Machine Learning 2 - Homework 4

Andrii Skliar, 11636785

deadline: April 30, 2018

During the process of solving the homework problems, I have collaborated with the following colleagues:

Pascal Esser   Sindy Loewe   Linda Petrini

NB: credits for the Latex-format go to Iris Verweij, 2nd year MSc AI Student.

Problem 1: Given the Bayesian network in Figure, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$:

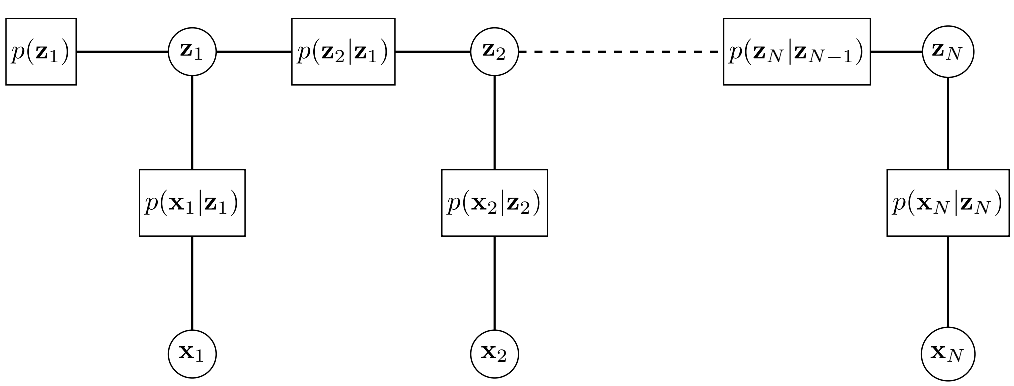
1. Write down the factorized joint probability distribution $p(\mathbf{Z}, \mathbf{X})$.

Solution:

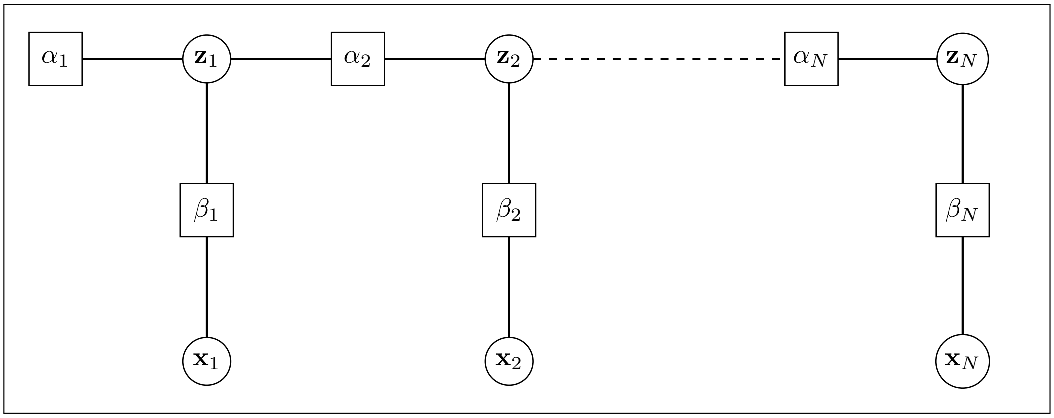
$$p(\mathbf{Z}, \mathbf{X}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1)\prod_{i=2}^{N} p(\mathbf{z}_i|\mathbf{z}_{i-1})p(\mathbf{x}_i|\mathbf{z}_i)$$

2. Draw the the corresponding factor graph.

Solution: First, let's draw factor graph with probabilities. Afterwards, they will be converted to proper factors.



Now, I would like to draw factor graph with α and β introduced as factors.



3. Write down the the joint probability distribution using the factors introduced in 2.

Solution:

$$\begin{aligned} p(\mathbf{Z}, \mathbf{X}) &= f_{\alpha_1}(\mathbf{z}_1) \prod_{i=1}^{N} f_{\beta_i}(\mathbf{x}_i, \mathbf{z}_i) \prod_{i=2}^{N} f_{\alpha_i}(\mathbf{z}_i, \mathbf{z}_{i-1}) \\ &= f_{\alpha_1}(\mathbf{z}_1) f_{\beta_1}(\mathbf{x}_1, \mathbf{z}_1) \prod_{i=2}^{N} f_{\beta_i}(\mathbf{x}_i, \mathbf{z}_i) f_{\alpha_i}(\mathbf{z}_i, \mathbf{z}_{i-1}) \end{aligned}$$

Note: I am not including normalizing factor $\frac{1}{z}$, because our potentials are probabilities.

4. Given $\mathbf{X}$, we want to infer z_n such that

$$\begin{aligned} p(z_n|\mathbf{X}) &= \frac{p(\mathbf{X}|z_n)p(z_n)}{p(\mathbf{X})} \\ &= \frac{\alpha(z_n)\beta(z_n)}{p(\mathbf{X})} \end{aligned}$$

Using the conditional independencies of the graph in Figure, derive $\alpha(z_n)$ and $\beta(z_n)$ so that they are recursive denitions of themselves, i.e. $\alpha(z_n)$ is calculated from $\alpha(z_{n-1})$ and $\beta(z_n)$ is calculated from $\beta(z_{n+1})$. Indicate where you use independencies inferred from the graphical model.

Solution:

$$
\begin{aligned}
p(z_n|\mathbf{X}) &= \frac{p(\mathbf{X}|z_n)p(z_n)}{p(\mathbf{X})} \\
&= \frac{p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N|\mathbf{z}_n)p(\mathbf{z}_n)}{p(\mathbf{X})} \\
&= \frac{p(\mathbf{x}_1, \dots \mathbf{x}_n|\mathbf{z}_n)p(\mathbf{x}_{n+1}, \dots \mathbf{x}_N|\mathbf{z}_n)p(\mathbf{z}_n)}{p(\mathbf{X})} \\
&= \frac{p(\mathbf{x}_1, \dots \mathbf{x}_n, \mathbf{z}_n)p(\mathbf{x}_{n+1}, \dots \mathbf{x}_N|\mathbf{z}_n)}{p(\mathbf{X})} \\
\implies \alpha(z_n) &= p(\mathbf{x}_1, \dots \mathbf{x}_n, \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots \mathbf{x}_n, \mathbf{z}_n, \mathbf{z}_{n-1}) \\
&= \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots \mathbf{x}_n, \mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{z}_{n-1}) \\
&= \{\text{Using d-separation of } \{\mathbf{x}_1, \dots \mathbf{x}_{n-1}\} \text{ and } \{\mathbf{x}_n, \mathbf{z}_n\} \text{ given } \mathbf{z}_{n-1}\} \\
&= \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots \mathbf{x}_{n-1}|\mathbf{z}_{n-1})p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{z}_{n-1}) \\
&= \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots \mathbf{x}_{n-1}, \mathbf{z}_{n-1})p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{z}_{n-1}) \\
&= \sum_{\mathbf{z}_{n-1}} \alpha(z_{n-1})p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{z}_{n-1}) \\
\implies \beta(z_n) &= p(\mathbf{x}_{n+1}, \dots \mathbf{x}_N|\mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \dots \mathbf{x}_N, \mathbf{z}_{n+1}|\mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} \frac{p(\mathbf{x}_{n+1}, \dots \mathbf{x}_N, \mathbf{z}_{n+1}, \mathbf{z}_n)}{p(\mathbf{z}_n)} \\
&= \{\text{Using d-separation of } \{\mathbf{x}_{n+2}, \dots \mathbf{x}_N\} \text{ and } \{\mathbf{x}_{n+1}, \mathbf{z}_n\} \text{ given } \mathbf{z}_{n+1}\} \\
&= \sum_{\mathbf{z}_{n+1}} \frac{p(\mathbf{x}_{n+1}, \mathbf{z}_n|\mathbf{z}_{n+1})p(\mathbf{x}_{n+2}, \dots \mathbf{x}_N|\mathbf{z}_{n+1})p(\mathbf{z}_{n+1})}{p(\mathbf{z}_n)} \\
&= \sum_{\mathbf{z}_{n+1}} \frac{p(\mathbf{x}_{n+1}, \mathbf{z}_n, \mathbf{z}_{n+1})p(\mathbf{x}_{n+2}, \dots \mathbf{x}_N|\mathbf{z}_{n+1})}{p(\mathbf{z}_n)} \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \mathbf{z}_{n+1}|\mathbf{z}_n)p(\mathbf{x}_{n+2}, \dots \mathbf{x}_N|\mathbf{z}_{n+1}) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \mathbf{z}_{n+1}|\mathbf{z}_n)\beta(z_{n+1})
\end{aligned}
$$

Therefore:

$$
\begin{aligned}
\alpha(z_n) &= \sum_{\mathbf{z}_{n-1}} \alpha(z_{n-1})p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{z}_{n-1}) \\
\beta(z_n) &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \mathbf{z}_{n+1}|\mathbf{z}_n)\beta(z_{n+1})
\end{aligned}
$$

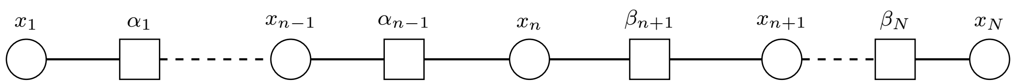
Problem 2:

1. Apply the sum-product algorithm (as in Bishop's section 8.4.4) to the chain of nodes model in Figure and show that the results of message passing algorithm (as in Bishop's section 8.4.1) are recovered as a special case, that is:

$$p(x_n) = \frac{1}{Z}\mu_\alpha(x_n)\mu_\beta(x_n)$$
$$\mu_\alpha(x_n) = \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n)\mu_\alpha(x_{n-1})$$
$$\mu_\beta(x_n) = \sum_{x_{n+1}} \psi_{n+1,n}(x_{n+1}, x_n)\mu_\beta(x_{n+1})$$

where $\psi_{i,i+1}(x_i, x_{i+1})$ is a potential function dened over clique $\{x_i, x_{i+1}\}$

Solution: First, I would like to rewrite model to be a factor graph.



Now, let's apply sum-product algorithm to a given factor graph.
First, we have to do a forward pass:

$$\mu_{x_1\to\alpha_1}(x_1) = 1$$
$$\mu_{\alpha_1\to x_2}(x_2) = \sum_{x_1} f_{\alpha_1}(x_1, x_2)\mu_{x_1\to\alpha_1}(x_1) = \sum_{x_1} f_{\alpha_1}(x_1, x_2)$$
$$\mu_{x_2\to\alpha_2}(x_2) = \mu_{\alpha_1\to x_2}(x_2) = \sum_{x_1} f_{\alpha_1}(x_1, x_2)$$
$$\mu_{\alpha_2\to x_3}(x_3) = \sum_{x_2} f_{\alpha_2}(x_2, x_3)\mu_{x_2\to\alpha_2}(x_2)$$
$$\vdots$$
$$\mu_{\alpha_{n-1}\to x_n}(x_n) = \sum_{x_{n-1}} f_{\alpha_{n-1}}(x_{n-1}, x_n)\mu_{x_{n-1}\to\alpha_{n-1}}(x_{n-1})$$

Applying the same logic to μ_β, we get following value for $\mu_{\beta_{n+1}\to x_n}$:

$$\mu_{\beta_{n+1}\to x_n}(x_n) = \sum_{x_{n+1}} f_{\beta_{n+1}}(x_{n+1}, x_n)\mu_{x_{n+1}\to\beta_{n+1}}(x_{n+1})$$

Therefore, we get following equations for $\mu_{\alpha_{n-1}\to x_n}(x_n)$ and $\mu_{\beta_{n+1}\to x_n}(x_n)$:

$$\forall n \in \{2, \dots N\} : \mu_{\alpha_{n-1}\to x_n}(x_n) = \sum_{x_{n-1}} f_{\alpha_{n-1}}(x_{n-1}, x_n)\mu_{x_{n-1}\to\alpha_{n-1}}(x_{n-1})$$
$$\forall n \in \{1, \dots N-1\} : \mu_{\beta_{n+1}\to x_n}(x_n) = \sum_{x_{n+1}} f_{\beta_{n+1}}(x_{n+1}, x_n)\mu_{x_{n+1}\to\beta_{n+1}}(x_{n+1})$$

Because these equations are valid for specified values of n, we don't have to do backward pass explicitly and can write the variable beliefs directly:

$$p(x_n) = \frac{1}{Z}\mu_{\alpha_{n-1}\to x_n}(x_n) \cdot \mu_{\beta_{n+1}\to x_n}(x_n), \text{where } Z \text{ is a normalizing factor.}$$

Note, that this formula won't be valid for $n = 1$ and $n = N$. However, to avoid the issues, we can specify:

$$\begin{cases} \mu_{\alpha_{n-1}\to x_n}(x_n) = 1, & \text{if } n = 1 \\ \mu_{\beta_{n+1}\to x_n}(x_n) = 1, & \text{if } n = N \end{cases}$$

It is easy to see correspondence between equations specified in the task and the ones, that we have derived.

$$\begin{aligned} \mu_\alpha(x_n) &= \mu_{\alpha_{n-1}\to x_n}(x_n) \\ \psi_{n-1,n}(x_{n-1}, x_n) &= f_{\alpha_{n-1}}(x_{n-1}, x_n) \\ \mu_\beta(x_n) &= \mu_{\beta_{n+1}\to x_n}(x_n) \\ \psi_{n+1,n}(x_{n+1}, x_n) &= f_{\beta_{n+1}}(x_{n+1}, x_n) \end{aligned}$$

2. Establish a relation of your results $\alpha(z_n)$ and $\beta(z_n)$ in 1.4 with the results of the sum-product algorithm $\mu_\alpha(z_n)$ and $\mu_\beta(z_n)$.

Solution:
Note: here x_n in equation for μ_α and μ_β is replaced with z_n to be consistent with task 1.
It is easy to see, that $\alpha(z_n)$ and $\beta(z_n)$ correspond directly to $\mu_\alpha(z_n)$ and $\mu_\beta(z_n)$ with following potentials:

$$\psi_{n-1,n}(z_{n-1}, z_n) = f_{\alpha_{n-1}}(z_{n-1}, z_n) = p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{z}_{n-1}), \qquad z_n = \{\mathbf{z}_n, \mathbf{x}_n\}, z_{n-1} = \mathbf{z}_{n-1}$$
$$\psi_{n+1,n}(z_{n+1}, z_n) = f_{\beta_{n+1}}(z_{n+1}, z_n) = p(\mathbf{x}_{n+1}, \mathbf{z}_{n+1}|\mathbf{z}_n), \quad z_{n+1} = \{\mathbf{z}_{n+1}, \mathbf{x}_{n+1}\}, z_n = \mathbf{z}_n$$

Also, for variable beliefs $p(z_n)$ to correspond to $p(\mathbf{z}_n|\mathbf{X})$, we need to specify normalizing constant, which in that case will be:

$$Z = p(\mathbf{X})$$

Problem 3: Consider the inference problem of evaluating $p(\mathbf{x}_n|\mathbf{x}_N)$ for the graph shown in Figure, for all nodes $n \in \{1, \dots, N-1\}$. Show that the message passing algorithm can be used to solve this eciently, and discuss which messages are modied and in what way.

Solution: From my point of understanding of the question, I think, that efficiency here deals with the fact that there are no additional messages that have to be passed to evaluate $p(\mathbf{x}_n|\mathbf{x}_N)$. Saying so, I assume that the message passing algorithm is already considered to be efficient for the evaluation of $p(\mathbf{x}_n)$.
If this assumption is considered as given, then to show that it can be used efficiently, the simplest way would be to specify how messages have to be changed to incorporate conditioning on $\mathbf{x}_N$.
Let's say, that we are insterested in $\mathbf{x}_N$ to have value of ξ.
In that case, $p(\mathbf{x}_n|\mathbf{x}_N) = p(\mathbf{x}_n, \mathbf{x}_N)\mathbb{1}(\mathbf{x}_N = \xi)$. To include it into the calculation of variable beliefs, we can change potential functions:

$$\psi_{N-1,N}(\mathbf{x}_{N-1}, \mathbf{x}_N) = \psi_{N-1,N}(\mathbf{x}_{N-1}, \mathbf{x}_N) \cdot \mathbb{1}(\mathbf{x}_N = \xi)$$

This will also change $\mu_\alpha(\mathbf{x}_N)$ and $\mu_\beta(\mathbf{x}_N)$ as sums will now contain one term, which has $\mathbf{x}_N = \xi$ and therefore, $p(z_n)$ will change as well. This means, that applying the same message passing algorithm, we can efficiently calculate conditional probabilities. However, note, that to make probabilities correct, we would have to re-normalize $p(\mathbf{x}_n|\mathbf{x}_N)$ by $p(\mathbf{x}_N = \xi)$.
So, as you can see, there is almost nothing, that has to be changed in order to incorporate conditioning on a specific random variable and therefore, the complexity of the message passing algorithm stays the same, which means, that it still provides an efficient way of evaluating probabilities for the given graphical model.

Problem 4: Show that the marginal distribution for the variables $\mathbf{x}_s$ in a factor $f_s(\mathbf{x}_s)$ in a tree-structured factor graph, after running the sum-product message passing algorithm, can be written as

$$p(\mathbf{x}_s) = f_s(\mathbf{x}_s) \prod_{i \in ne(f_s)} \mu_{x_i \to f_s(x_i)}$$

where $ne(f_s)$ denotes the set of variable nodes that are neighbors of the factor node f_s.

Solution: Using the same way of reasoning and notation as provided in Bishop section 8.4:

$$\begin{aligned}
p(\mathbf{x}_s) &= f_s(\mathbf{x}_s) \prod_{i \in ne(f_s)} \mu_{x_i \to f_s(x_i)} \\
&= f_s(\mathbf{x}_s) \prod_{i \in ne(f_s)} \prod_{l \in ne(x_i) \setminus f_s} \mu_{f_l(x_i) \to x_i} \\
&= f_s(\mathbf{x}_s) \prod_{i \in ne(f_s)} \prod_{l \in ne(x_i) \setminus f_s} \Big[\sum_{X_l} F_l(x_i, X_l) \Big] \\
&= \sum_{\mathbf{x} \setminus \mathbf{x}_s} f_s(\mathbf{x}_s) \prod_{i \in ne(f_s)} \prod_{l \in ne(x_i) \setminus f_s} F_l(x_i, X_l) \\
&= \sum_{\mathbf{x} \setminus \mathbf{x}_s} p(\mathbf{x}) \\
&= p(\mathbf{x}_s)
\end{aligned}$$