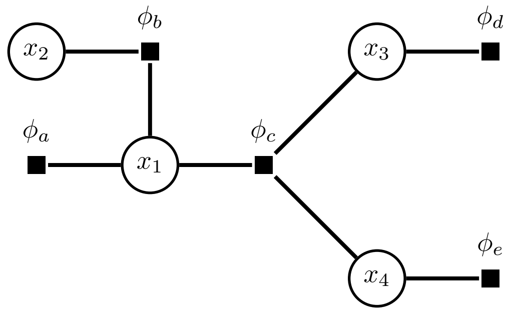
Exercise 1. ***Sum-product message passing***

The following factor graph represents a Gibbs distribution over four binary variables $x_i \in \{0, 1\}$.



The factors ϕ_a, ϕ_b, ϕ_d are defined as follows:

x_1	ϕ_a
0	2
1	1

x_1	x_2	ϕ_b
0	0	5
1	0	2
0	1	2
1	1	6

x_3	ϕ_d
0	1
1	2

and $\phi_c(x_1, x_3, x_4) = 1$ if $x_1 = x_3 = x_4$, and is zero otherwise.

For all questions below, justify your answer:

(a) *Compute the values of $\mu_{x_2 \to \phi_b}(x_2)$ for $x_2 = 0$ and $x_2 = 1$.*

Solution. Messages from leaf-variable nodes to factor nodes are equal to one, so that $\mu_{x_2 \to \phi_b}(x_2) = 1$ for all x_2.

(b) *Assume the message $\mu_{x_4 \to \phi_c}(x_4)$ equals*

$$\mu_{x_4 \to \phi_c}(x_4) = \begin{cases} 1 & \text{if } x_4 = 0 \\ 3 & \text{if } x_4 = 1 \end{cases}$$

Compute the values of $\phi_e(x_4)$ for $x_4 = 0$ and $x_4 = 1$.

Solution. Messages from leaf-factors to their variable nodes are equal to the leaf-factors, and variable nodes with single incoming messages copy the message. We thus have

$$\mu_{\phi_e \to x_4}(x_4) = \phi_e(x_4) \tag{S.1}$$

$$\mu_{x_4 \to \phi_c}(x_4) = \mu_{\phi_e \to x_4}(x_4) \tag{S.2}$$

and hence

$$\phi_e(x_4) = \begin{cases} 1 & \text{if } x_4 = 0 \\ 3 & \text{if } x_4 = 1 \end{cases} \tag{S.3}$$

(c) *Compute the values of $\mu_{\phi_c \to x_1}(x_1)$ for $x_1 = 0$ and $x_1 = 1$.*

Solution. We first compute $\mu_{x_3 \to \phi_c}(x_3)$:

$$\mu_{x_3 \to \phi_c}(x_3) = \mu_{\phi_d \to x_3}(x_3) \tag{S.4}$$
$$= \begin{cases} 1 & \text{if } x_3 = 0 \\ 2 & \text{if } x_3 = 1 \end{cases} \tag{S.5}$$

The desired message $\mu_{\phi_c \to x_1}(x_1)$ is by definition

$$\mu_{\phi_c \to x_1}(x_1) = \sum_{x_3, x_4} \phi_c(x_1, x_3, x_4) \mu_{x_3 \to \phi_c}(x_3) \mu_{x_4 \to \phi_c}(x_4) \tag{S.6}$$

Since $\phi_c(x_1, x_3, x_4)$ is only non-zero if $x_1 = x_3 = x_4$, where it equals one, the computations simplify:

$$\mu_{\phi_c \to x_1}(x_1 = 0) = \phi_c(0, 0, 0) \mu_{x_3 \to \phi_c}(0) \mu_{x_4 \to \phi_c}(0) \tag{S.7}$$
$$= 1 \cdot 1 \cdot 1 \tag{S.8}$$
$$= 1 \tag{S.9}$$

$$\mu_{\phi_c \to x_1}(x_1 = 1) = \phi_c(1, 1, 1) \mu_{x_3 \to \phi_c}(1) \mu_{x_4 \to \phi_c}(1) \tag{S.10}$$
$$= 1 \cdot 2 \cdot 3 \tag{S.11}$$
$$= 6 \tag{S.12}$$

(d) The message $\mu_{\phi_b \to x_1}(x_1)$ equals

$$\mu_{\phi_b \to x_1}(x_1) = \begin{cases} 7 & \textit{if } x_1 = 0 \\ 8 & \textit{if } x_1 = 1 \end{cases}$$

What is the probability that $x_1 = 1$, i.e. $p(x_1 = 1)$?

Solution. The unnormalised marginal $p(x_1)$ is given by the product of the three incoming messages

$$p(x_1) \propto \mu_{\phi_a \to x_1}(x_1) \mu_{\phi_b \to x_1}(x_1) \mu_{\phi_c \to x_1}(x_1) \tag{S.13}$$

With

$$\mu_{\phi_b \to x_1}(x_1) = \sum_{x_2} \phi_b(x_1, x_2) \tag{S.14}$$

it follows that

$$\mu_{\phi_b \to x_1}(x_1 = 0) = \sum_{x_2} \phi_b(0, x_2) \tag{S.15}$$
$$= 5 + 2 \tag{S.16}$$
$$= 7 \tag{S.17}$$
$$\mu_{\phi_b \to x_1}(x_1 = 1) = \sum_{x_2} \phi_b(1, x_2) \tag{S.18}$$
$$= 2 + 6 \tag{S.19}$$
$$= 8 \tag{S.20}$$

Hence, we obtain

$$p(x_1 = 0) \propto 2 \cdot 7 \cdot 1 = 14 \tag{S.21}$$
$$p(x_1 = 1) \propto 1 \cdot 8 \cdot 6 = 48 \tag{S.22}$$

and normalisation yields the desired result

$$p(x_1 = 1) = \frac{48}{14 + 48} = \frac{48}{62} = \frac{24}{31} = 0.774 \tag{S.23}$$

Exercise 2. ***Viterbi algorithm***

For the hidden Markov model

$$p(h_{1:t}, v_{1:t}) = p(v_1|h_1)p(h_1) \prod_{i=2}^{t} p(v_i|h_i)p(h_i|h_{i-1})$$

assume you have observations for v_i, $i = 1, \ldots, t$. Use the max-sum algorithm to derive an iterative algorithm to compute

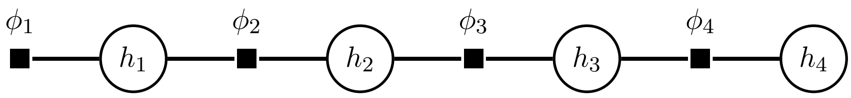
$$\hat{\mathbf{h}} = \underset{h_1,\ldots,h_t}{\operatorname{argmax}} \, p(h_{1:t}|v_{1:t}) \tag{1}$$

Assume that the latent variables h_i can take K different values, e.g. $h_i \in \{0, \ldots, K-1\}$. The resulting algorithm is known as Viterbi algorithm.

Solution. We first form the factors

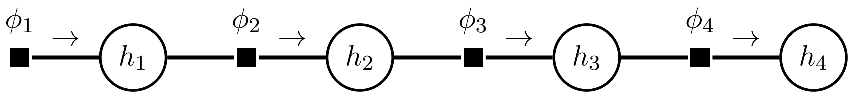
$$\phi_1(h_1) = p(v_1|h_1)p(h_1) \qquad \phi_2(h_1, h_2) = p(v_2|h_2)p(h_2|h_1) \tag{S.24}$$
$$\ldots \qquad \phi_t(h_{t-1}, h_t) = p(v_t|h_t)p(h_t|h_{t-1}) \tag{S.25}$$

where the v_i are known and fixed. The posterior $p(h_1, \ldots, h_t|v_1, \ldots, v_t)$ is then represented by the following factor graph (assuming $t = 4$).



For the max-sum algorithm, we here choose h_t to be the "sink". We thus initialise the algorithm with $\gamma_{\phi_1 \to h_1}(h_1) = \log \phi_1(h_1) = \log p(v_1|h_1) + \log p(h_1)$ and then compute the messages from left to right, moving from the leaf ϕ_1 to h_t.

Since we are dealing with a chain, the variable nodes, much like in the sum-product algorithm, just copy the incoming messages. It thus suffices to compute the factor to variable messages shown in the graph, and then backtrack to h_1.



With $\gamma_{h_{i-1}\to\phi_i}(h_{i-1}) = \gamma_{\phi_{i-1}\to h_{i-1}}(h_{i-1})$, the factor-to-variable update equation is

$$\gamma_{\phi_i\to h_i}(h_i) = \max_{h_{i-1}} \log\phi_i(h_{i-1}, h_i) + \gamma_{h_{i-1}\to\phi_i}(h_{i-1}) \tag{S.26}$$

$$= \max_{h_{i-1}} \log\phi_i(h_{i-1}, h_i) + \gamma_{\phi_{i-1}\to h_{i-1}}(h_{i-1}) \tag{S.27}$$

To simplify notation, denote $\gamma_{\phi_i\to h_i}(h_i)$ by $V_i(h_i)$. We thus have

$$V_1(h_1) = \log p(v_1|h_1) + \log p(h_1) \tag{S.28}$$

$$V_i(h_i) = \max_{h_{i-1}} \log\phi_i(h_{i-1}, h_i) + V_{i-1}(h_{i-1}) \qquad i = 2,\dots,t \tag{S.29}$$

In general, $V_1(h_1)$ and $V_i(h_i)$ are functions that depend on h_1 and h_i, respectively. Assuming that the h_i can take on the values $0,\dots,K-1$, the above equations can be written as

$$v_{1,k} = \log p(v_1|k) + \log p(k) \qquad k = 0,\dots,K-1 \tag{S.30}$$

$$v_{i,k} = \max_{m\in 0,\dots,K-1} \log\phi_i(m,k) + v_{i-1,m} \qquad k = 0,\dots,K-1, \quad i = 2,\dots,t, \tag{S.31}$$

At the end of the algorithm, we thus have a $t\times K$ matrix $\mathbf{V}$ with elements $v_{i,k}$.

The maximisation can be performed by computing the temporary matrix $\mathbf{A}$ (via broadcasting) where the (m,k)-th element is $\log\phi_i(m,k) + v_{i-1,m}$. Maximisation then corresponds to determining the maximal value in each column.

To support the backtracking, when we compute $V_i(h_i)$ by maximising over h_{i-1}, we compute at the same time the look-up table

$$\gamma_i^*(h_i) = \operatorname*{argmax}_{h_{i-1}} \log\phi_i(h_{i-1}, h_i) + V_{i-1}(h_{i-1}) \tag{S.32}$$

When h_i takes on the values $0,\dots,K-1$, this can be written as

$$\gamma_{i,k}^* = \operatorname*{argmax}_{m\in 0,\dots,K-1} \log\phi_i(m,k) + v_{i-1,m} \tag{S.33}$$

This is the (row) index of the maximal element in each column of the temporary matrix $\mathbf{A}$.

After computing $v_{t,k}$ and $\gamma_{t,k}^*$, we then perform backtracking via

$$\hat{h}_t = \operatorname*{argmax}_{k} v_{t,k} \tag{S.34}$$

$$\hat{h}_i = \gamma^*_{i+1,\hat{h}_{i+1}} \qquad i = t-1,\dots,1 \tag{S.35}$$

This gives recursively $\hat{\mathbf{h}} = (\hat{h}_1,\dots,\hat{h}_t) = \operatorname{argmax}_{h_1,\dots,h_t} p(h_{1:t}|v_{1:t})$.