,s})$$

1	0.0912	0.005712 / 2	0.0027936 / 2
2	0.0714	0.003492 / 3	
3	0.097	0.00912 / 1	

$$\begin{aligned}
\delta_3(1) &= E_{1,3} \cdot \max_q(\delta_1(q) \cdot T_{q,1}) \\
&= 0.2 \cdot \max\{0.005712 \cdot 0.1, 0.03492 \cdot 0.4, 0.0912 \cdot 0\} \\
&= 0.2 \cdot \max\{0.0005712, 0.013968, 0\} \\
&= 0.0027936 \\
pred_3(1) &= 2
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

recursive computation (2):

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\, T_{q,s})$$

1	0.0912	0.005712 / 2	0.0027936 / 2
2	0.0714	0.003492 / 3	0.010944 / 3
3	0.097	0.00912 / 1	

$$\begin{aligned}
\delta_3(2) &= E_{2,3} \cdot \max_q(\delta_1(q) \cdot T_{q,2}) \\
&= 0.2 \cdot \max\{0.005712 \cdot 0.4, 0.03492 \cdot 0, 0.0912 \cdot 0.6\} \\
&= 0.2 \cdot \max\{0.0022848, 0, 0.05472\} \\
&= 0.010944 \\
pred_3(2) &= 3
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

recursive computation (2):

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\ T_{q,s})$$

1	0.0912	0.005712 / 2	0.0027936 / 2
2	0.0714	0.003492 / 3	0.010944 / 3
3	0.097	0.00912 / 1	0.021888 / 3

$$\begin{aligned}
\delta_3(3) &= E_{3,3} \cdot \max_q(\delta_1(q) \cdot T_{q,3}) \\
&= 0.6 \cdot \max\{0.005712 \cdot 0.5, 0.03492 \cdot 0.6, 0.0912 \cdot 0.4 \\
&= 0.6 \cdot \max\{0.002856, 0.020952, 0.03648\} \\
&= 0.021888 \\
pred_3(3) &= 3
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

recursive computation (3):

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\, T_{q,s})$$

1	0.0912	0.005712 / 2	0.0027936 / 2	
2	0.0714	0.003492 / 3	0.010944 / 3	
3	0.097	0.00912 / 1	0.021888 / 3	

Whack-the-mole: VITERBI Algorithm

recursive computation (3):

$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$

$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\, T_{q,s})$

1	0.0912	0.005712 / 2	0.0027936 / 2	0.00262656 / 2
2	0.0714	0.003492 / 3	0.010944 / 3	
3	0.097	0.00912 / 1	0.021888 / 3	

$$
\begin{aligned}
\delta_4(1) &= E_{1,1} \cdot \max_q(\delta_1(q) \cdot T_{q,1}) \\
&= 0.6 \cdot \max\{0.0027936 \cdot 0.1, 0.010944 \cdot 0.4, \\
&\qquad 0.021888 \cdot 0\} \\
&= 0.6 \cdot \max\{0.0002793, 0.0043776, 0\} \\
&= 0.00262656 \\
pred_4(1) &= 2
\end{aligned}
$$

Whack-the-mole: VITERBI Algorithm

recursive computation (3):

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\, T_{q,s})$$

1	0.0912	0.005712 / 2	0.0027936 / 2	0.00262656 / 2
2	0.0714	0.003492 / 3	0.010944 / 3	0.00262656 / 3
3	0.097	0.00912 / 1	0.021888 / 3	

$$\begin{aligned}
\delta_4(2) &= E_{2,1} \cdot \max_q(\delta_1(q) \cdot T_{q,2}) \\
&= 0.2 \cdot \max\{0.0027936 \cdot 0.4, 0.010944 \cdot 0, \\
&\qquad 0.021888 \cdot 0.6\} \\
&= 0.2 \cdot \max\{0.00111744, 0, 0.0131328\} \\
&= 0.00262656 \\
pred_4(2) &= 3
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

recursive computation (3):

$$\delta_{k+1}(s) = E_{s,o_{k+1}} \cdot \max_q(\delta_k(q) \cdot T_{q,s})$$
$$pred_{k+1}(s) = \arg\max_q(\delta_k(q)\ T_{q,s})$$

1	0.0912	0.005712 / 2	0.0027936 / 2	0.00262656 / 2
2	0.0714	0.003492 / 3	0.010944 / 3	0.00262656 / 3
3	0.097	0.00912 / 1	0.021888 / 3	0.00175104 / {2,3}

$$\begin{aligned}
\delta_4(3) &= E_{3,1} \cdot \max_q(\delta_1(q) \cdot T_{q,3}) \\
&= 0.2 \cdot \max\{0.0027936 \cdot 0.5, 0.010944 \cdot 0.6, \\
&\qquad 0.021888 \cdot 0.4\} \\
&= 0.2 \cdot \max\{0.0013968, 0.0087552, 0.0087552\} \\
&= 0.00175104 \\
pred_4(3) &= \{2, 3\}
\end{aligned}$$

Whack-the-mole: VITERBI Algorithm

$$\delta_{k+1}(s) = \max_q \left(\delta_k(q)\, T_{q,s}\right) E_{s,o_{k+1}}$$

repeat recursively

- $\delta_{k+1}(s) = \max_q(\delta_k(q)\, T_{q,s})\, E_{s,o_{k+1}}$
- $pred_{k+1}(s) = \arg\max_q(\delta_k(q)\, T_{q,s})$

select the most likely terminal state $\hat{s}_t = \arg\max_s \delta_t(s)$
with $\hat{p} = \delta(\hat{s}_t)$ being the probability of the most likely path
reconstruct the most likely path backwards:
$\hat{q}_k = pred_{k+1}(\hat{q}_{k+1})$

Whack-the-mole: VITERBI Algorithm

- observation:

  $$(x_1, x_2, x_3, x_1)$$

- two optimal final states:

  $$\{s_1, s_2\}$$

- two optimal state sequences:

  $$\{(s_1, s_3, s_2, s_1), (s_1, s_3, s_3, s_2)\}$$