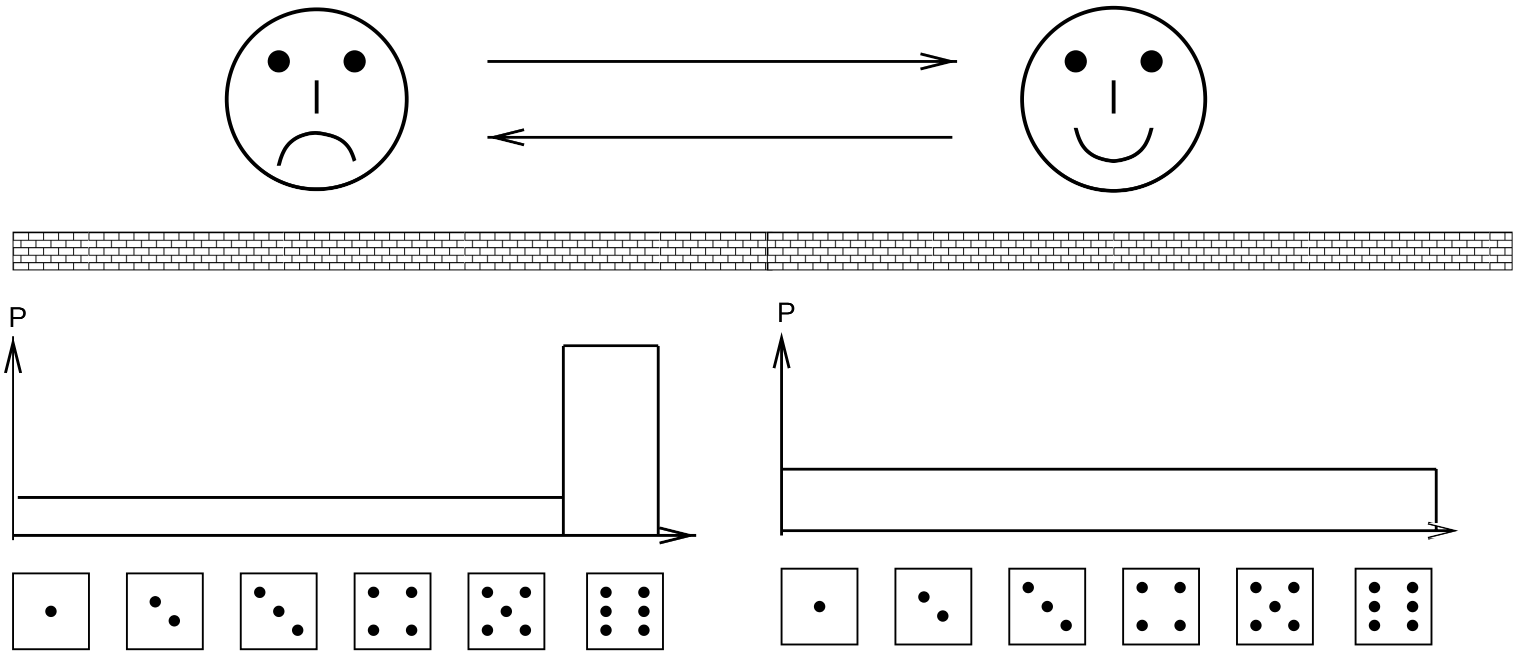
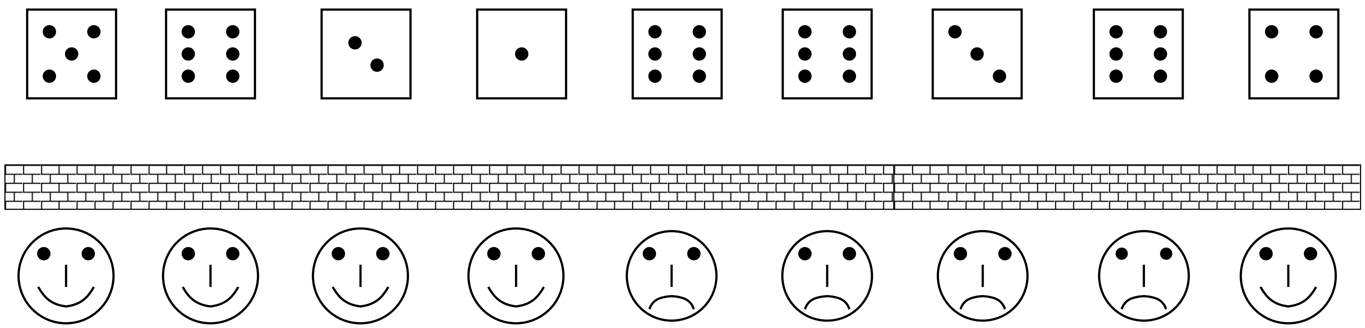
A Tutorial on Hidden Markov Models

Dirk Husmeier
Biomathematics and Statistics Scotland
at the Scottish Crop Research Institute
Invergowrie, Dundee DD2 5DA, UK
Email: dirk@bioss.ac.uk
http://www.bioss.ac.uk/∼dirk

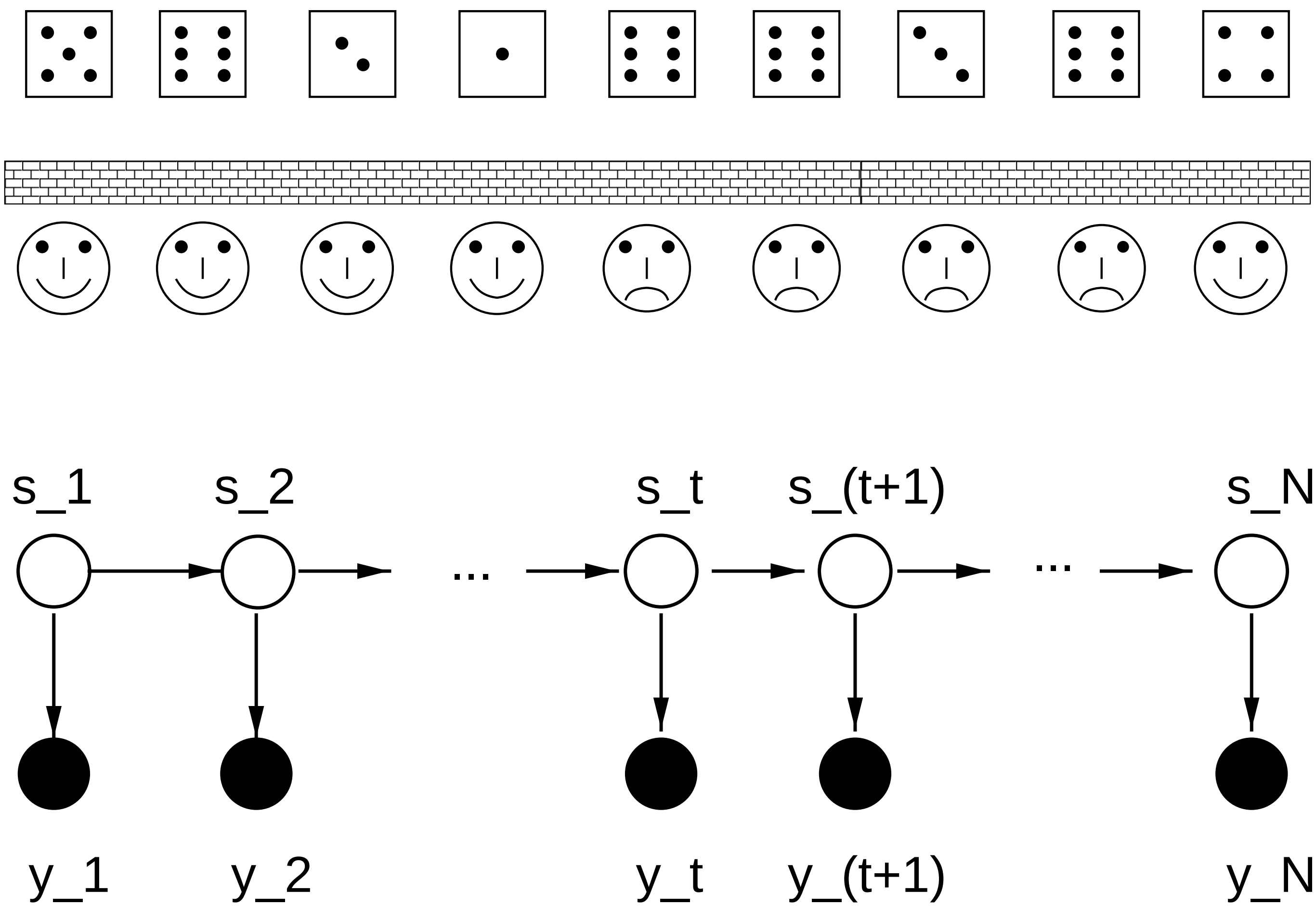
Example: The Occasionally Corrupt Casino



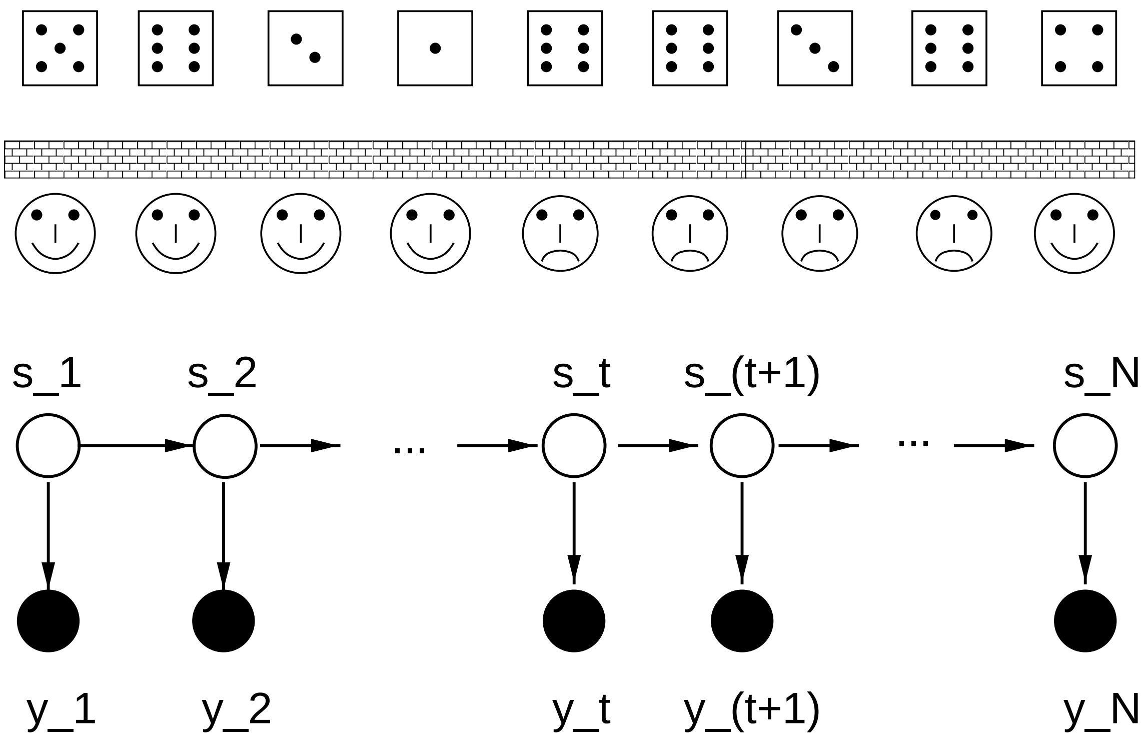
Example: HMM



Example: HMM

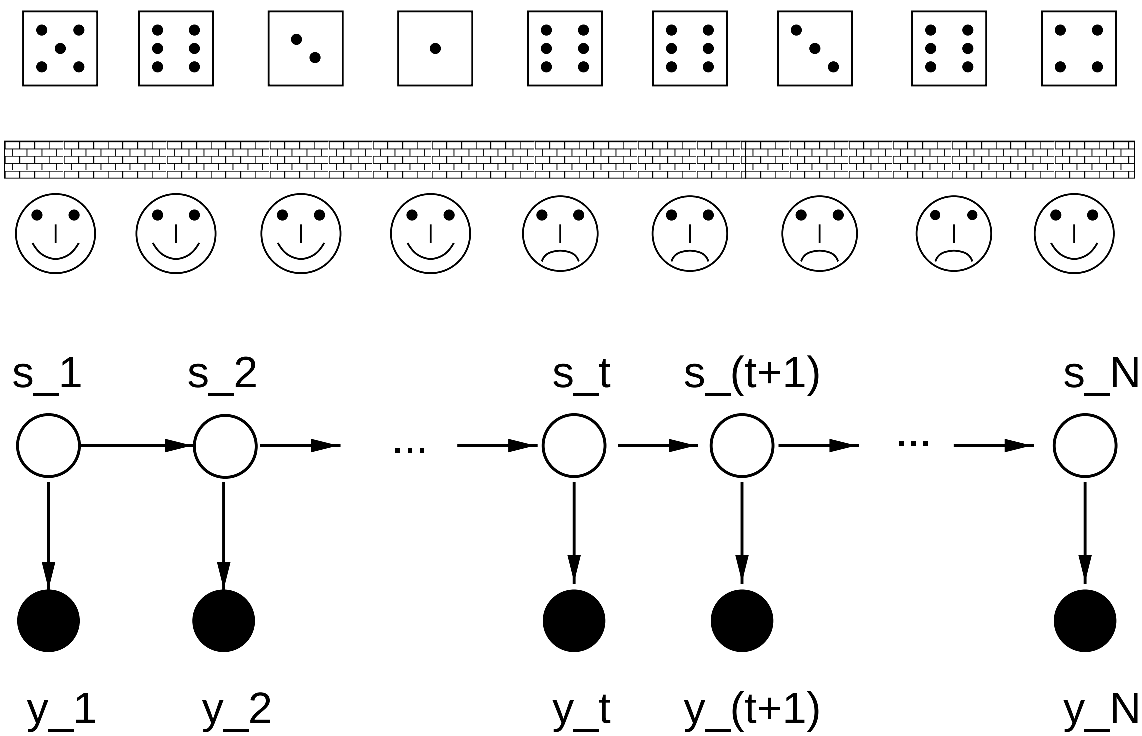


The Most Likely State Sequence



Find the mode of $P(S_1, \ldots, S_N | y_1, \ldots, y_N)$

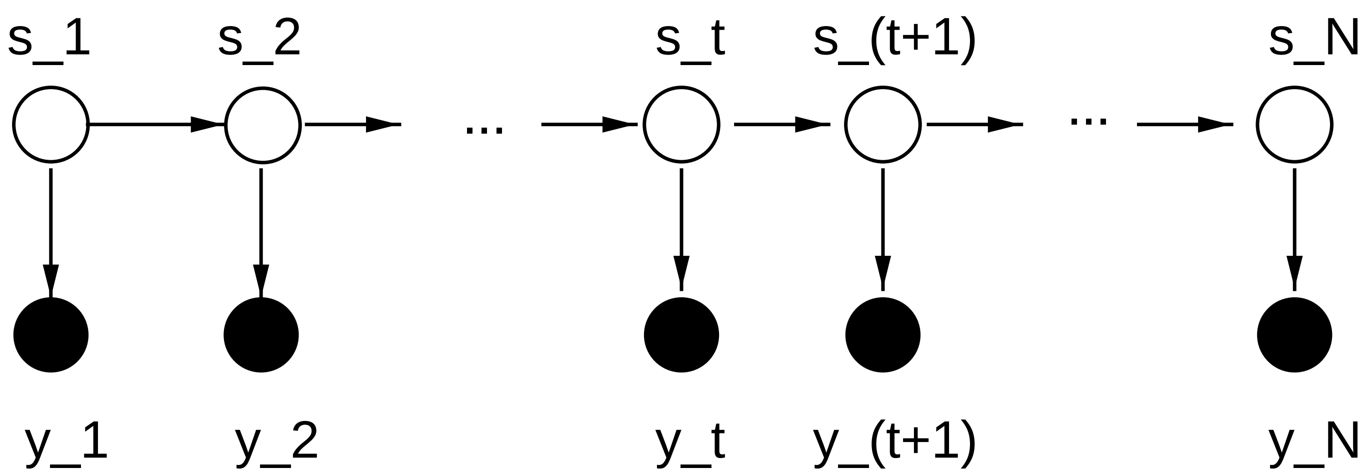
The Most Likely State Sequence



Find the mode of $P(S_1, \ldots, S_N | y_1, \ldots, y_N)$

Problem: $(S_1, \ldots, S_N) : 2^N$ different sequences.

Factorisation in HMMs



$$
\begin{aligned}
P(y_1, \ldots, y_N, S_1, \ldots, S_N) &= \prod_{t=1}^{N} P(y_t|S_t) \prod_{t=2}^{N} P(S_t|S_{t-1}) P(S_1) \\
&:= \prod_{t=1}^{N} P(y_t|S_t) \prod_{t=1}^{N} P(S_t|S_{t-1})
\end{aligned}
$$

The Most Likely State Sequence

$$\begin{aligned}\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) &= \max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N,y_1,\ldots,y_N) \\ &= \max_{S_N} \gamma_N(S_N)\end{aligned}$$

The Most Likely State Sequence

$$\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) = \max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N,y_1,\ldots,y_N)$$
$$= \max_{S_N} \gamma_N(S_N)$$

$$\gamma_n(S_n) = \max_{S_1,\ldots,S_{n-1}} P(y_1,\ldots,y_n,S_1,\ldots,S_n)$$

The Most Likely State Sequence

$$\begin{aligned}\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) &= \max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N,y_1,\ldots,y_N)\\ &= \max_{S_N} \gamma_N(S_N)\end{aligned}$$

$$\begin{aligned}\gamma_n(S_n) &= \max_{S_1,\ldots,S_{n-1}} P(y_1,\ldots,y_n,S_1,\ldots,S_n)\\ &= \max_{S_1,\ldots,S_{n-1}} \prod_{t=1}^{n} P(y_t|S_t)P(S_t|S_{t-1})\end{aligned}$$

The Most Likely State Sequence

$$\begin{aligned}\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) &= \max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N,y_1,\ldots,y_N) \\ &= \max_{S_N} \gamma_N(S_N)\end{aligned}$$

$$\begin{aligned}\gamma_n(S_n) &= \max_{S_1,\ldots,S_{n-1}} P(y_1,\ldots,y_n,S_1,\ldots,S_n) \\ &= \max_{S_1,\ldots,S_{n-1}} \prod_{t=1}^{n} P(y_t|S_t)P(S_t|S_{t-1}) \\ &= \max_{S_1,\ldots,S_{n-1}} P(y_n|S_n)P(S_n|S_{n-1}) \prod_{t=1}^{n-1} P(y_t|S_t)P(S_t|S_{t-1})\end{aligned}$$

The Most Likely State Sequence

$$\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) = \max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N,y_1,\ldots,y_N) = \max_{S_N} \gamma_N(S_N)$$

$$\gamma_n(S_n) = \max_{S_1,\ldots,S_{n-1}} P(y_1,\ldots,y_n,S_1,\ldots,S_n)$$

$$= \max_{S_1,\ldots,S_{n-1}} \prod_{t=1}^{n} P(y_t|S_t)P(S_t|S_{t-1})$$

$$= \max_{S_1,\ldots,S_{n-1}} P(y_n|S_n)P(S_n|S_{n-1}) \prod_{t=1}^{n-1} P(y_t|S_t)P(S_t|S_{t-1})$$

$$= P(y_n|S_n) \max_{S_{n-1}} P(S_n|S_{n-1}) \max_{S_1,\ldots,S_{n-2}} \prod_{t=1}^{n-1} P(y_t|S_t)P(S_t|S_{t-1})$$

The Most Likely State Sequence

$$\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) = \max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N,y_1,\ldots,y_N)$$
$$= \max_{S_N} \gamma_N(S_N)$$

$$\begin{aligned}
\gamma_n(S_n) &= \max_{S_1,\ldots,S_{n-1}} P(y_1,\ldots,y_n,S_1,\ldots,S_n) \\
&= \max_{S_1,\ldots,S_{n-1}} \prod_{t=1}^{n} P(y_t|S_t)P(S_t|S_{t-1}) \\
&= \max_{S_1,\ldots,S_{n-1}} P(y_n|S_n)P(S_n|S_{n-1}) \prod_{t=1}^{n-1} P(y_t|S_t)P(S_t|S_{t-1}) \\
&= P(y_n|S_n) \max_{S_{n-1}} P(S_n|S_{n-1}) \max_{S_1,\ldots,S_{n-2}} \prod_{t=1}^{n-1} P(y_t|S_t)P(S_t|S_{t-1}) \\
&= P(y_n|S_n) \max_{S_{n-1}} P(S_n|S_{n-1})\gamma_{n-1}(S_{n-1})
\end{aligned}$$

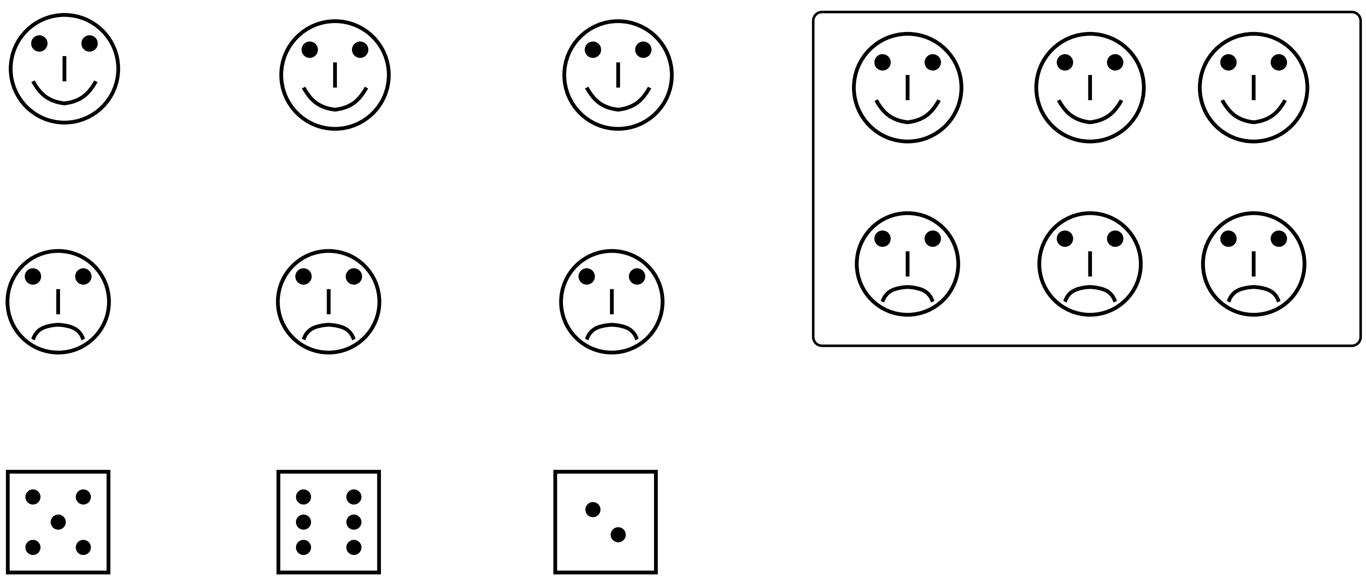
The Most Likely State Sequence

$$\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) = \max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N,y_1,\ldots,y_N)$$
$$= \max_{S_N} \gamma_N(S_N)$$

$$\begin{aligned}
\gamma_n(S_n) &= \max_{S_1,\ldots,S_{n-1}} P(y_1,\ldots,y_n,S_1,\ldots,S_n) \\
&= \max_{S_1,\ldots,S_{n-1}} \prod_{t=1}^{n} P(y_t|S_t)P(S_t|S_{t-1}) \\
&= \max_{S_1,\ldots,S_{n-1}} P(y_n|S_n)P(S_n|S_{n-1}) \prod_{t=1}^{n-1} P(y_t|S_t)P(S_t|S_{t-1}) \\
&= P(y_n|S_n) \max_{S_{n-1}} P(S_n|S_{n-1}) \max_{S_1,\ldots,S_{n-2}} \prod_{t=1}^{n-1} P(y_t|S_t)P(S_t|S_{t-1}) \\
&= P(y_n|S_n) \max_{S_{n-1}} P(S_n|S_{n-1}) \gamma_{n-1}(S_{n-1})
\end{aligned}$$

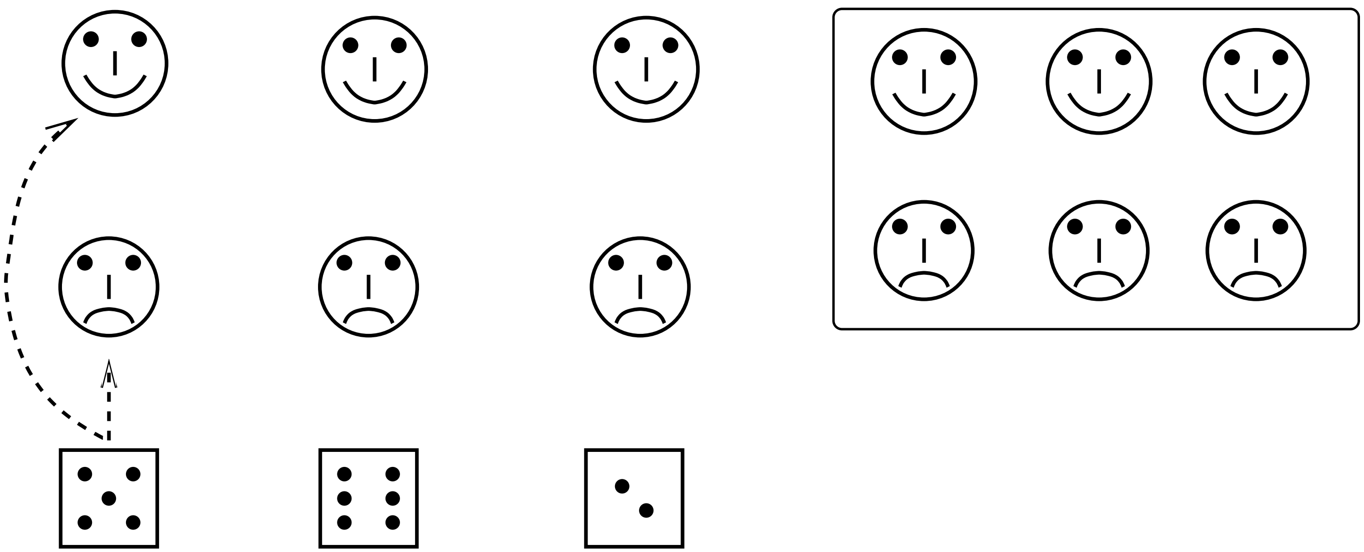
Computational complexity $K^N \rightarrow NK^2$ (K= number of states).

Viterbi Algorithm: Demo



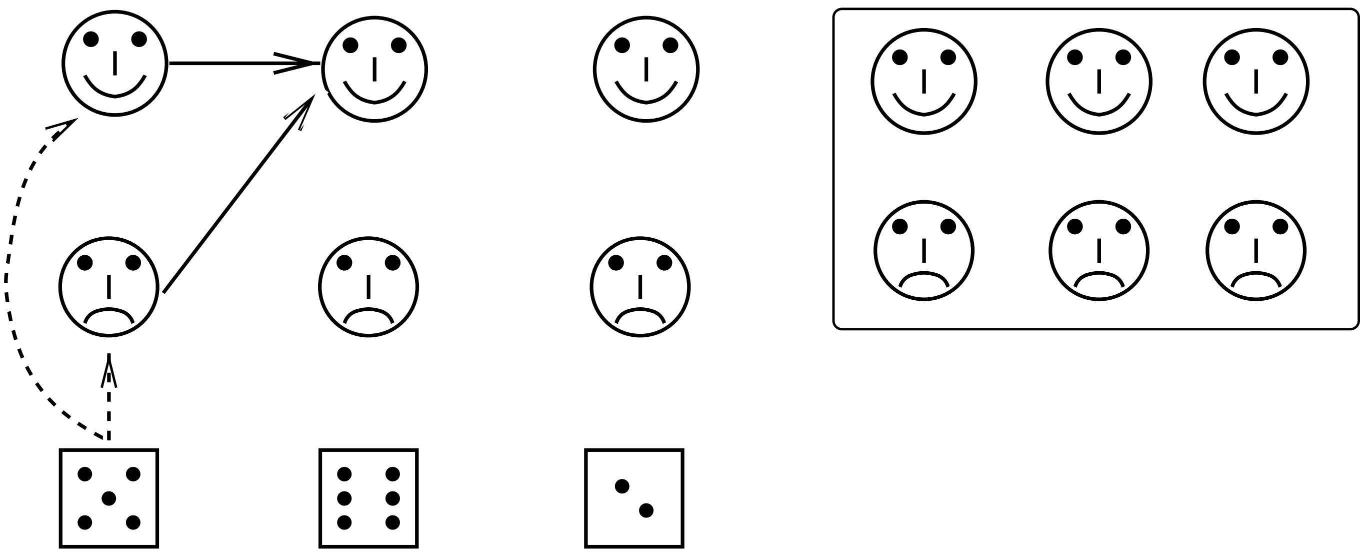
Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1)P(S_1)$
Recursion ($n = 2, \ldots, N$)	$\gamma_n(S_n) = P(y_n \mid S_n)\max_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback ($n = N-1, \ldots, 1$)	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

Viterbi Algorithm: Demo



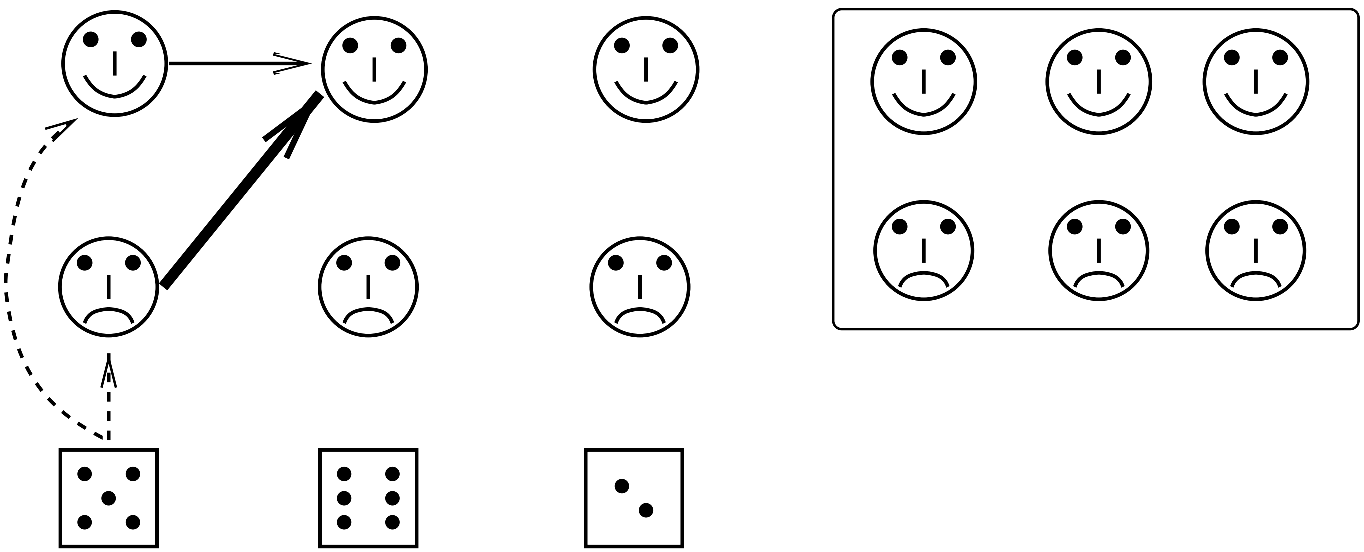
Initialisation	$\gamma_1(S_1) = P(y_1 \vert S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \vert S_n) \max_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$ $\mathrm{Pointer}(S_{n-1}) = \mathrm{argmax}_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \mathrm{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \mathrm{Pointer}(\hat{S}_{n+1})$

Viterbi Algorithm: Demo



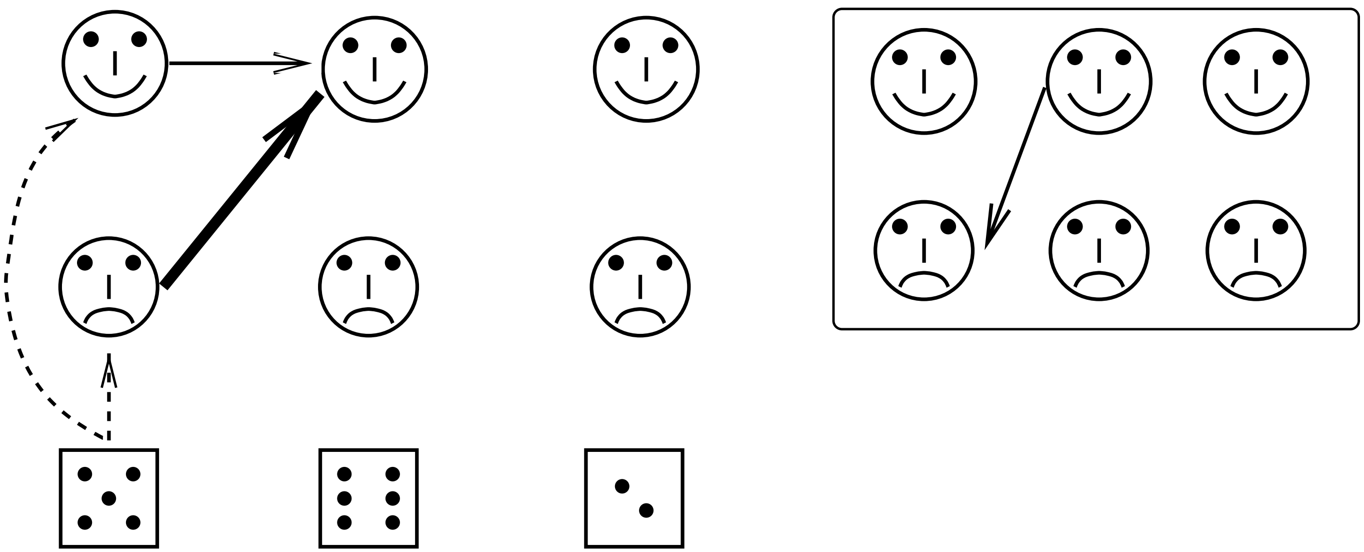
Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1) P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \mid S_n) \max_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

Viterbi Algorithm: Demo



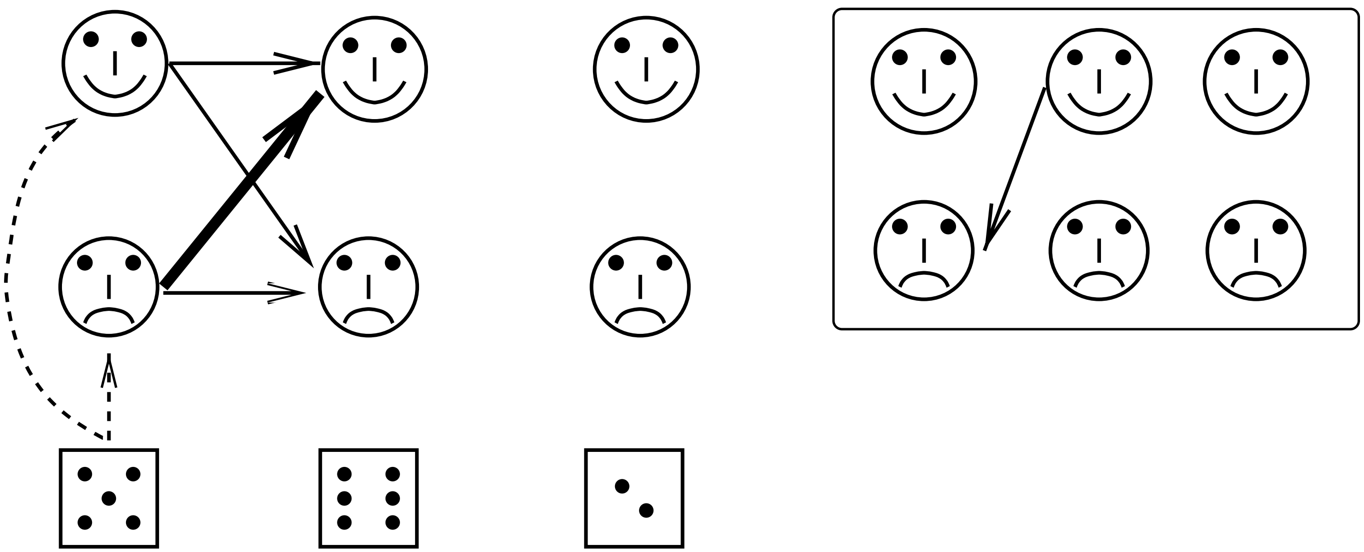
Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1) P(S_1)$
Recursion $(n = 2, \dots, N)$	$\gamma_n(S_n) = P(y_n \mid S_n) \max_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \dots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

Viterbi Algorithm: Demo



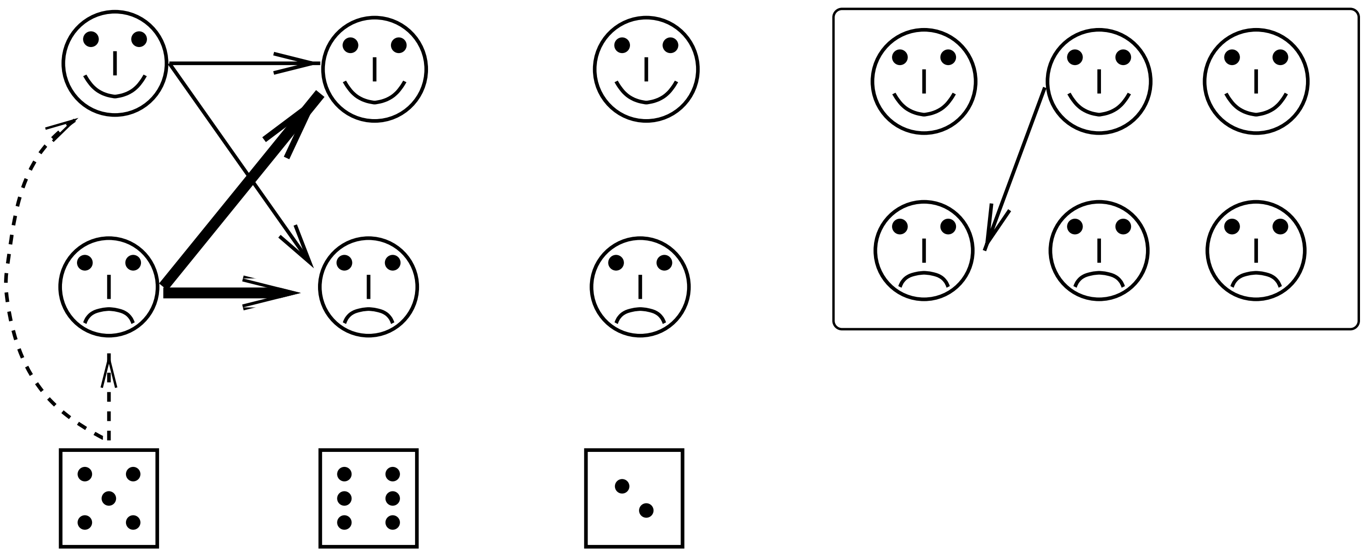
Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \mid S_n) \max_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

Viterbi Algorithm: Demo



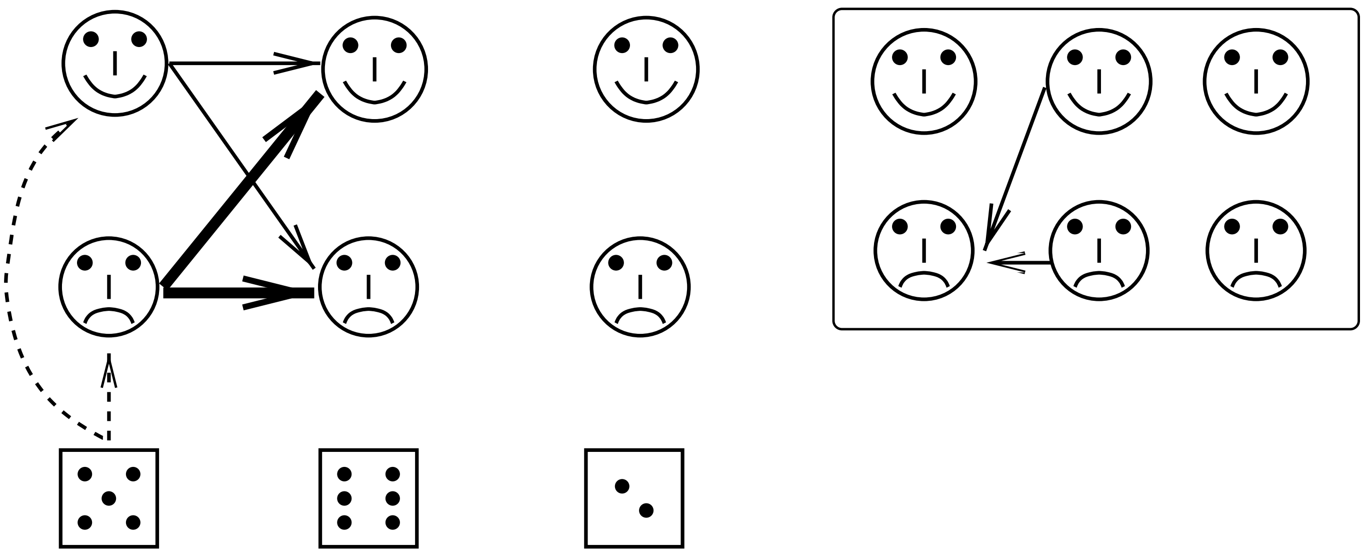
Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \mid S_n) \max_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

Viterbi Algorithm: Demo



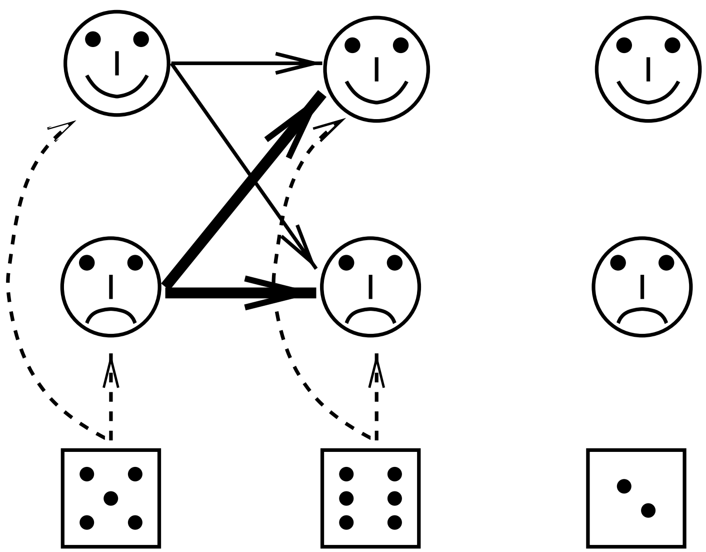
Initialisation	$\gamma_1(S_1) = P(y_1 \vert S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \vert S_n) \max_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

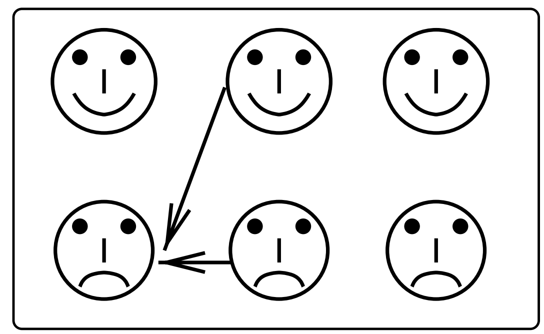
Viterbi Algorithm: Demo



Initialisation	$\gamma_1(S_1) = P(y_1\|S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n\|S_n) \max_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

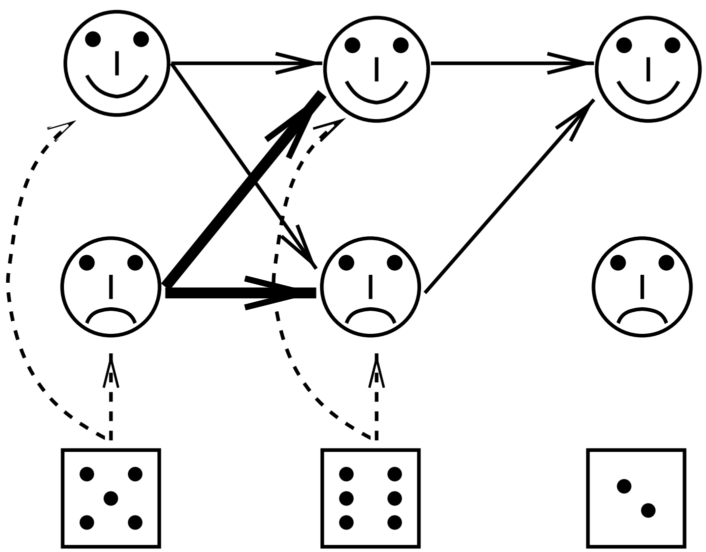
Viterbi Algorithm: Demo

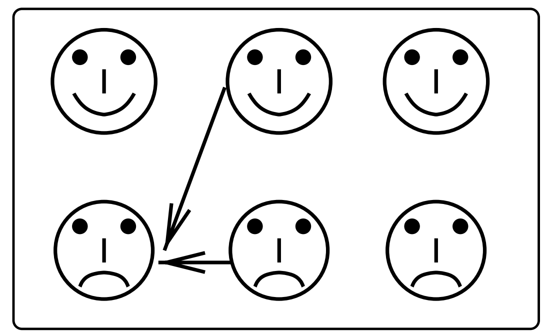




Initialisation	$\gamma_1(S_1) = P(y_1\|S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n\|S_n) \max_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

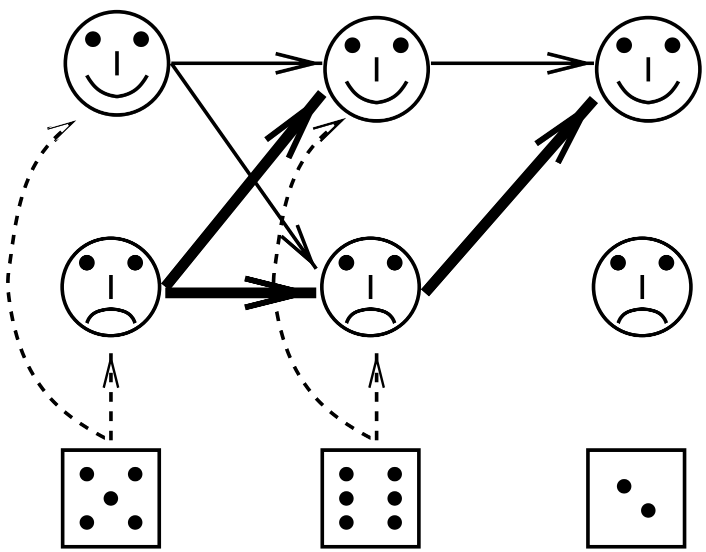
Viterbi Algorithm: Demo

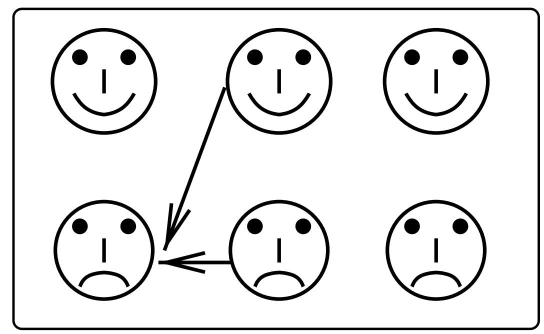




Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1) P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \mid S_n) \max_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

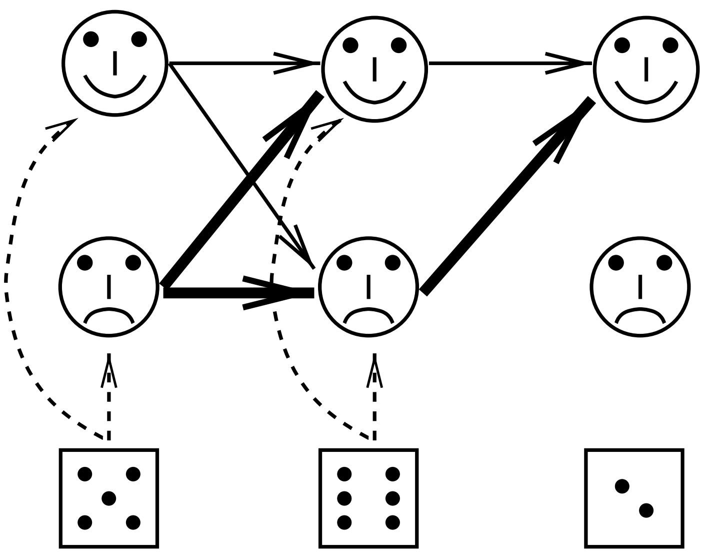
Viterbi Algorithm: Demo

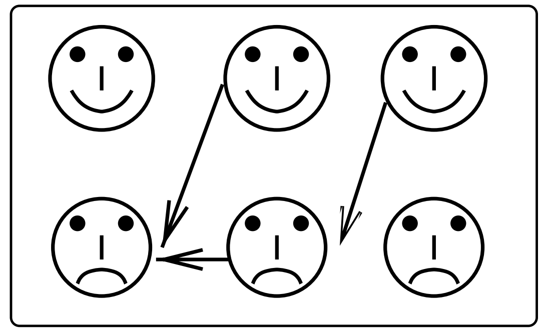




Initialisation	$\gamma_1(S_1) = P(y_1\|S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n\|S_n) \max_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

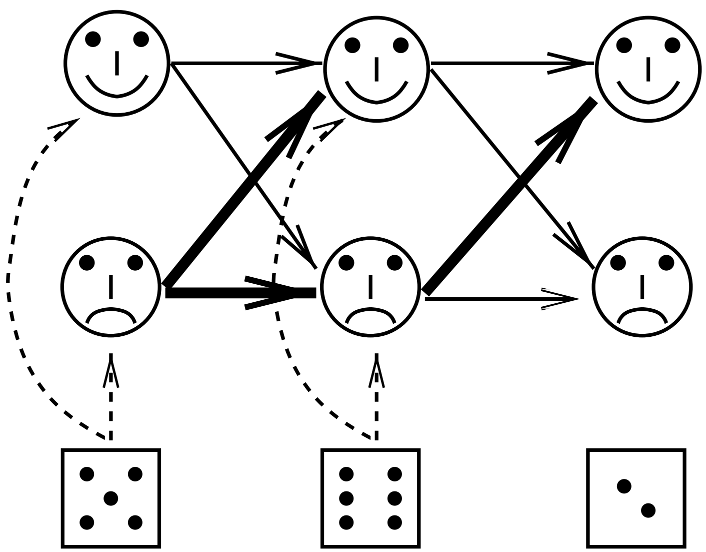
Viterbi Algorithm: Demo

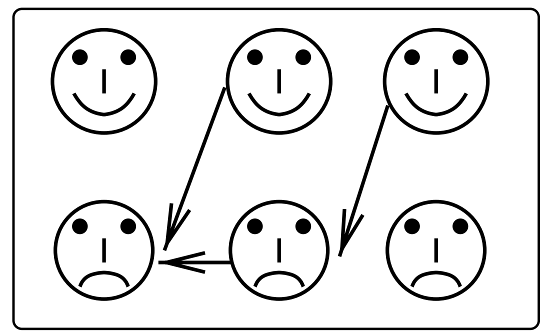




Initialisation	$\gamma_1(S_1) = P(y_1 \vert S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \vert S_n) \max_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$ $\mathrm{Pointer}(S_{n-1}) = \mathrm{argmax}_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \mathrm{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \mathrm{Pointer}(\hat{S}_{n+1})$

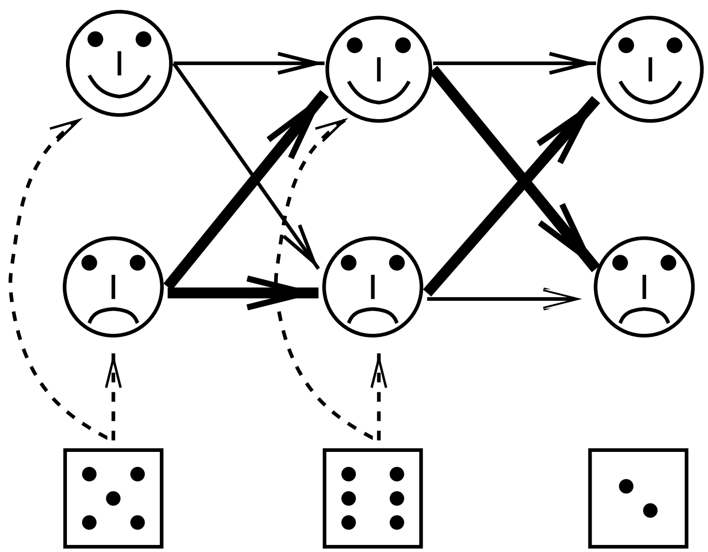
Viterbi Algorithm: Demo

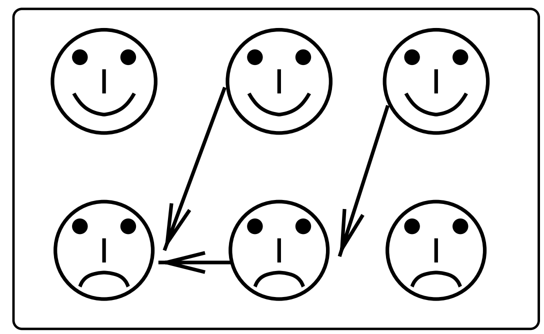




Initialisation	$\gamma_1(S_1) = P(y_1\|S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n\|S_n) \max_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n\|S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

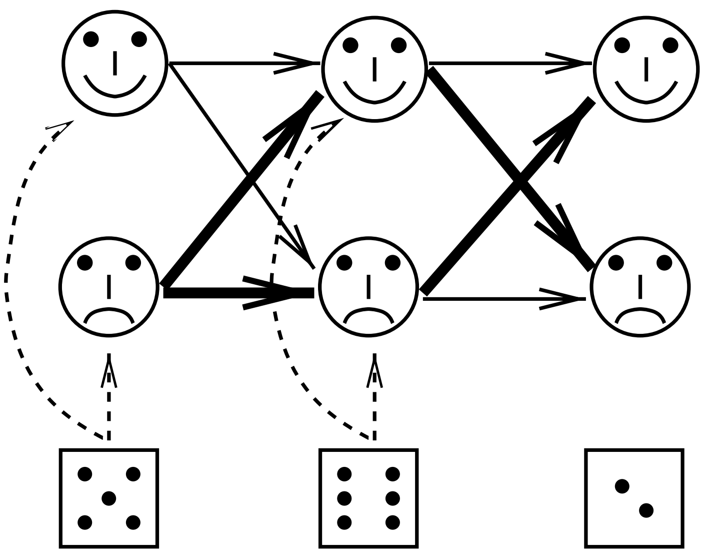
Viterbi Algorithm: Demo

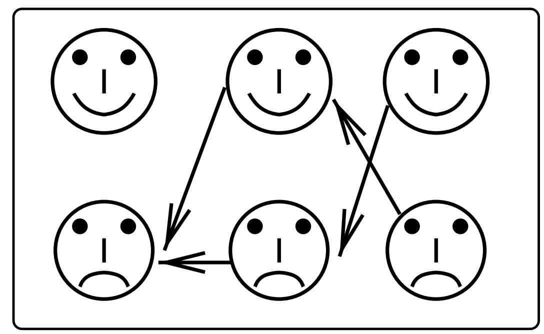




Initialisation	$\gamma_1(S_1) = P(y_1 \vert S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \vert S_n) \max_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$ $\mathrm{Pointer}(S_{n-1}) = \mathrm{argmax}_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \mathrm{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \mathrm{Pointer}(\hat{S}_{n+1})$

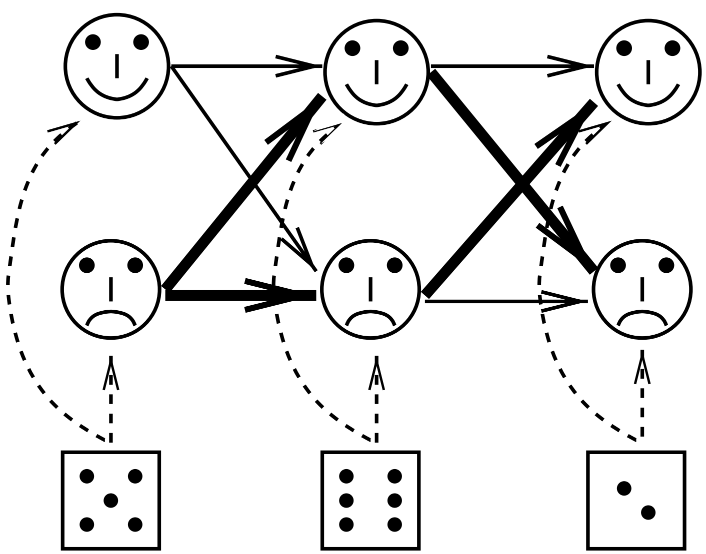
Viterbi Algorithm: Demo

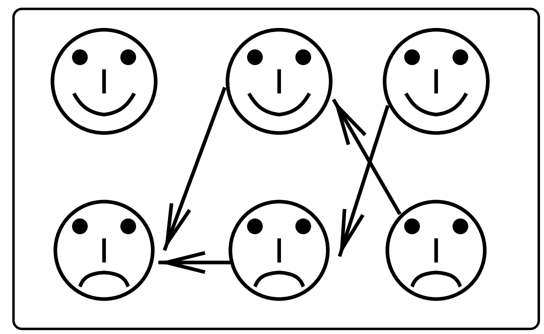




Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1) P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \mid S_n) \max_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1}) \gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

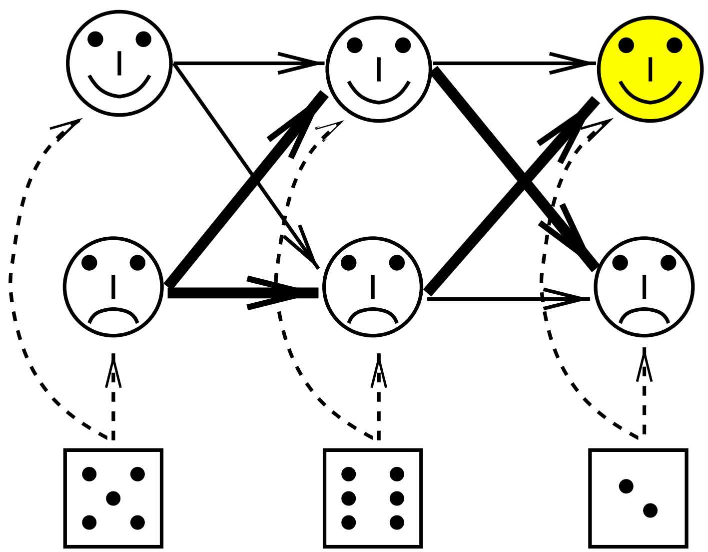
Viterbi Algorithm: Demo

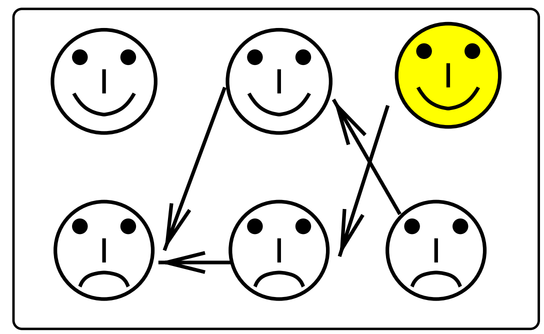




Initialisation	$\gamma_1(S_1) = P(y_1 \vert S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \vert S_n) \max_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

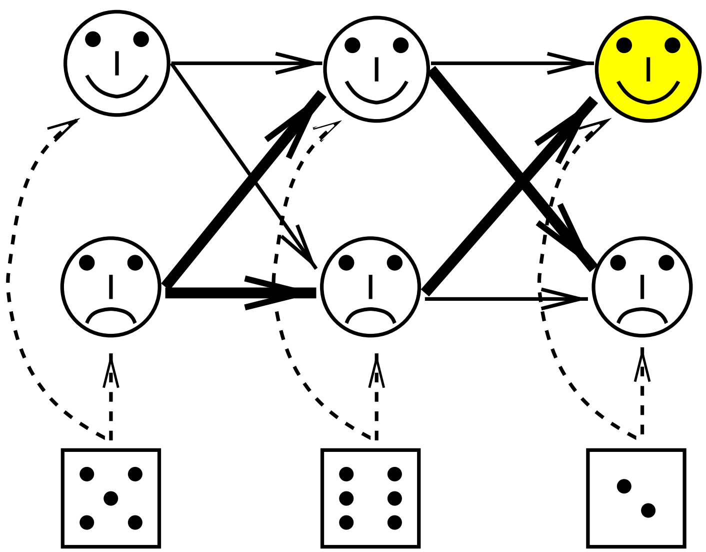
Viterbi Algorithm: Demo

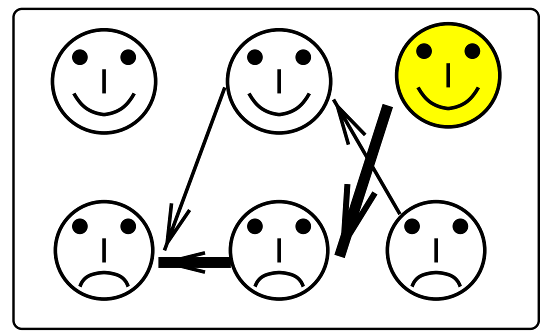




Initialisation	$\gamma_1(S_1) = P(y_1 \vert S_1)P(S_1)$
Recursion ($n = 2, \ldots, N$)	$\gamma_n(S_n) = P(y_n \vert S_n) \max_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$ $\mathrm{Pointer}(S_{n-1}) = \mathrm{argmax}_{S_{n-1}} P(S_n \vert S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \mathrm{argmax}_{S_N} \gamma_N(S_N)$
Traceback ($n = N-1, \ldots, 1$)	$\hat{S}_n = \mathrm{Pointer}(\hat{S}_{n+1})$

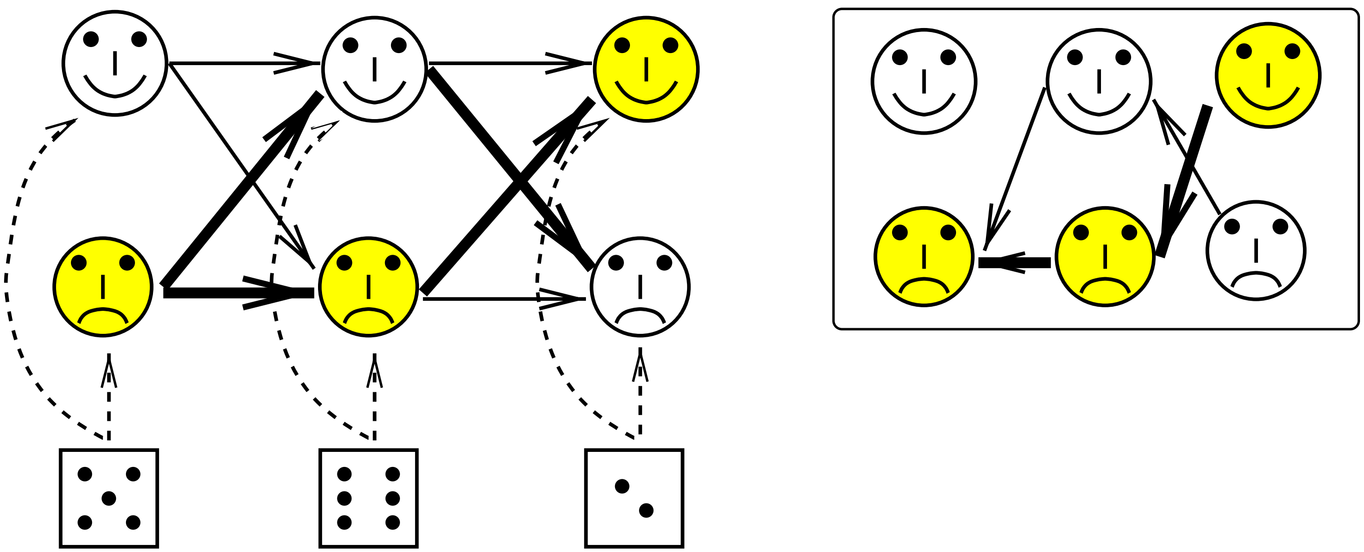
Viterbi Algorithm: Demo





Initialisation	$\gamma_1(S_1) = P(y_1 \mid S_1)P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \mid S_n) \max_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \mid S_{n-1})\gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

Viterbi Algorithm: Demo



Initialisation	$\gamma_1(S_1) = P(y_1 \vert S_1) P(S_1)$
Recursion $(n = 2, \ldots, N)$	$\gamma_n(S_n) = P(y_n \vert S_n) \max_{S_{n-1}} P(S_n \vert S_{n-1}) \gamma_{n-1}(S_{n-1})$ $\text{Pointer}(S_{n-1}) = \text{argmax}_{S_{n-1}} P(S_n \vert S_{n-1}) \gamma_{n-1}(S_{n-1})$
Termination	$\hat{S}_N = \text{argmax}_{S_N} \gamma_N(S_N)$
Traceback $(n = N-1, \ldots, 1)$	$\hat{S}_n = \text{Pointer}(\hat{S}_{n+1})$

Logarithmic Version of the Viterbi Algorithm

Prevent numerical underflow

$$\max_{S_1,\ldots,S_N} P(S_1,\ldots,S_N|y_1,\ldots,y_N) = \max_{S_1,\ldots,S_N} \log P(S_1,\ldots,S_N,y_1,\ldots,y_N) = \max_{S_N} \log \gamma_N(S_N)$$

$$\begin{aligned}
\gamma_n(S_n) &= \max_{S_1,\ldots,S_{n-1}} P(y_1,\ldots,y_n,S_1,\ldots,S_n) \\
&= \max_{S_1,\ldots,S_{n-1}} \prod_{t=1}^{n} P(y_t|S_t)P(S_t|S_{t-1}) \\
&= P(y_n|S_n) \max_{S_{n-1}} P(S_n|S_{n-1})\gamma_{n-1}(S_{n-1}) \\
\log \gamma_n(S_n) &= \max_{S_1,\ldots,S_{n-1}} \sum_{t=1}^{n} \Big[\log P(y_t|S_t) + \log P(S_t|S_{t-1}) \Big] \\
&= \log P(y_n|S_n) + \max_{S_{n-1}} [\log P(S_n|S_{n-1}) + \log \gamma_{n-1}(S_{n-1})]
\end{aligned}$$

Computing the Likelihood: The Forward Algorithm

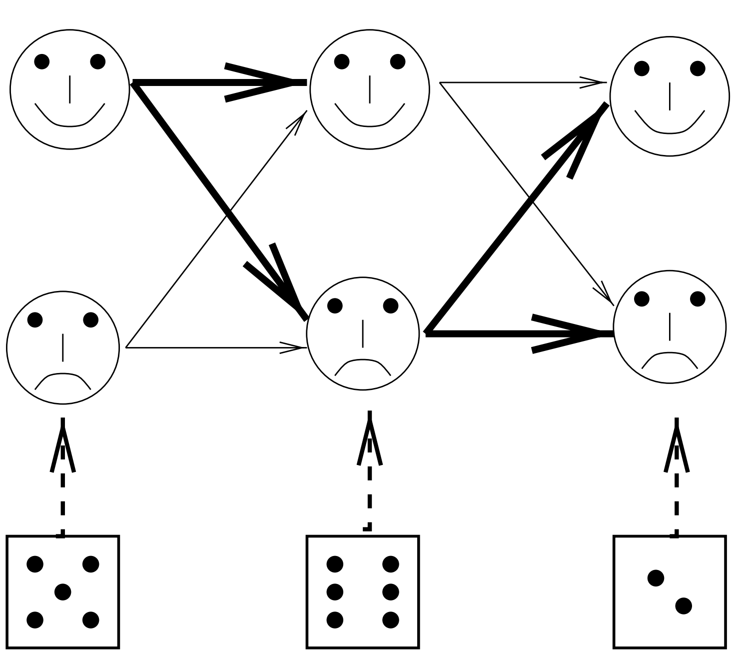
$$P(y_1, \ldots, y_N) = \sum_{S_1, \ldots, S_N} P(y_1, \ldots, y_N, S_1, \ldots, S_N)$$

Computing the Likelihood: The Forward Algorithm

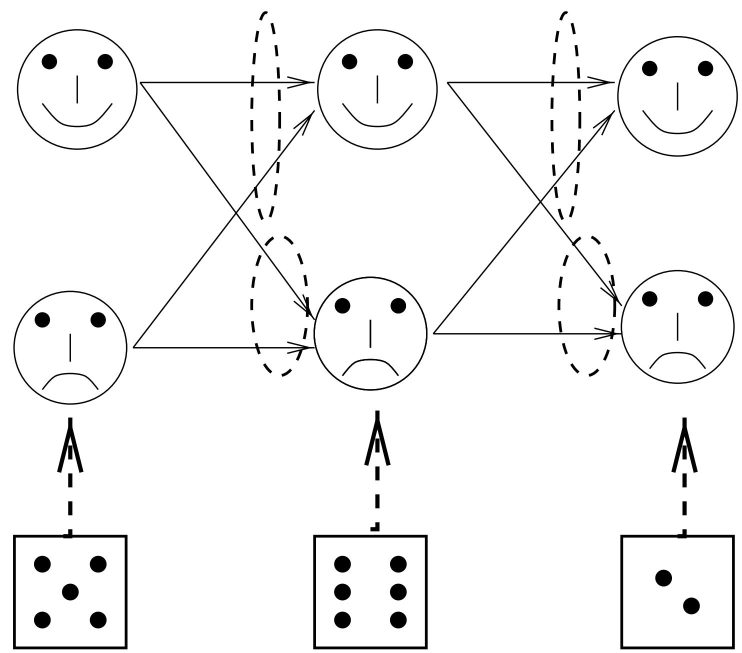
$$P(y_1, \ldots, y_N) = \sum_{S_1, \ldots, S_N} P(y_1, \ldots, y_N, S_1, \ldots, S_N)$$

$$\begin{aligned}
P(y_1, \ldots, y_N) &= \sum_{S_N} \alpha_N(S_N) \\
\alpha_n(S_n) &= P(y_1, \ldots, y_n, S_n) \\
&= \sum_{S_1} \ldots \sum_{S_{n-1}} P(y_1, \ldots, y_n, S_1, \ldots, S_{n-1}, S_n) \\
&= \sum_{S_1} \ldots \sum_{S_{n-1}} \prod_{t=1}^{n} P(y_t|S_t) P(S_t|S_{t-1}) \\
&= \sum_{S_1} \ldots \sum_{S_{n-1}} P(y_n|S_n) P(S_n|S_{n-1}) \prod_{t=1}^{n-1} P(y_t|S_t) P(S_t|S_{t-1}) \\
&= P(y_n|S_n) \sum_{S_{n-1}} P(S_n|S_{n-1}) \sum_{S_1} \ldots \sum_{S_{n-2}} \prod_{t=1}^{n-1} P(y_t|S_t) P(S_t|S_{t-1}) \\
&= P(y_n|S_n) \sum_{S_{n-1}} P(S_n|S_{n-1}) \alpha_{n-1}(S_{n-1})
\end{aligned}$$

Comparison between the Viterbi and the Forward Algorithms



Viterbi Algorithm:
Maximisation



Forward Algorithm:
Summation