Pen & Paper

Exercises in Machine Learning

Michael U. Gutmann

University of Edinburgh

Contents

Preface

We may have all heard the saying "use it or lose it". We experience it when we feel rusty in a foreign language or sports that we have not practised in a while. Practice is important to maintain skills but it is also key when learning new ones. This is a reason why many textbooks and courses feature exercises. However, the solutions to the exercises feel often overly brief, or are sometimes not available at all. Rather than an opportunity to practice the new skills, the exercises then become a source of frustration and are ignored.

This book contains a collection of exercises with *detailed* solutions. The level of detail is, hopefully, sufficient for the reader to follow the solutions and understand the techniques used. The exercises, however, are not a replacement of a textbook or course on machine learning. I assume that the reader has already seen the relevant theory and concepts and would now like to deepen their understanding through solving exercises.

While coding and computer simulations are extremely important in machine learning, the exercises in the book can (mostly) be solved with pen and paper. The focus on pen-and-paper exercises reduced length and simplified the presentation. Moreover, it allows the reader to strengthen their mathematical skills. However, the exercises are ideally paired with computer exercises to further deepen the understanding.

The exercises collected here are mostly a union of exercises that I developed for the courses "Unsupervised Machine Learning" at the University of Helsinki and "Probabilistic Modelling and Reasoning" at the University of Edinburgh. The exercises do not comprehensively cover all of machine learning but focus strongly on unsupervised methods, inference and learning.

I am grateful to my students for providing feedback and asking questions. Both helped to improve the quality of the exercises and solutions. I am further grateful to both universities for providing the research and teaching environment.

My hope is that the collection of exercises will grow with time. I intend to add new exercises in the future and welcome contributions from the community. Latex source code is available at `https://github.com/michaelgutmann/ml-pen-and-paper-exercises`. Please use GitHub's issues to report mistakes or typos, and please get in touch if you would like to make larger contributions.

Michael Gutmann
Edinburgh, June 2022
`https://michaelgutmann.github.io`

Chapter 1

Linear Algebra

Exercises

1.1 Gram–Schmidt orthogonalisation

(a) Given two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ in $\mathbb{R}^n$, show that

$$\mathbf{u}_1 = \mathbf{a}_1 \tag{1.1}$$

$$\mathbf{u}_2 = \mathbf{a}_2 - \frac{\mathbf{u}_1^\top \mathbf{a}_2}{\mathbf{u}_1^\top \mathbf{u}_1}\mathbf{u}_1 \tag{1.2}$$

are orthogonal to each other.

Solution. Two vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ of $\mathbb{R}^n$ are orthogonal if their inner product equals zero. Computing the inner product $\mathbf{u}_1^\top \mathbf{u}_2$ gives

$$\mathbf{u}_1^\top \mathbf{u}_2 = \mathbf{u}_1^\top (\mathbf{a}_2 - \frac{\mathbf{u}_1^\top \mathbf{a}_2}{\mathbf{u}_1^\top \mathbf{u}_1}\mathbf{u}_1) \tag{S.1.1}$$

$$= \mathbf{u}_1^\top \mathbf{a}_2 - \frac{\mathbf{u}_1^\top \mathbf{a}_2}{\mathbf{u}_1^\top \mathbf{u}_1}\mathbf{u}_1^\top \mathbf{u}_1 \tag{S.1.2}$$

$$= \mathbf{u}_1^\top \mathbf{a}_2 - \mathbf{u}_1^\top \mathbf{a}_2 \tag{S.1.3}$$

$$= 0. \tag{S.1.4}$$

Hence the vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthogonal.

If $\mathbf{a}_2$ is a multiple of $\mathbf{a}_1$, the orthogonalisation procedure produces a zero vector for $\mathbf{u}_2$. To see this, let $\mathbf{a}_2 = \alpha \mathbf{a}_1$ for some real number α. We then obtain

$$\mathbf{u}_2 = \mathbf{a}_2 - \frac{\mathbf{u}_1^\top \mathbf{a}_2}{\mathbf{u}_1^\top \mathbf{u}_1}\mathbf{u}_1 \tag{S.1.5}$$

$$= \alpha \mathbf{u}_1 - \frac{\alpha \mathbf{u}_1^\top \mathbf{u}_1}{\mathbf{u}_1^\top \mathbf{u}_1}\mathbf{u}_1 \tag{S.1.6}$$

$$= \alpha \mathbf{u}_1 - \alpha \mathbf{u}_1 \tag{S.1.7}$$

$$= \mathbf{0}. \tag{S.1.8}$$

(b) Show that any linear combination of (linearly independent) $\mathbf{a}_1$ and $\mathbf{a}_2$ can be written in terms of $\mathbf{u}_1$ and $\mathbf{u}_2$.

Solution. Let $\mathbf{v}$ be a linear combination of $\mathbf{a}_1$ and $\mathbf{a}_2$, i.e. $\mathbf{v} = \alpha \mathbf{a}_1 + \beta \mathbf{a}_2$ for some real numbers α and β. Expressing $\mathbf{u}_1$ and $\mathbf{u}_2$ in term of $\mathbf{a}_1$ and $\mathbf{a}_2$, we can write $\mathbf{v}$ as

$$\mathbf{v} = \alpha \mathbf{a}_1 + \beta \mathbf{a}_2 \tag{S.1.9}$$

$$= \alpha \mathbf{u}_1 + \beta(\mathbf{u}_2 + \frac{\mathbf{u}_1^\top \mathbf{a}_2}{\mathbf{u}_1^\top \mathbf{u}_1}\mathbf{u}_1) \tag{S.1.10}$$

$$= \alpha \mathbf{u}_1 + \beta \mathbf{u}_2 + \beta \frac{\mathbf{u}_1^\top \mathbf{a}_2}{\mathbf{u}_1^\top \mathbf{u}_1}\mathbf{u}_1 \tag{S.1.11}$$

$$= (\alpha + \beta \frac{\mathbf{u}_1^\top \mathbf{a}_2}{\mathbf{u}_1^\top \mathbf{u}_1})\mathbf{u}_1 + \beta \mathbf{u}_2, \tag{S.1.12}$$

Since $\alpha + \beta((\mathbf{u}_1^\top \mathbf{a}_2)/(\mathbf{u}_1^\top \mathbf{u}_1))$ and β are real numbers, we can write $\mathbf{v}$ as a linear combination of $\mathbf{u}_1$ and $\mathbf{u}_2$. Overall, this means that any vector in the span of $\{\mathbf{a}_1, \mathbf{a}_2\}$ can be expressed in the orthogonal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$.

(c) Show by induction that for any $k \leq n$ linearly independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_k$, the vectors $\mathbf{u}_i$, $i = 1, \ldots k$, are orthogonal, where

$$\mathbf{u}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} \frac{\mathbf{u}_j^\top \mathbf{a}_i}{\mathbf{u}_j^\top \mathbf{u}_j} \mathbf{u}_j. \tag{1.3}$$

The calculation of the vectors $\mathbf{u}_i$ is called Gram–Schmidt orthogonalisation.

Solution. We have shown above that the claim holds for two vectors. This is the base case for the proof by induction. Assume now that the claim holds for k vectors. The induction step in the proof by induction then consists of showing that the claim also holds for $k + 1$ vectors.

Assume that $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ are orthogonal vectors. The linear independence assumption ensures that none of the $\mathbf{u}_i$ is a zero vector. We then have for $\mathbf{u}_{k+1}$

$$\mathbf{u}_{k+1} = \mathbf{a}_{k+1} - \frac{\mathbf{u}_1^\top \mathbf{a}_{k+1}}{\mathbf{u}_1^\top \mathbf{u}_1} \mathbf{u}_1 - \frac{\mathbf{u}_2^\top \mathbf{a}_{k+1}}{\mathbf{u}_2^\top \mathbf{u}_2} \mathbf{u}_2 - \ldots - \frac{\mathbf{u}_k^\top \mathbf{a}_{k+1}}{\mathbf{u}_k^\top \mathbf{u}_k} \mathbf{u}_k, \tag{S.1.13}$$

and for all $i = 1, 2, \ldots, k$

$$\mathbf{u}_i^\top \mathbf{u}_{k+1} = \mathbf{u}_i^\top \mathbf{a}_{k+1} - \frac{\mathbf{u}_1^\top \mathbf{a}_{k+1}}{\mathbf{u}_1^\top \mathbf{u}_1} \mathbf{u}_i^\top \mathbf{u}_1 - \ldots - \frac{\mathbf{u}_k^\top \mathbf{a}_{k+1}}{\mathbf{u}_k^\top \mathbf{u}_k} \mathbf{u}_i^\top \mathbf{u}_k. \tag{S.1.14}$$

By assumption $\mathbf{u}_i^\top \mathbf{u}_j = 0$ if $i \neq j$, so that

$$\begin{aligned} \mathbf{u}_i^\top \mathbf{u}_{k+1} &= \mathbf{u}_i^\top \mathbf{a}_{k+1} - 0 - \ldots - \frac{\mathbf{u}_i^\top \mathbf{a}_{k+1}}{\mathbf{u}_i^\top \mathbf{u}_i} \mathbf{u}_i^\top \mathbf{u}_i - 0 \ldots - 0 && \text{(S.1.15)} \\ &= \mathbf{u}_i^\top \mathbf{a}_{k+1} - \mathbf{u}_i^\top \mathbf{a}_{k+1} && \text{(S.1.16)} \\ &= 0, && \text{(S.1.17)} \end{aligned}$$

which means that $\mathbf{u}_{k+1}$ is orthogonal to $\mathbf{u}_1, \ldots, \mathbf{u}_k$.

(d) Show by induction that any linear combination of (linear independent) $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ can be written in terms of $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$.

Solution. The base case of two vectors was proved above. Using induction, we assume that the claim holds for k vectors and we will prove that it then also holds for $k + 1$ vectors: Let $\mathbf{v}$ be a linear combination of $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{k+1}$, i.e. $\mathbf{v} = \alpha_1 \mathbf{a}_1 + \alpha_2 \mathbf{a}_2 + \ldots + \alpha_k \mathbf{a}_k + \alpha_{k+1} \mathbf{a}_{k+1}$ for some real numbers $\alpha_1, \alpha_2, \ldots, \alpha_{k+1}$. Using the induction assumption, $\mathbf{v}$ can be written as

$$\mathbf{v} = \beta_1 \mathbf{u}_1 + \beta_2 \mathbf{u}_2 + \ldots + \beta_k \mathbf{u}_k + \alpha_{k+1} \mathbf{a}_{k+1}, \tag{S.1.18}$$

for some real numbers $\beta_1, \beta_2, \ldots, \beta_k$ Furthermore, using equation (S.1.13), $\mathbf{v}$ can be written as

$$\mathbf{v} = \beta_1 \mathbf{u}_1 + \ldots + \beta_k \mathbf{u}_k + \alpha_{k+1} \mathbf{u}_{k+1} + \alpha_{k+1} \frac{\mathbf{u}_1^\top \mathbf{a}_{k+1}}{\mathbf{u}_1^\top \mathbf{u}_1} \mathbf{u}_1 \tag{S.1.19}$$

$$+ \ldots + \alpha_{k+1} \frac{\mathbf{u}_k^\top \mathbf{a}_{k+1}}{\mathbf{u}_k^\top \mathbf{u}_k} \mathbf{u}_k. \tag{S.1.20}$$

With $\gamma_i = \beta_i + \alpha_{k+1}(\mathbf{u}_i^\top \mathbf{a}_{k+1})/(\mathbf{u}_i^\top \mathbf{u}_i)$, $\mathbf{v}$ can thus be written as

$$\mathbf{v} = \gamma_1 \mathbf{u}_1 + \gamma_2 \mathbf{u}_2 + \ldots + \gamma_k \mathbf{u}_k + \alpha_{k+1} \mathbf{u}_{k+1}, \tag{S.1.21}$$

which completes the proof. Overall, this means that the $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ form an orthogonal basis for $\text{span}(\mathbf{a}_1, \ldots, \mathbf{a}_k)$, i.e. the set of all vectors that can be obtained by linearly combining the $\mathbf{a}_i$.

(e) Consider the case where $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ are linearly independent and $\mathbf{a}_{k+1}$ is a linear combination of $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$. Show that $\mathbf{u}_{k+1}$, computed according to (1.3), is zero.

Solution. Starting with (1.3), we have

$$\mathbf{u}_{k+1} = \mathbf{a}_{k+1} - \sum_{j=1}^{k} \frac{\mathbf{u}_j^\top \mathbf{a}_{k+1}}{\mathbf{u}_j^\top \mathbf{u}_j} \mathbf{u}_j. \tag{S.1.22}$$

By assumption, $\mathbf{a}_{k+1}$ is a linear combination of $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$. By the previous question, it can thus also be written as a linear combination of the $\mathbf{u}_1, \ldots, \mathbf{u}_k$. This means that there are some β_i so that

$$\mathbf{a}_{k+1} = \sum_{i=1}^{k} \beta_i \mathbf{u}_i \tag{S.1.23}$$

holds. Inserting this expansion into the equation above gives

$$\mathbf{u}_{k+1} = \sum_{i=1}^{k} \beta_i \mathbf{u}_i - \sum_{j=1}^{k} \sum_{i=1}^{k} \beta_i \frac{\mathbf{u}_j^\top \mathbf{u}_i}{\mathbf{u}_j^\top \mathbf{u}_j} \mathbf{u}_j \tag{S.1.24}$$

$$= \sum_{i=1}^{k} \beta_i \mathbf{u}_i - \sum_{i=1}^{k} \beta_i \frac{\mathbf{u}_i^\top \mathbf{u}_i}{\mathbf{u}_i^\top \mathbf{u}_i} \mathbf{u}_i \tag{S.1.25}$$

because $\mathbf{u}_j^\top \mathbf{u}_i = 0$ if $i \neq j$. We thus obtain the desired result:

$$\mathbf{u}_{k+1} = \sum_{i=1}^{k} \beta_i \mathbf{u}_i - \sum_{i=1}^{k} \beta_i \mathbf{u}_i \tag{S.1.26}$$

$$= 0 \tag{S.1.27}$$

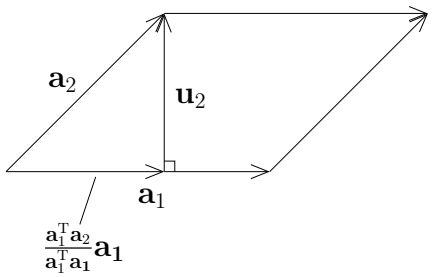
This property of the Gram-Schmidt process in (1.3) can be used to check whether a list of vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d$ is linearly independent or not. If, for example, $\mathbf{u}_{k+1}$ is zero, $\mathbf{a}_{k+1}$ is a linear combination of the $\mathbf{a}_1, \ldots, \mathbf{a}_k$. Moreover, the result can be used to extract a sublist of linearly independent vectors: We would remove $\mathbf{a}_{k+1}$ from the list and restart the procedure in (1.3) with $\mathbf{a}_{k+2}$ taking the place of $\mathbf{a}_{k+1}$. Continuing in this way constructs a list of linearly independent $\mathbf{a}_j$ and orthogonal $\mathbf{u}_j$, $j = 1, \ldots, r$, where r is the number of linearly independent vectors among the $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d$.

1.2 Linear transforms

(a) Assume two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ are in $\mathbb{R}^2$. Together, they span a parallelogram. Use Exercise 1.1 to show that the squared area S^2 of the parallelogram is given by

$$S^2 = (\mathbf{a}_2^T\mathbf{a}_2)(\mathbf{a}_1^T\mathbf{a}_1) - (\mathbf{a}_2^T\mathbf{a}_1)^2 \tag{1.4}$$

Solution. Let $\mathbf{a}_1$ and $\mathbf{a}_2$ be the vectors that span the parallelogram. From geometry we know that the area of parallelogram is base times height, which is equivalent to the length of the base vector times the length of the height vector. Denote this by $S^2 = ||\mathbf{a}_1||^2||\mathbf{u}_2||^2$, where is $\mathbf{a}_1$ is the base vector and $\mathbf{u}_2$ is the height vector which is orthogonal to the base vector. Using the Gram–Schmidt process for the vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ in that order, we obtain the vector $\mathbf{u}_2$ as the second output.



Therefore $||\mathbf{u}_2||^2$ equals

$$\begin{aligned}
||\mathbf{u}_2||^2 &= \mathbf{u}_2^\top\mathbf{u}_2 && \text{(S.1.28)}\\
&= \left(\mathbf{a}_2 - \frac{\mathbf{a}_1^\top\mathbf{a}_2}{\mathbf{a}_1^\top\mathbf{a}_1}\mathbf{a}_1\right)^\top\left(\mathbf{a}_2 - \frac{\mathbf{a}_1^\top\mathbf{a}_2}{\mathbf{a}_1^\top\mathbf{a}_1}\mathbf{a}_1\right) && \text{(S.1.29)}\\
&= \mathbf{a}_2^\top\mathbf{a}_2 - \frac{(\mathbf{a}_1^\top\mathbf{a}_2)^2}{\mathbf{a}_1^\top\mathbf{a}_1} - \frac{(\mathbf{a}_1^\top\mathbf{a}_2)^2}{\mathbf{a}_1^\top\mathbf{a}_1} + \left(\frac{\mathbf{a}_1^\top\mathbf{a}_2}{\mathbf{a}_1^\top\mathbf{a}_1}\right)^2\mathbf{a}_1^\top\mathbf{a}_1 && \text{(S.1.30)}\\
&= \mathbf{a}_2^\top\mathbf{a}_2 - \frac{(\mathbf{a}_1^\top\mathbf{a}_2)^2}{\mathbf{a}_1^\top\mathbf{a}_1}. && \text{(S.1.31)}
\end{aligned}$$

Thus, S^2 is:

$$\begin{aligned}
S^2 &= ||\mathbf{a}_1||^2||\mathbf{u}_2||^2 && \text{(S.1.32)}\\
&= (\mathbf{a}_1^\top\mathbf{a}_1)(\mathbf{u}_2^\top\mathbf{u}_2) && \text{(S.1.33)}\\
&= (\mathbf{a}_1^\top\mathbf{a}_1)\left(\mathbf{a}_2^\top\mathbf{a}_2 - \frac{(\mathbf{a}_1^\top\mathbf{a}_2)^2}{\mathbf{a}_1^\top\mathbf{a}_1}\right) && \text{(S.1.34)}\\
&= (\mathbf{a}_2^\top\mathbf{a}_2)(\mathbf{a}_1^\top\mathbf{a}_1) - (\mathbf{a}_1^\top\mathbf{a}_2)^2. && \text{(S.1.35)}
\end{aligned}$$

(b) Form the matrix $\mathbf{A} = (\mathbf{a}_1\ \mathbf{a}_2)$ where $\mathbf{a}_1$ and $\mathbf{a}_2$ are the first and second column vector, respectively. Show that

$$S^2 = (\det\mathbf{A})^2. \tag{1.5}$$

Solution. We form the matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \tag{S.1.36}$$

The determinant of $\mathbf{A}$ is $\det \mathbf{A} = a_{11}a_{22} - a_{12}a_{21}$. By multiplying out $(\mathbf{a}_2^\top \mathbf{a}_2)$, $(\mathbf{a}_1^\top \mathbf{a}_1)$ and $(\mathbf{a}_1^\top \mathbf{a}_2)^2$, we get

$$\mathbf{a}_2^\top \mathbf{a}_2 = a_{12}^2 + a_{22}^2 \tag{S.1.37}$$

$$\mathbf{a}_1^\top \mathbf{a}_1 = a_{11}^2 + a_{21}^2 \tag{S.1.38}$$

$$(\mathbf{a}_1^\top \mathbf{a}_2)^2 = (a_{11}a_{12} + a_{21}a_{22})^2 = a_{11}^2 a_{12}^2 + a_{21}^2 a_{22}^2 + 2a_{11}a_{12}a_{21}a_{22}. \tag{S.1.39}$$

Therefore the area equals

$$S^2 = (a_{12}^2 + a_{22}^2)(a_{11}^2 + a_{21}^2) - (\mathbf{a}_1^\top \mathbf{a}_2)^2 \tag{S.1.40}$$

$$= a_{12}^2 a_{11}^2 + a_{12}^2 a_{21}^2 + a_{22}^2 a_{11}^2 + a_{22}^2 a_{21}^2 - (a_{12}^2 a_{11}^2 + a_{21}^2 a_{22}^2 + 2a_{11}a_{12}a_{21}a_{22}) \tag{S.1.41}$$

$$= a_{12}^2 a_{21}^2 + a_{22}^2 a_{11}^2 - 2a_{11}a_{12}a_{21}a_{22} \tag{S.1.42}$$

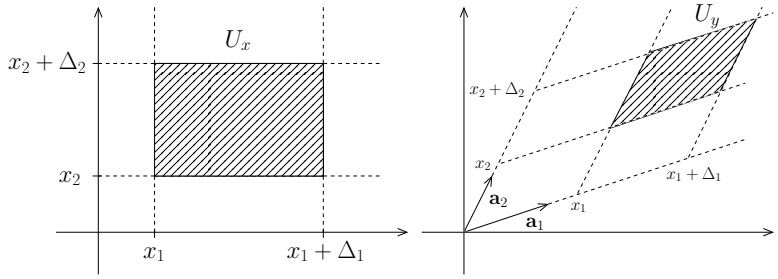
$$= (a_{11}a_{22} - a_{12}a_{21})^2, \tag{S.1.43}$$

which equals $(\det \mathbf{A})^2$.

(c) Consider the linear transform $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{A}$ is a 2×2 matrix. Denote the image of the rectangle $U_x = [x_1 \; x_1 + \triangle_1] \times [x_2 \; x_2 + \triangle_2]$ under the transform $\mathbf{A}$ by U_y. What is U_y? What is the area of U_y?

Solution. U_y is parallelogram that is spanned by the column vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ of $\mathbf{A}$, when $\mathbf{A} = (\mathbf{a}_1 \; \mathbf{a}_2)$.

A rectangle with the same area as U_x is spanned by vectors $(\Delta_1, 0)$ and $(0, \Delta_2)$. Under the linear transform $\mathbf{A}$ these spanning vectors become $\Delta_1\mathbf{a}_1$ and $\Delta_2\mathbf{a}_2$. Therefore a parallelogram with the same area as U_y is spanned by $\Delta_1\mathbf{a}_1$ and $\Delta_2\mathbf{a}_2$ as shown in the following figure.



From the previous question, the A_{U_y} of U_y equals the absolute value of the determinant of the matrix $(\Delta_1\mathbf{a}_1 \; \Delta_2\mathbf{a}_2)$:

$$A_{U_y} = |\det \begin{pmatrix} \Delta_1 a_{11} & \Delta_2 a_{12} \\ \Delta_1 a_{21} & \Delta_2 a_{22} \end{pmatrix}| \tag{S.1.44}$$

$$= |\Delta_1\Delta_2 a_{11}a_{22} - \Delta_1\Delta_2 a_{12}a_{21}| \tag{S.1.45}$$

$$= |\Delta_1\Delta_2(a_{11}a_{22} - a_{12}a_{21})| \tag{S.1.46}$$

$$= \Delta_1\Delta_2 |\det \mathbf{A}| \tag{S.1.47}$$

Therefore the area of U_y is the area of U_x times $|\det \mathbf{A}|$.

(d) Give an intuitive explanation why we have equality in the change of variables formula

$$\int_{U_y} f(\mathbf{y}) d\mathbf{y} = \int_{U_x} f(\mathbf{A}\mathbf{x}) |\det \mathbf{A}| d\mathbf{x}. \tag{1.6}$$

where $\mathbf{A}$ is such that U_x is an axis-aligned (hyper-) rectangle as in the previous question.

Solution. We can think that, loosely speaking, the two integrals are limits of the following two sums

$$\sum_{\mathbf{y}_i \in U_y} f(\mathbf{y}_i) \text{vol}(\Delta_{\mathbf{y}_i}) \qquad \sum_{\mathbf{x}_i \in U_x} f(\mathbf{A}\mathbf{x}_i) |\det \mathbf{A}| \text{vol}(\Delta_{\mathbf{x}_i}) \tag{S.1.48}$$

where $\mathbf{x}_i = \mathbf{A}^{-1}\mathbf{y}_i$, which means that $\mathbf{x}$ and $\mathbf{y}$ are related by $\mathbf{y} = \mathbf{A}\mathbf{x}$. The set of function values $f(\mathbf{y}_i)$ and $f(\mathbf{A}\mathbf{x}_i)$ that enter the two sums are exactly the same. The volume $\text{vol}(\Delta_{\mathbf{x}_i})$ of a small axis-aligned hypercube (in d dimensions) equals $\prod_{i=1}^{d} \Delta_i$. The image of this small axis-aligned hypercube under $\mathbf{A}$ is a parallelogram $\Delta_{\mathbf{y}_i}$ with volume $\text{vol}(\Delta_{\mathbf{y}_i}) = |\det \mathbf{A}| \text{vol}(\Delta_{\mathbf{x}_i})$. Hence

$$\sum_{\mathbf{y}_i \in U_y} f(\mathbf{y}_i) \text{vol}(\Delta_{\mathbf{y}_i}) = \sum_{\mathbf{x}_i \in U_x} f(\mathbf{A}\mathbf{x}_i) |\det \mathbf{A}| \text{vol}(\Delta_{\mathbf{x}_i}). \tag{S.1.49}$$

We must have the term $|\det \mathbf{A}|$ to compensate for the fact that the volume of U_x and U_y are not the same. For example, let $\mathbf{A}$ be a diagonal matrix $\text{diag}(10, 100)$ so that U_x is much smaller than U_y. The determinant $\det \mathbf{A} = 1000$ then compensates for the fact that the $\mathbf{x}_i$ values are more condensed than the $\mathbf{y}_i$.

1.3 Eigenvalue decomposition

For a square matrix $\mathbf{A}$ of size $n \times n$, a vector $\mathbf{u}_i \neq 0$ which satisfies

$$\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i \tag{1.7}$$

is called a eigenvector of $\mathbf{A}$, and λ_i is the corresponding eigenvalue. For a matrix of size $n \times n$, there are n eigenvalues λ_i (which are not necessarily distinct).

(a) Show that if $\mathbf{u}_1$ and $\mathbf{u}_2$ are eigenvectors with $\lambda_1 = \lambda_2$, then $\mathbf{u} = \alpha\mathbf{u}_1 + \beta\mathbf{u}_2$ is also an eigenvector with the same eigenvalue.

Solution. We compute

$$\begin{aligned} \mathbf{A}\mathbf{u} &= \alpha\mathbf{A}\mathbf{u}_1 + \beta\mathbf{A}\mathbf{u}_2 && \text{(S.1.50)} \\ &= \alpha\lambda\mathbf{u}_1 + \beta\lambda\mathbf{u}_2 && \text{(S.1.51)} \\ &= \lambda(\alpha\mathbf{u}_1 + \beta\mathbf{u}_2) && \text{(S.1.52)} \\ &= \lambda\mathbf{u}, && \text{(S.1.53)} \end{aligned}$$

so $\mathbf{u}$ is an eigenvector of $\mathbf{A}$ with the same eigenvalue as $\mathbf{u}_1$ and $\mathbf{u}_2$.

(b) Assume that none of the eigenvalues of $\mathbf{A}$ is zero. Denote by $\mathbf{U}$ the matrix where the column vectors are linearly independent eigenvectors $\mathbf{u}_i$ of $\mathbf{A}$. Verify that (1.7) can be written in matrix form as $\mathbf{AU} = \mathbf{U\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues λ_i as diagonal elements.

Solution. By basic properties of matrix multiplication, we have

$$\mathbf{AU} = (\mathbf{Au}_1\ \mathbf{Au}_2\ \ldots\ \mathbf{Au}_n) \tag{S.1.54}$$

With $\mathbf{Au}_i = \lambda_i \mathbf{u}_i$ for all $i = 1, 2, \ldots, n$, we thus obtain

$$\mathbf{AU} = (\lambda_1\mathbf{u}_1\ \lambda_2\mathbf{u}_2\ \ldots\ \lambda_n\mathbf{u}_n) \tag{S.1.55}$$

$$= \mathbf{U\Lambda}. \tag{S.1.56}$$

(c) Show that we can write, with $\mathbf{V}^T = \mathbf{U}^{-1}$,

$$\mathbf{A} = \mathbf{U\Lambda V}^\top, \qquad \mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{v}_i^T, \tag{1.8}$$

$$\mathbf{A}^{-1} = \mathbf{U\Lambda}^{-1}\mathbf{V}^\top, \qquad \mathbf{A}^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top, \tag{1.9}$$

where $\mathbf{v}_i$ is the i-th column of $\mathbf{V}$.

Solution.

(i) Since the columns of $\mathbf{U}$ are linearly independent, $\mathbf{U}$ is invertible. Because $\mathbf{AU} = \mathbf{U\Lambda}$, multiplying from the right with the inverse of $\mathbf{U}$ gives $\mathbf{A} = \mathbf{U\Lambda U}^{-1} = \mathbf{U\Lambda V}^\top$.

(ii) Denote by $\mathbf{u}^{[i]}$ the ith row of $\mathbf{U}$, $\mathbf{v}^{(j)}$ the jth column of $\mathbf{V}^\top$ and $\mathbf{v}^{[j]}$ the jth row of $\mathbf{V}$ and denote $\mathbf{B} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{v}_i^\top$. Let $\mathbf{e}^{[i]}$ be a row vector with 1 in the ith place and 0 elsewhere and $\mathbf{e}^{(j)}$ be a column vector with 1 in the jth place and 0 elsewhere. Notice that because $\mathbf{A} = \mathbf{U\Lambda V}^\top$, the element in the ith row and jth column is

$$A_{ij} = \mathbf{u}^{[i]}\mathbf{\Lambda}\mathbf{v}^{(j)} \tag{S.1.57}$$

$$= \mathbf{u}^{[i]}\mathbf{\Lambda}{\mathbf{v}^{[j]}}^\top \tag{S.1.58}$$

$$= \mathbf{u}^{[i]} \begin{pmatrix} \lambda_1 V_{j1} \\ \vdots \\ \lambda_n V_{jn} \end{pmatrix} \tag{S.1.59}$$

$$= \sum_{k=1}^{n} \lambda_k V_{jk} U_{ik}. \tag{S.1.60}$$

On the other hand, for matrix $\mathbf{B}$ the element in the ith row and jth column is

$$B_{ij} = \sum_{k=1}^{n} \lambda_k \mathbf{e}^{[i]} \mathbf{u}_k \mathbf{v}_k^\top \mathbf{e}^{(j)} \tag{S.1.61}$$

$$= \sum_{k=1}^{n} \lambda_k U_{ik} V_{jk}, \tag{S.1.62}$$

which is the same as A_{ij}. Therefore $\mathbf{A} = \mathbf{B}$.

(iii) Since $\mathbf{\Lambda}$ is a diagonal matrix with no zeros as diagonal elements, it is invertible. We have thus

$$\begin{aligned}\mathbf{A}^{-1} &= (\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1})^{-1} && \text{(S.1.63)}\\ &= (\mathbf{\Lambda}\mathbf{U}^{-1})^{-1}\mathbf{U}^{-1} && \text{(S.1.64)}\\ &= \mathbf{U}\Lambda^{-1}\mathbf{U}^{-1} && \text{(S.1.65)}\\ &= \mathbf{U}\Lambda^{-1}\mathbf{V}^{\top}. && \text{(S.1.66)}\end{aligned}$$

(iv) This follows from $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\top} = \sum_i \mathbf{u}_i\lambda_i\mathbf{v}_i^{\top}$, when λ_i is replaced with $1/\lambda_i$.

1.4 Trace, determinants and eigenvalues

(a) Use Exercise 1.3 to show that $\mathrm{tr}(\mathbf{A}) = \sum_i A_{ii} = \sum_i \lambda_i$. (You can use $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$.)

Solution. Since $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA}$ and $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$

$$\begin{aligned}\mathrm{tr}(\mathbf{A}) &= \mathrm{tr}(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}) && \text{(S.1.67)}\\ &= \mathrm{tr}(\mathbf{\Lambda}\mathbf{U}^{-1}\mathbf{U}) && \text{(S.1.68)}\\ &= \mathrm{tr}(\Lambda) && \text{(S.1.69)}\\ &= \sum_i \lambda_i. && \text{(S.1.70)}\end{aligned}$$

(b) Use Exercise 1.3 to show that $\det \mathbf{A} = \prod_i \lambda_i$. (Use $\det \mathbf{A}^{-1} = 1/(\det \mathbf{A})$ and $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$ for any $\mathbf{A}$ and $\mathbf{B}$.)

Solution. We use the eigenvalue decomposition of $\mathbf{A}$ to obtain

$$\begin{aligned}\det(\mathbf{A}) &= \det(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}) && \text{(S.1.71)}\\ &= \det(\mathbf{U})\det(\mathbf{\Lambda})\det(\mathbf{U}^{-1}) && \text{(S.1.72)}\\ &= \frac{\det(\mathbf{U})\det(\mathbf{\Lambda})}{\det(\mathbf{U})} && \text{(S.1.73)}\\ &= \det(\Lambda) && \text{(S.1.74)}\\ &= \prod_i \lambda_i, && \text{(S.1.75)}\end{aligned}$$

where, in the last line, we have used that the determinant of a diagonal matrix is the product of its elements.

1.5 Eigenvalue decomposition for symmetric matrices

(a) Assume that a matrix $\mathbf{A}$ is symmetric, i.e. $\mathbf{A}^{\top} = \mathbf{A}$. Let $\mathbf{u}_1$ and $\mathbf{u}_2$ be two eigenvectors of $\mathbf{A}$ with corresponding eigenvalues λ_1 and λ_2, with $\lambda_1 \neq \lambda_2$. Show that the two vectors are orthogonal to each other.

Solution. Since $\mathbf{A}\mathbf{u}_2 = \lambda_2\mathbf{u}_2$, we have

$$\mathbf{u}_1^\top \mathbf{A}\mathbf{u}_2 = \lambda_2 \mathbf{u}_1^\top \mathbf{u}_2. \tag{S.1.76}$$

Taking the transpose of $\mathbf{u}_1^\top \mathbf{A}\mathbf{u}_2$ gives

$$(\mathbf{u}_1^\top \mathbf{A}\mathbf{u}_2)^\top = (\mathbf{A}\mathbf{u}_2)^\top (\mathbf{u}_1^\top)^\top = \mathbf{u}_2^\top \mathbf{A}^\top \mathbf{u}_1 = \mathbf{u}_2^\top \mathbf{A}\mathbf{u}_1 \tag{S.1.77}$$

$$= \lambda_1 \mathbf{u}_2^\top \mathbf{u}_1 \tag{S.1.78}$$

because $\mathbf{A}$ is symmetric and $\mathbf{A}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$. On the other hand, the same operation gives

$$(\mathbf{u}_1^\top \mathbf{A}\mathbf{u}_2)^\top = (\lambda_2 \mathbf{u}_1^\top \mathbf{u}_2)^\top = \lambda_2 \mathbf{u}_2^\top \mathbf{u}_1 \tag{S.1.79}$$

Therefore $\lambda_1 \mathbf{u}_2^\top \mathbf{u}_1 = \lambda_2 \mathbf{u}_2^\top \mathbf{u}_1$, which is equivalent to $\mathbf{u}_2^\top \mathbf{u}_1(\lambda_1 - \lambda_2) = 0$. Because $\lambda_1 \neq \lambda_2$, the only possibility is that $\mathbf{u}_2^\top \mathbf{u}_1 = 0$. Therefore $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthogonal to each other.

The result implies that the eigenvectors of a symmetric matrix $\mathbf{A}$ with distinct eigenvalues λ_i forms an orthogonal basis. The result extends to the case where some of the eigenvalues are the same (not proven).

(b) A symmetric matrix $\mathbf{A}$ is said to be positive definite if $\mathbf{v}^T\mathbf{A}\mathbf{v} > 0$ for all non-zero vectors $\mathbf{v}$. Show that positive definiteness implies that $\lambda_i > 0$, $i = 1, \ldots, M$. Show that, vice versa, $\lambda_i > 0$, $i = 1 \ldots M$ implies that the matrix $\mathbf{A}$ is positive definite. Conclude that a positive definite matrix is invertible.

Solution. Assume that $\mathbf{v}^\top \mathbf{A}\mathbf{v} > 0$ for all $\mathbf{v} \neq 0$. Since eigenvectors are not zero vectors, the assumption holds also for eigenvector $\mathbf{u}_k$ with corresponding eigenvalue λ_k. Now

$$\mathbf{u}_k^\top \mathbf{A}\mathbf{u}_k = \mathbf{u}_k^\top \lambda_k \mathbf{u}_k = \lambda_k(\mathbf{u}_k^\top \mathbf{u}_k) = \lambda_k ||\mathbf{u}_k|| > 0 \tag{S.1.80}$$

and because $||\mathbf{u}_k|| > 0$, we obtain $\lambda_k > 0$.

Assume now that all the eigenvalues of $\mathbf{A}$, $\lambda_1, \lambda_2, \ldots, \lambda_n$, are positive and nonzero. We have shown above that there exists an orthogonal basis consisting of eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ and therefore every vector $\mathbf{v}$ can be written as a linear combination of those vectors (we have only shown it for the case of distinct eigenvalues but it holds more generally). Hence for a nonzero vector $\mathbf{v}$ and for some real numbers $\alpha_1, \alpha_2, \ldots, \alpha_n$, we have

$$\mathbf{v}^\top \mathbf{A}\mathbf{v} = (\alpha_1\mathbf{u}_1 + + \ldots + \alpha_n\mathbf{u}_n)^\top \mathbf{A}(\alpha_1\mathbf{u}_1 + \ldots + \alpha_n\mathbf{u}_n) \tag{S.1.81}$$

$$= (\alpha_1\mathbf{u}_1 + \ldots + \alpha_n\mathbf{u}_n)^\top (\alpha_1\mathbf{A}\mathbf{u}_1 + \ldots + \alpha_n\mathbf{A}\mathbf{u}_n) \tag{S.1.82}$$

$$= (\alpha_1\mathbf{u}_1 + \ldots + \alpha_n\mathbf{u}_n)^\top (\alpha_1\lambda_1\mathbf{u}_1 + \ldots + \alpha_n\lambda_n\mathbf{u}_n) \tag{S.1.83}$$

$$= \sum_{i,j} \alpha_i \mathbf{u}_i^\top \alpha_j \lambda_j \mathbf{u}_j \tag{S.1.84}$$

$$= \sum_i \alpha_i \alpha_i \lambda_i \mathbf{u}_i^\top \mathbf{u}_i \tag{S.1.85}$$

$$= \sum_i (\alpha_i)^2 ||\mathbf{u}_i||^2 \lambda_i, \tag{S.1.86}$$

where we have used that $\mathbf{u}_i^T \mathbf{u}_j = 0$ if $i \neq j$, due to orthogonality of the basis. Since $(\alpha_i)^2 > 0$, $||\mathbf{u}_i||^2 > 0$ and $\lambda_i > 0$ for all i, we find that $\mathbf{v}^\top \mathbf{A}\mathbf{v} > 0$.

Since every eigenvalue of $\mathbf{A}$ is nonzero, we can use Exercise 1.3 to conclude that inverse of $\mathbf{A}$ exists and equals $\sum_i 1/\lambda_i \mathbf{u}_i \mathbf{u}_i^\top$.

1.6 Power method

We here analyse an algorithm called the "power method". The power method takes as input a positive definite symmetric matrix $\mathbf{\Sigma}$ and calculates the eigenvector that has the largest eigenvalue (the "first eigenvector"). For example, in case of principal component analysis, $\mathbf{\Sigma}$ is the covariance matrix of the observed data and the first eigenvector is the first principal component direction.

The power method consists in iterating the update equations

$$\mathbf{v}_{k+1} = \mathbf{\Sigma}\mathbf{w}_k, \qquad \mathbf{w}_{k+1} = \frac{\mathbf{v}_{k+1}}{||\mathbf{v}_{k+1}||_2}, \tag{1.10}$$

where $||\mathbf{v}_{k+1}||_2$ denotes the Euclidean norm.

(a) Let $\mathbf{U}$ the matrix with the (orthonormal) eigenvectors $\mathbf{u}_i$ of $\mathbf{\Sigma}$ as columns. What is the eigenvalue decomposition of the covariance matrix $\mathbf{\Sigma}$?

Solution. Since the columns of $\mathbf{U}$ are orthonormal (eigen)vectors, $\mathbf{U}$ is orthogonal, i.e. $\mathbf{U}^{-1} = \mathbf{U}^\top$. With Exercise 1.3 and Exercise 1.5, we obtain

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{S.1.87}$$

where $\mathbf{\Lambda}$ is the diagonal matrix with eigenvalues λ_i of $\mathbf{\Sigma}$ as diagonal elements. Let the eigenvalues be ordered $\lambda_1 > \lambda_2 > \ldots > \lambda_n > 0$ (and, as additional assumption, all distinct).

(b) Let $\tilde{\mathbf{v}}_k = \mathbf{U}^T \mathbf{v}_k$ and $\tilde{\mathbf{w}}_k = \mathbf{U}^T \mathbf{w}_k$. Write the update equations of the power method in terms of $\tilde{\mathbf{v}}_k$ and $\tilde{\mathbf{w}}_k$. This means that we are making a change of basis to represent the vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ in the basis given by the eigenvectors of $\mathbf{\Sigma}$.

Solution. With

$$\mathbf{v}_{k+1} = \mathbf{\Sigma}\mathbf{w}_k \tag{S.1.88}$$
$$= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \mathbf{w}_k \tag{S.1.89}$$

we obtain

$$\mathbf{U}^\top \mathbf{v}_{k+1} = \mathbf{\Lambda}\mathbf{U}^\top \mathbf{w}_k. \tag{S.1.90}$$

Hence $\tilde{\mathbf{v}}_{k+1} = \mathbf{\Lambda}\tilde{\mathbf{w}}_k$. The norm of $\tilde{\mathbf{v}}_{k+1}$ is the same as the norm of $\mathbf{v}_{k+1}$:

$$
\begin{aligned}
||\tilde{\mathbf{v}}_{k+1}||_2 &= ||\mathbf{U}^\top \mathbf{v}_{k+1}||_2 && \text{(S.1.91)}\\
&= \sqrt{(\mathbf{U}^\top \mathbf{v}_{k+1})^\top (\mathbf{U}^\top \mathbf{v}_{k+1})} && \text{(S.1.92)}\\
&= \sqrt{\mathbf{v}_{k+1}^\top \mathbf{U}\mathbf{U}^\top \mathbf{v}_{k+1}} && \text{(S.1.93)}\\
&= \sqrt{\mathbf{v}_{k+1}^\top \mathbf{v}_{k+1}} && \text{(S.1.94)}\\
&= ||\mathbf{v}_{k+1}||_2. && \text{(S.1.95)}
\end{aligned}
$$

Hence, the update equation, in terms of $\tilde{\mathbf{v}}_k$ and $\tilde{\mathbf{w}}_k$, is

$$
\tilde{\mathbf{v}}_{k+1} = \mathbf{\Lambda}\tilde{\mathbf{w}}_k, \qquad \tilde{\mathbf{w}}_{k+1} = \frac{\tilde{\mathbf{v}}_{k+1}}{||\tilde{\mathbf{v}}_{k+1}||}. \tag{S.1.96}
$$

(c) Assume you start the iteration with $\tilde{\mathbf{w}}_0$. To which vector $\tilde{\mathbf{w}}^*$ does the iteration converge to?

Solution. Let $\tilde{\mathbf{w}}_0 = \begin{pmatrix}\alpha_1 & \alpha_2 & \dots & \alpha_n\end{pmatrix}^\top$. Since $\mathbf{\Lambda}$ is a diagonal matrix, we obtain

$$
\tilde{\mathbf{v}}_1 = \begin{pmatrix}\lambda_1\alpha_1\\ \lambda_2\alpha_2\\ \vdots\\ \lambda_n\alpha_n\end{pmatrix} = \lambda_1\alpha_1 \begin{pmatrix}1\\ \frac{\alpha_2}{\alpha_1}\frac{\lambda_2}{\lambda_1}\\ \vdots\\ \frac{\alpha_n}{\alpha_1}\frac{\lambda_n}{\lambda_1}\end{pmatrix} \tag{S.1.97}
$$

and therefore

$$
\tilde{\mathbf{w}}_1 = \frac{\lambda_1\alpha_1}{c_1} \begin{pmatrix}1\\ \frac{\alpha_2}{\alpha_1}\frac{\lambda_2}{\lambda_1}\\ \vdots\\ \frac{\alpha_n}{\alpha_1}\frac{\lambda_n}{\lambda_1}\end{pmatrix}, \tag{S.1.98}
$$

where c_1 is a normalisation constant such that $\|\tilde{\mathbf{w}}_1\| = 1$ (i.e. $c_1 = \|\tilde{\mathbf{v}}_1\|$). Hence, for $\tilde{\mathbf{w}}_k$ it holds that

$$
\tilde{\mathbf{w}}_k = \tilde{c}_k \begin{pmatrix}1\\ \frac{\alpha_2}{\alpha_1}\left(\frac{\lambda_2}{\lambda_1}\right)^k\\ \vdots\\ \frac{\alpha_n}{\alpha_1}\left(\frac{\lambda_n}{\lambda_1}\right)^k\end{pmatrix}, \tag{S.1.99}
$$

where $\tilde{c}_k$ is again a normalisation constant such that $||\tilde{\mathbf{w}}_k|| = 1$.

As λ_1 is the dominant eigenvalue, $|\lambda_j/\lambda_1| < 1$ for $j = 2, 3, \dots, n$, so that

$$
\lim_{k\to\infty}\left(\frac{\lambda_j}{\lambda_1}\right)^k = 0, \quad j = 2, 3, \dots, n, \tag{S.1.100}
$$

and hence

$$\lim_{k\to\infty} \begin{pmatrix} 1 \\ \frac{\alpha_2}{\alpha_1}\left(\frac{\lambda_2}{\lambda_1}\right)^k \\ \vdots \\ \frac{\alpha_n}{\alpha_1}\left(\frac{\lambda_n}{\lambda_1}\right)^k \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{S.1.101}$$

For the normalisation constant $\tilde{c}_k$, we obtain

$$\tilde{c}_k = \frac{1}{\sqrt{1+\sum_{i=2}^{n}\left(\frac{\alpha_i}{\alpha_1}\right)^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2k}}}, \tag{S.1.102}$$

and therefore

$$\lim_{k\to\infty} \tilde{c}_k = \frac{1}{\sqrt{1+\sum_{i=2}^{n}\left(\frac{\alpha_i}{\alpha_1}\right)^2 \lim\limits_{k\to\infty}\left(\frac{\lambda_i}{\lambda_1}\right)^{2k}}} \tag{S.1.103}$$

$$= \frac{1}{\sqrt{1+\sum_{i=2}^{n}\left(\frac{\alpha_i}{\alpha_1}\right)^2 \cdot 0}} \tag{S.1.104}$$

$$= 1. \tag{S.1.105}$$

The limit of the product of two convergent sequences is the product of the limits so that

$$\lim_{k\to\infty} \tilde{\mathbf{w}}_k = \lim_{k\to\infty} \tilde{c}_k \lim_{k\to\infty} \begin{pmatrix} 1 \\ \frac{\alpha_2}{\alpha_1}\left(\frac{\lambda_2}{\lambda_1}\right)^k \\ \vdots \\ \frac{\alpha_n}{\alpha_1}\left(\frac{\lambda_n}{\lambda_1}\right)^k \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{S.1.106}$$

(d) Conclude that the power method finds the first eigenvector.

Solution. Since $\mathbf{w}_k = \mathbf{U}\tilde{\mathbf{w}}_k$, we obtain

$$\lim_{k\to\infty} \mathbf{w}_k = \mathbf{U}\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{u}_1, \tag{S.1.107}$$

which is the eigenvector with the largest eigenvalue, i.e. the "first" or "dominant" eigenvector.

Chapter 2

Optimisation

Exercises

2.1 Gradient of vector-valued functions

For a function J that maps a column vector $\mathbf{w} \in \mathbb{R}^n$ to $\mathbb{R}$, the gradient is defined as

$$\nabla J(\mathbf{w}) = \begin{pmatrix} \frac{\partial J(\mathbf{w})}{\partial w_1} \\ \vdots \\ \frac{\partial J(\mathbf{w})}{\partial w_n} \end{pmatrix}, \tag{2.1}$$

where $\partial J(\mathbf{w})/\partial w_i$ are the partial derivatives of $J(\mathbf{w})$ with respect to the i-th element of the vector $\mathbf{w} = (w_1, \ldots, w_n)^\top$ (in the standard basis). Alternatively, it is defined to be the column vector $\nabla J(\mathbf{w})$ such that

$$J(\mathbf{w} + \epsilon \mathbf{h}) = J(\mathbf{w}) + \epsilon \left(\nabla J(\mathbf{w})\right)^\top \mathbf{h} + O(\epsilon^2) \tag{2.2}$$

for an arbitrary perturbation $\epsilon\mathbf{h}$. This phrases the derivative in terms of a first-order, or affine, approximation to the perturbed function $J(\mathbf{w} + \epsilon\mathbf{h})$. The derivative ∇J is a linear transformation that maps $\mathbf{h} \in \mathbb{R}^n$ to $\mathbb{R}$ (see e.g. Rudin, 1976, Chapter 9, for a formal treatment of derivatives).

Use either definition to determine $\nabla J(\mathbf{w})$ for the following functions where $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $f : \mathbb{R} \to \mathbb{R}$ is a differentiable function.

(a) $J(\mathbf{w}) = \mathbf{a}^\top \mathbf{w}$.

Solution. First method:

$$J(\mathbf{w}) = \mathbf{a}^\top \mathbf{w} = \sum_{k=1}^{n} a_k w_k \qquad \Longrightarrow \qquad \frac{\partial J(\mathbf{w})}{\partial w_i} = a_i \tag{S.2.1}$$

Hence

$$\nabla J(\mathbf{w}) = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \mathbf{a}. \tag{S.2.2}$$

Second method:

$$J(\mathbf{w} + \epsilon\mathbf{h}) = \mathbf{a}^\top(\mathbf{w} + \epsilon\mathbf{h}) = \underbrace{\mathbf{a}^\top \mathbf{w}}_{J(\mathbf{w})} + \epsilon \underbrace{\mathbf{a}^\top \mathbf{h}}_{\nabla J^\top \mathbf{h}} \tag{S.2.3}$$

Hence we find again $\nabla J(\mathbf{w}) = \mathbf{a}$.

(b) $J(\mathbf{w}) = \mathbf{w}^\top \mathbf{A} \mathbf{w}$.

Solution. First method: We start with

$$J(\mathbf{w}) = \mathbf{w}^\top \mathbf{A} \mathbf{w} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i A_{ij} w_j \tag{S.2.4}$$

Hence,

$$\frac{\partial J(\mathbf{w})}{\partial w_k} = \sum_{j=1}^{n} A_{kj} w_j + \sum_{i=1}^{n} w_i A_{ik} \tag{S.2.5}$$

$$= \sum_{j=1}^{n} A_{kj} w_j + \sum_{i=1}^{n} w_i (\mathbf{A}^\top)_{ki} \tag{S.2.6}$$

$$= \sum_{j=1}^{m} \left(A_{kj} + (\mathbf{A}^\top)_{kj} \right) w_j \tag{S.2.7}$$

where we have used that the entry in row i and column k of the matrix $\mathbf{A}$ equals the entry in row k and column i of its transpose $\mathbf{A}^\top$. It follows that

$$\nabla J(\mathbf{w}) = \begin{pmatrix} \sum_{j=1}^{n} \left(A_{1j} + (\mathbf{A}^\top)_{1j} \right) w_j \\ \vdots \\ \sum_{j=1}^{n} \left(A_{nj} + (\mathbf{A}^\top)_{nj} \right) w_j \end{pmatrix} \tag{S.2.8}$$

$$= (\mathbf{A} + \mathbf{A}^\top)\mathbf{w}, \tag{S.2.9}$$

where we have used that sums like $\sum_j B_{ij} w_j$ are equal to the i-th element of the matrix-vector product $\mathbf{Bw}$.

Second method:

$$J(\mathbf{w} + \epsilon\mathbf{h}) = (\mathbf{w} + \epsilon\mathbf{h})^\top \mathbf{A} (\mathbf{w} + \epsilon\mathbf{h}) \tag{S.2.10}$$

$$= \mathbf{w}^\top \mathbf{A}\mathbf{w} + \mathbf{w}^\top \mathbf{A}(\epsilon\mathbf{h}) + \epsilon\mathbf{h}^\top \mathbf{A}\mathbf{w} + \underbrace{\epsilon\mathbf{h}^\top \mathbf{A}\epsilon\mathbf{h}}_{O(\epsilon^2)} \tag{S.2.11}$$

$$= \mathbf{w}^\top \mathbf{A}\mathbf{w} + \epsilon(\mathbf{w}^\top \mathbf{A}\mathbf{h} + \mathbf{w}^\top \mathbf{A}^\top \mathbf{h}) + O(\epsilon^2) \tag{S.2.12}$$

$$= \underbrace{\mathbf{w}^\top \mathbf{A}\mathbf{w}}_{J(\mathbf{w})} + \epsilon(\underbrace{\mathbf{w}^\top \mathbf{A} + \mathbf{w}^\top \mathbf{A}^\top}_{\nabla J(\mathbf{w})^\top})\mathbf{h} + O(\epsilon^2) \tag{S.2.13}$$

where we have used that $\mathbf{h}^\top \mathbf{A}\mathbf{w}$ is a scalar so that $\mathbf{h}^\top \mathbf{A}\mathbf{w} = (\mathbf{h}^\top \mathbf{A}\mathbf{w})^\top = \mathbf{w}^\top \mathbf{A}^\top \mathbf{h}$. Hence

$$\nabla J(\mathbf{w})^\top = \mathbf{w}^\top \mathbf{A} + \mathbf{w}^\top \mathbf{A}^\top = \mathbf{w}^\top (\mathbf{A} + \mathbf{A}^\top) \tag{S.2.14}$$

and

$$\nabla J(\mathbf{w}) = (\mathbf{A} + \mathbf{A}^\top)\mathbf{w}. \tag{S.2.15}$$

(c) $J(\mathbf{w}) = \mathbf{w}^\top \mathbf{w}$.

Solution. The easiest way to calculate the gradient of $J(\mathbf{w}) = \mathbf{w}^\top \mathbf{w}$ is to use the previous question with $\mathbf{A} = \mathbf{I}$ (the identity matrix). Therefore

$$\nabla J(\mathbf{w}) = \mathbf{I}\mathbf{w} + \mathbf{I}^\top \mathbf{w} = \mathbf{w} + \mathbf{w} = 2\mathbf{w}. \tag{S.2.16}$$

(d) $J(\mathbf{w}) = ||\mathbf{w}||_2$.

Solution. Note that $||\mathbf{w}||_2 = \sqrt{\mathbf{w}^\top \mathbf{w}}$.

First method: We use the chain rule

$$\frac{\partial J(\mathbf{w})}{\partial w_k} = \frac{\partial \sqrt{\mathbf{w}^\top \mathbf{w}}}{\partial \mathbf{w}^\top \mathbf{w}} \frac{\partial \mathbf{w}^\top \mathbf{w}}{\partial w_k} \tag{S.2.17}$$

and that

$$\frac{\partial \sqrt{\mathbf{w}^\top \mathbf{w}}}{\partial \mathbf{w}^\top \mathbf{w}} = \frac{1}{2\sqrt{\mathbf{w}^\top \mathbf{w}}} \tag{S.2.18}$$

The derivatives $\partial \mathbf{w}^\top \mathbf{w} / \partial w_k$ were calculated in the question above so that

$$\nabla J(\mathbf{w}) = \frac{1}{2\sqrt{\mathbf{w}^\top \mathbf{w}}} 2\mathbf{w} = \frac{\mathbf{w}}{||\mathbf{w}||_2} \tag{S.2.19}$$

Second method: Let $f(\mathbf{w}) = \mathbf{w}^\top \mathbf{w}$. From the previous question, we know that

$$f(\mathbf{w} + \epsilon \mathbf{h}) = f(\mathbf{w}) + \epsilon 2\mathbf{w}^\top \mathbf{h} + O(\epsilon^2). \tag{S.2.20}$$

Moreover,

$$\sqrt{z + \epsilon u + O(\epsilon^2)} = \sqrt{z} + \frac{1}{2\sqrt{z}}(\epsilon u + O(\epsilon^2)) + O(\epsilon^2) \tag{S.2.21}$$

$$= \sqrt{z} + \epsilon \frac{1}{2\sqrt{z}} u + O(\epsilon^2) \tag{S.2.22}$$

With $z = f(\mathbf{w})$ and $u = 2\mathbf{w}^\top \mathbf{h}$, we thus obtain

$$J(\mathbf{w} + \epsilon \mathbf{h}) = \sqrt{f(\mathbf{w} + \epsilon \mathbf{h})} \tag{S.2.23}$$

$$= \sqrt{f(\mathbf{w})} + \epsilon \frac{1}{2\sqrt{f(\mathbf{w})}} 2\mathbf{w}^\top \mathbf{h} + O(\epsilon^2) \tag{S.2.24}$$

$$= \sqrt{f(\mathbf{w})} + \epsilon \frac{\mathbf{w}^\top}{\sqrt{f(\mathbf{w})}} \mathbf{h} + O(\epsilon^2) \tag{S.2.25}$$

$$= J(\mathbf{w}) + \epsilon \frac{\mathbf{w}^\top}{\sqrt{||\mathbf{w}||_2}} \mathbf{h} + O(\epsilon^2) \tag{S.2.26}$$

so that

$$\nabla J(\mathbf{w}) = \frac{\mathbf{w}}{||\mathbf{w}||_2}. \tag{S.2.27}$$

(e) $J(\mathbf{w}) = f(||\mathbf{w}||_2)$.

Solution. Either the chain rule or the approach with the Taylor expansion can be used to deal with the outer function f. In any case:

$$\nabla J(\mathbf{w}) = f'(||\mathbf{w}||_2) \nabla ||\mathbf{w}||_2 = f'(||\mathbf{w}||_2) \frac{\mathbf{w}}{||\mathbf{w}||_2}, \tag{S.2.28}$$

where f' is the derivative of the function f.

(f) $J(\mathbf{w}) = f(\mathbf{w}^\top \mathbf{a})$.

Solution. We have seen that $\nabla_{\mathbf{w}} \mathbf{a}^\top \mathbf{w} = \mathbf{a}$. Using the chain rule then yields

$$\nabla J(\mathbf{w}) = f'(\mathbf{w}^\top \mathbf{a}) \nabla (\mathbf{w}^\top \mathbf{a}) \tag{S.2.29}$$

$$= f'(\mathbf{w}^\top \mathbf{a}) \mathbf{a} \tag{S.2.30}$$

2.2 Newton's method

Assume that in the neighbourhood of $\mathbf{w}_0$, a function $J(\mathbf{w})$ can be described by the quadratic approximation

$$f(\mathbf{w}) = c + \mathbf{g}^\top (\mathbf{w} - \mathbf{w}_0) + \frac{1}{2} (\mathbf{w} - \mathbf{w}_0)^\top \mathbf{H} (\mathbf{w} - \mathbf{w}_0), \tag{2.3}$$

where $c = J(\mathbf{w}_0)$, $\mathbf{g}$ is the gradient of J with respect to $\mathbf{w}$, and $\mathbf{H}$ a symmetric positive definite matrix (e.g. the Hessian matrix for $J(\mathbf{w})$ at $\mathbf{w}_0$ if positive definite).

(a) Use Exercise 2.1 to determine $\nabla f(\mathbf{w})$.

Solution. We first write f as

$$f(\mathbf{w}) = c + \mathbf{g}^\top (\mathbf{w} - \mathbf{w}_0) + \frac{1}{2} (\mathbf{w} - \mathbf{w}_0)^T \mathbf{H} (\mathbf{w} - \mathbf{w}_0) \tag{S.2.31}$$

$$= c - \mathbf{g}^\top \mathbf{w}_0 + \frac{1}{2} \mathbf{w}_0^\top \mathbf{H} \mathbf{w}_0 + \mathbf{g}^\top \mathbf{w} + \frac{1}{2} \mathbf{w}^\top \mathbf{H} \mathbf{w} - \frac{1}{2} \mathbf{w}_0^\top \mathbf{H} \mathbf{w} - \frac{1}{2} \mathbf{w}^\top \mathbf{H} \mathbf{w}_0 \tag{S.2.32}$$

Using now that $\mathbf{w}^\top \mathbf{H} \mathbf{w}_0$ is a scalar and that $\mathbf{H}$ is symmetric, we have

$$\mathbf{w}^\top \mathbf{H} \mathbf{w}_0 = (\mathbf{w}^\top \mathbf{H} \mathbf{w}_0)^\top = \mathbf{w}_0^\top \mathbf{H}^\top \mathbf{w} = \mathbf{w}_0^\top \mathbf{H} \mathbf{w} \tag{S.2.33}$$

and hence

$$f(\mathbf{w}) = \text{const} + (\mathbf{g}^\top - \mathbf{w}_0^\top \mathbf{H}) \mathbf{w} + \frac{1}{2} \mathbf{w}^\top \mathbf{H} \mathbf{w} \tag{S.2.34}$$

With the results from Exercise 2.1 and the fact that $\mathbf{H}$ is symmetric, we thus obtain

$$\nabla f(\mathbf{w}) = \mathbf{g} - \mathbf{H}^\top \mathbf{w}_0 + \frac{1}{2} (\mathbf{H}^\top \mathbf{w} + \mathbf{H} \mathbf{w}) \tag{S.2.35}$$

$$= \mathbf{g} - \mathbf{H} \mathbf{w}_0 + \mathbf{H} \mathbf{w} \tag{S.2.36}$$

The expansion of $f(\mathbf{w})$ due to the $\mathbf{w} - \mathbf{w}_0$ terms is a bit tedious. It is simpler to note that gradients define a linear approximation of the function. We can more efficiently deal with $\mathbf{w} - \mathbf{w}_0$ by changing the coordinates and determine the linear approximation of f as a function of $\mathbf{v} = \mathbf{w} - \mathbf{w}_0$, i.e. locally around the point $\mathbf{w}_0$. We then have

$$\tilde{f}(\mathbf{v}) = f(\mathbf{v} + \mathbf{w}_0) \tag{S.2.37}$$

$$= c + \mathbf{g}^\top \mathbf{v} + \frac{1}{2} \mathbf{v}^\top \mathbf{H} \mathbf{v} \tag{S.2.38}$$

With Exercise 2.1, the derivative is

$$\nabla_{\mathbf{v}} \tilde{f}(\mathbf{v}) = \mathbf{g} + \mathbf{H}\mathbf{v} \tag{S.2.39}$$

and the linear approximation becomes

$$\tilde{f}(\mathbf{v} + \epsilon\mathbf{h}) = c + \epsilon(\mathbf{g} + \mathbf{H}\mathbf{v})^\top \mathbf{h} + O(\epsilon^2) \tag{S.2.40}$$

The linear approximation for $\tilde{f}$ determines a linear approximation of f around $\mathbf{w}_0$, i.e.

$$f(\mathbf{w} + \epsilon\mathbf{h}) = \tilde{f}(\mathbf{w} - \mathbf{w}_0 + \epsilon\mathbf{h}) = c + \epsilon(\mathbf{g} + \mathbf{H}(\mathbf{w} - \mathbf{w}_0))^\top \mathbf{h} + O(\epsilon^2) \tag{S.2.41}$$

so that the derivative for f is

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \mathbf{g} + \mathbf{H}(\mathbf{w} - \mathbf{w}_0) = \mathbf{g} - \mathbf{H}\mathbf{w}_0 + \mathbf{H}\mathbf{w}, \tag{S.2.42}$$

which is the same result as before.

(b) A necessary condition for $\mathbf{w}$ being optimal (leading either to a maximum, minimum or a saddle point) is $\nabla f(\mathbf{w}) = 0$. Determine $\mathbf{w}^*$ such that $\nabla f(\mathbf{w})\big|_{\mathbf{w}=\mathbf{w}^*} = 0$. Provide arguments why $\mathbf{w}^*$ is a minimiser of $f(\mathbf{w})$.

Solution. We set the gradient to zero and solve for $\mathbf{w}$:

$$\mathbf{g} + \mathbf{H}(\mathbf{w} - \mathbf{w}_0) = 0 \quad \leftrightarrow \quad \mathbf{w} - \mathbf{w}_0 = -\mathbf{H}^{-1}\mathbf{g} \tag{S.2.43}$$

so that

$$\mathbf{w}^* = \mathbf{w}_0 - \mathbf{H}^{-1}\mathbf{g}. \tag{S.2.44}$$

As we assumed that $\mathbf{H}$ is positive definite, the inverse $\mathbf{H}$ exists (and is positive definite too).

Let us consider f as a function of $\mathbf{v}$ around $\mathbf{w}^*$, i.e. $\mathbf{w} = \mathbf{w}^* + \mathbf{v}$. With $\mathbf{w}^* + \mathbf{v} - \mathbf{w}_0 = -\mathbf{H}^{-1}\mathbf{g} + \mathbf{v}$, we have

$$f(\mathbf{w}^* + \mathbf{v}) = c + \mathbf{g}^\top(-\mathbf{H}^{-1}\mathbf{g} + \mathbf{v}) + \frac{1}{2}(-\mathbf{H}^{-1}\mathbf{g} + \mathbf{v})^\top \mathbf{H}(-\mathbf{H}^{-1}\mathbf{g} + \mathbf{v}) \tag{S.2.45}$$

Since $\mathbf{H}$ is positive definite, we have that $(-\mathbf{H}^{-1}\mathbf{g} + \mathbf{v})^\top \mathbf{H}(-\mathbf{H}^{-1}\mathbf{g} + \mathbf{v}) > 0$ for all $\mathbf{v}$. Hence, as we move away from $\mathbf{w}^*$, the function increases quadratically, so that $\mathbf{w}^*$ minimises $f(\mathbf{w})$.

(c) In terms of Newton's method to minimise $J(\mathbf{w})$, what do $\mathbf{w}_0$ and $\mathbf{w}^*$ stand for?

Solution. The equation

$$\mathbf{w}^* = \mathbf{w}_0 - \mathbf{H}^{-1}\mathbf{g}. \tag{S.2.46}$$

corresponds to one update step in Newton's method where $\mathbf{w}_0$ is the current value of $\mathbf{w}$ in the optimisation of $J(\mathbf{w})$ and $\mathbf{w}^*$ is the updated value. In practice rather than determining the inverse $\mathbf{H}^{-1}$, we solve

$$\mathbf{H}\mathbf{p} = \mathbf{g} \tag{S.2.47}$$

for $\mathbf{p}$ and then set $\mathbf{w}^* = \mathbf{w}_0 - \mathbf{p}$. The vector $\mathbf{p}$ is the search direction, and it is possible include a step-length α so that the update becomes $\mathbf{w}^* = \mathbf{w}_0 - \alpha\mathbf{p}$. The value of α may be set by hand or can be determined via line-search methods (see e.g. Nocedal and Wright, 1999).

2.3 Gradient of matrix-valued functions

For functions J that map a matrix $\mathbf{W} \in \mathbb{R}^{n \times m}$ to $\mathbb{R}$, the gradient is defined as

$$\nabla J(\mathbf{W}) = \begin{pmatrix} \frac{\partial J(\mathbf{W})}{\partial W_{11}} & \cdots & \frac{\partial J(\mathbf{W})}{\partial W_{1m}} \\ \vdots & \vdots & \vdots \\ \frac{\partial J(\mathbf{W})}{\partial W_{n1}} & \cdots & \frac{\partial J(\mathbf{W})}{\partial W_{nm}} \end{pmatrix}. \tag{2.4}$$

Alternatively, it is defined to be the matrix ∇J such that

$$J(\mathbf{W} + \epsilon \mathbf{H}) = J(\mathbf{w}) + \epsilon \operatorname{tr}(\nabla J^\top \mathbf{H}) + O(\epsilon^2) \tag{2.5}$$
$$= J(\mathbf{w}) + \epsilon \operatorname{tr}(\nabla J \mathbf{H}^\top) + O(\epsilon^2) \tag{2.6}$$

This definition is analogue to the one for vector-valued functions in (2.2). It phrases the derivative in terms of a linear approximation to the perturbed objective $J(\mathbf{W} + \epsilon \mathbf{H})$ and, more formally, $\operatorname{tr} \nabla J^\top$ is a linear transformation that maps $\mathbf{H} \in \mathbb{R}^{n \times m}$ to $\mathbb{R}$ (see e.g. Rudin, 1976, Chapter 9, for a formal treatment of derivatives).

Let $\mathbf{e}^{(i)}$ be *column* vector which is everywhere zero but in slot i where it is 1. Moreover let $\mathbf{e}^{[j]}$ be a *row* vector which is everywhere zero but in slot j where it is 1. The outer product $\mathbf{e}^{(i)}\mathbf{e}^{[j]}$ is then a matrix that is everywhere zero but in row i and column j where it is one. For $\mathbf{H} = \mathbf{e}^{(i)}\mathbf{e}^{[j]}$, we obtain

$$J(\mathbf{W} + \epsilon \mathbf{e}^{(i)}\mathbf{e}^{[j]}) = J(\mathbf{W}) + \epsilon \operatorname{tr}((\nabla J)^\top \mathbf{e}^{(i)}\mathbf{e}^{[j]}) + O(\epsilon^2) \tag{2.7}$$
$$= J(\mathbf{W}) + \epsilon \mathbf{e}^{[j]}(\nabla J)^\top \mathbf{e}^{(i)} + O(\epsilon^2) \tag{2.8}$$
$$= J(\mathbf{W}) + \epsilon \mathbf{e}^{[i]} \nabla J \mathbf{e}^{(j)} + O(\epsilon^2) \tag{2.9}$$

Note that $\mathbf{e}^{[i]} \nabla J \mathbf{e}^{(j)}$ picks the element of the matrix ∇J that is in row i and column j, i.e. $\mathbf{e}^{[i]} \nabla J \mathbf{e}^{(j)} = \partial J / \partial W_{ij}$.

Use either of the two definitions to find $\nabla J(\mathbf{W})$ for the functions below, where $\mathbf{u} \in \mathbb{R}^n, \mathbf{v} \in \mathbb{R}^m, \mathbf{A} \in \mathbb{R}^{n \times m}$, and $f : \mathbb{R} \to \mathbb{R}$ is differentiable.

(a) $J(\mathbf{W}) = \mathbf{u}^\top \mathbf{W} \mathbf{v}$.

Solution. First method: With $J(\mathbf{W}) = \sum_{i=1}^n \sum_{j=1}^m u_i W_{ij} v_j$ we have

$$\frac{\partial J(\mathbf{W})}{W_{kl}} = u_k v_l = (\mathbf{u}\mathbf{v}^\top)_{kl} \tag{S.2.48}$$

and hence

$$\nabla J(\mathbf{W}) = \mathbf{u}\mathbf{v}^\top \tag{S.2.49}$$

Second method:

$$J(\mathbf{W} + \epsilon \mathbf{H}) = \mathbf{u}^\top (\mathbf{W} + \epsilon \mathbf{H}) \mathbf{v} \tag{S.2.50}$$
$$= J(\mathbf{W}) + \epsilon \mathbf{u}^\top \mathbf{H} \mathbf{v} \tag{S.2.51}$$
$$= J(\mathbf{W}) + \epsilon \operatorname{tr}(\mathbf{u}^\top \mathbf{H} \mathbf{v}) \tag{S.2.52}$$
$$= J(\mathbf{W}) + \epsilon \operatorname{tr}(\mathbf{v}\mathbf{u}^\top \mathbf{H}) \tag{S.2.53}$$

Hence:

$$\nabla J(\mathbf{W}) = \mathbf{u}\mathbf{v}^\top \tag{S.2.54}$$

(b) $J(\mathbf{W}) = \mathbf{u}^\top(\mathbf{W} + \mathbf{A})\mathbf{v}$.

Solution. Expanding the objective function gives $J(\mathbf{W}) = \mathbf{u}^\top\mathbf{W}\mathbf{v} + \mathbf{u}^\top\mathbf{A}\mathbf{v}$. The second term does not depend on $\mathbf{W}$. With the previous question, the derivative thus is

$$\nabla J(\mathbf{W}) = \mathbf{u}\mathbf{v}^\top \tag{S.2.55}$$

(c) $J(\mathbf{W}) = \sum_n f(\mathbf{w}_n^\top\mathbf{v})$, where $\mathbf{w}_n^\top$ are the rows of the matrix $\mathbf{W}$.

Solution. First method:

$$\frac{\partial J(\mathbf{W})}{\partial W_{ij}} = \sum_{k=1}^{n} \frac{\partial}{\partial W_{ij}} f(\mathbf{w}_k^\top\mathbf{v}) \tag{S.2.56}$$

$$= f'(\mathbf{w}_i^\top\mathbf{v}) \frac{\partial}{\partial W_{ij}} \underbrace{\mathbf{w}_i^\top\mathbf{v}}_{\sum_{j=1}^{m} W_{ij}v_j} \tag{S.2.57}$$

$$= f'(\mathbf{w}_i^\top\mathbf{v})v_j \tag{S.2.58}$$

Hence

$$\nabla J(\mathbf{W}) = f'(\mathbf{W}\mathbf{v})\mathbf{v}^\top, \tag{S.2.59}$$

where f' operates element-wise on the vector $\mathbf{W}\mathbf{v}$.

Second method:

$$J(\mathbf{W}) = \sum_{k=1}^{n} f(\mathbf{w}_k^\top\mathbf{v}) \tag{S.2.60}$$

$$= \sum_{k=1}^{n} f(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v}), \tag{S.2.61}$$

where $\mathbf{e}^{[k]}$ is the unit row vector that is zero everywhere but for element k which equals one. We now perform a perturbation of $\mathbf{W}$ by $\epsilon\mathbf{H}$.

$$J(\mathbf{W} + \epsilon\mathbf{H}) = \sum_{k=1}^{n} f(\mathbf{e}^{[k]}(\mathbf{W} + \epsilon\mathbf{H})\mathbf{v}) \tag{S.2.62}$$

$$= \sum_{k=1}^{n} f(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v} + \epsilon\mathbf{e}^{[k]}\mathbf{H}\mathbf{v}) \tag{S.2.63}$$

$$= \sum_{k=1}^{n} (f(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v}) + \epsilon f'(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v})\mathbf{e}^{[k]}\mathbf{H}\mathbf{v} + O(\epsilon^2) \tag{S.2.64}$$

$$= J(\mathbf{W}) + \epsilon \left(\sum_{k=1}^{n} f'(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v})\mathbf{e}^{[k]} \right) \mathbf{H}\mathbf{v} + O(\epsilon^2) \tag{S.2.65}$$

The term $f'(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v})\mathbf{e}^{[k]}$ is a row vector that equals $(0,\dots,0,f'(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v}),0,\dots,0)$. Hence, we have

$$\sum_{k=1}^{n} f'(\mathbf{e}^{[k]}\mathbf{W}\mathbf{v})\mathbf{e}^{[k]}) = f'(\mathbf{W}\mathbf{v})^\top \tag{S.2.66}$$

where f' operates element-wise on the column vector $\mathbf{W}\mathbf{v}$. The perturbed objective function thus is

$$\begin{aligned} J(\mathbf{W}+\epsilon\mathbf{H}) &= J(\mathbf{W}) + \epsilon f'(\mathbf{W}\mathbf{v})^\top \mathbf{H}\mathbf{v} + O(\epsilon^2) && \text{(S.2.67)}\\ &= J(\mathbf{W}) + \epsilon \operatorname{tr}\left(f'(\mathbf{W}\mathbf{v})^\top \mathbf{H}\mathbf{v}\right) + O(\epsilon^2) && \text{(S.2.68)}\\ &= J(\mathbf{W}) + \epsilon \operatorname{tr}\left(\mathbf{v} f'(\mathbf{W}\mathbf{v})^\top \mathbf{H}\right) + O(\epsilon^2) && \text{(S.2.69)} \end{aligned}$$

Hence, the gradient is the transpose of $\mathbf{v}f'(\mathbf{W}\mathbf{v})^\top$, i.e.

$$\nabla J(\mathbf{W}) = f'(\mathbf{W}\mathbf{v})\mathbf{v}^\top \tag{S.2.70}$$

(d) $J(\mathbf{W}) = \mathbf{u}^\top \mathbf{W}^{-1}\mathbf{v}$ (*Hint:* $(\mathbf{W}+\epsilon\mathbf{H})^{-1} = \mathbf{W}^{-1} - \epsilon\mathbf{W}^{-1}\mathbf{H}\mathbf{W}^{-1} + O(\epsilon^2)$.)

Solution. We first verify the hint:

$$\begin{aligned} \left(\mathbf{W}^{-1} - \epsilon\mathbf{W}^{-1}\mathbf{H}\mathbf{W}^{-1} + O(\epsilon^2)\right)(\mathbf{W}+\epsilon\mathbf{H}) &= \mathbf{I} + \epsilon\mathbf{W}^{-1}\mathbf{H} - \epsilon\mathbf{W}^{-1}\mathbf{H} + O(\epsilon^2) && \text{(S.2.71)}\\ &= \mathbf{I} + O(\epsilon^2) && \text{(S.2.72)} \end{aligned}$$

Hence the identity holds up to terms smaller than ϵ^2, which is sufficient we do not care about terms of order ϵ^2 and smaller in the definition of the gradient in (2.5).

Let us thus make a first-order approximation of the perturbed objective $J(\mathbf{W}+\epsilon\mathbf{H})$:

$$\begin{aligned} J(\mathbf{W}+\epsilon\mathbf{H}) &= \mathbf{u}^\top(\mathbf{W}+\epsilon\mathbf{H}\mathbf{v})^{-1}\mathbf{v} && \text{(S.2.73)}\\ &\overset{\text{hint}}{=} \mathbf{u}^\top(\mathbf{W}^{-1} - \epsilon\mathbf{W}^{-1}\mathbf{H}\mathbf{W}^{-1} + O(\epsilon^2))\mathbf{v} && \text{(S.2.74)}\\ &= \mathbf{u}^\top\mathbf{W}^{-1}\mathbf{v} - \epsilon\mathbf{u}^\top\mathbf{W}^{-1}\mathbf{H}\mathbf{W}^{-1}\mathbf{v} + O(\epsilon^2) && \text{(S.2.75)}\\ &= J(\mathbf{W}) - \epsilon \operatorname{tr}\left(\mathbf{u}^\top\mathbf{W}^{-1}\mathbf{H}\mathbf{W}^{-1}\mathbf{v}\right) + O(\epsilon^2) && \text{(S.2.76)}\\ &= J(\mathbf{W}) - \epsilon \operatorname{tr}\left(\mathbf{W}^{-1}\mathbf{v}\mathbf{u}^\top\mathbf{W}^{-1}\mathbf{H}\right) + O(\epsilon^2) && \text{(S.2.77)} \end{aligned}$$

Comparison with (2.5) gives

$$\nabla J^\top = -\mathbf{W}^{-1}\mathbf{v}\mathbf{u}^\top\mathbf{W}^{-1} \tag{S.2.78}$$

and hence

$$\nabla J = -\mathbf{W}^{-\top}\mathbf{u}\mathbf{v}^\top\mathbf{W}^{-\top}, \tag{S.2.79}$$

where $\mathbf{W}^{-\top}$ is the transpose of the inverse of $\mathbf{W}$.

2.4 Gradient of the log-determinant

The goal of this exercise is to determine the gradient of

$$J(\mathbf{W}) = \log|\det(\mathbf{W})|. \tag{2.10}$$

(a) Show that the n-th eigenvalue λ_n can be written as

$$\lambda_n = \mathbf{v}_n^\top \mathbf{W}\mathbf{u}_n, \tag{2.11}$$

where $\mathbf{u}_n$ is the nth eigenvector and $\mathbf{v}_n$ the nth column vector of $\mathbf{U}^{-1}$, with $\mathbf{U}$ being the matrix with the eigenvectors $\mathbf{u}_n$ as columns.

Solution. As in Exercise 1.3, let $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$ be the eigenvalue decomposition of $\mathbf{W}$ (with $\mathbf{V}^\top = \mathbf{U}^{-1}$). Then $\mathbf{\Lambda} = \mathbf{V}^\top\mathbf{W}\mathbf{U}$ and

$$\begin{aligned}
\lambda_n &= \mathbf{e}^{[n]}\mathbf{\Lambda}\mathbf{e}^{(n)} && \text{(S.2.80)}\\
&= \mathbf{e}^{[n]}\mathbf{V}^\top\mathbf{W}\mathbf{U}\mathbf{e}^{(n)} && \text{(S.2.81)}\\
&= (\mathbf{V}\mathbf{e}^{(n)})^\top\mathbf{W}\mathbf{U}\mathbf{e}^{(n)} && \text{(S.2.82)}\\
&= \mathbf{v}_n^\top\mathbf{W}\mathbf{u}_n, && \text{(S.2.83)}
\end{aligned}$$

where $\mathbf{e}^{(n)}$ is the standard basis (unit) vector with a 1 in the n-th slot and zeros elsewhere, and $\mathbf{e}^{[n]}$ is the corresponding row vector.

(b) Calculate the gradient of λ_n with respect to $\mathbf{W}$, i.e. $\nabla\lambda_n(\mathbf{W})$.

Solution. With Exercise 2.3, we have

$$\nabla_{\mathbf{W}}\lambda_n(\mathbf{W}) = \nabla_{\mathbf{W}}\mathbf{v}_n^\top\mathbf{W}\mathbf{u}_n = \mathbf{v}_n\mathbf{u}_n^\top. \tag{S.2.84}$$

(c) Write $J(\mathbf{W})$ in terms of the eigenvalues λ_n and calculate $\nabla J(\mathbf{W})$.

Solution. In Exercise 1.4, we have shown that $\det(\mathbf{W}) = \prod_i \lambda_i$ and hence $|\det(W)| = \prod_i |\lambda_i|$.

(i) If $\mathbf{W}$ is positive definite, its eigenvalues are positive and we can drop the absolute values so that $|\det(W)| = \prod_i \lambda_i$.

(ii) If $\mathbf{W}$ is a matrix with real entries, then $\mathbf{W}\mathbf{u} = \lambda\mathbf{u}$ implies $\mathbf{W}\bar{\mathbf{u}} = \bar{\lambda}\bar{\mathbf{u}}$, i.e. if λ is a complex eigenvalue, then $\bar{\lambda}$ (the complex conjugate of λ) is also an eigenvalue. Since $|\lambda|^2 = \lambda\bar{\lambda}$,

$$|\det(\mathbf{W})| = \left(\prod_{\lambda_i\in\mathbb{C}} \lambda_i\right)\left(\prod_{\lambda_j\in\mathbb{R}} |\lambda_j|\right). \tag{S.2.85}$$

Now we can write $J(\mathbf{W})$ in terms of the eigenvalues:

$$J(\mathbf{W}) = \log|\det(\mathbf{W})| \tag{S.2.86}$$

$$= \log\left(\prod_{\lambda_i\in\mathbb{C}} \lambda_i\right)\left(\prod_{\lambda_j\in\mathbb{R}} |\lambda_j|\right) \tag{S.2.87}$$

$$= \log\left(\prod_{\lambda_i\in\mathbb{C}} \lambda_i\right) + \log\left(\prod_{\lambda_j\in\mathbb{R}} |\lambda_j|\right) \tag{S.2.88}$$

$$= \sum_{\lambda_i\in\mathbb{C}} \log\lambda_i + \sum_{\lambda_j\in\mathbb{R}} \log|\lambda_j|. \tag{S.2.89}$$

Assume that the real-valued λ_j are non-zero so that

$$\nabla_{\mathbf{W}} \log|\lambda_j| = \frac{1}{|\lambda_j|}\nabla_{\mathbf{W}}|\lambda_j| \tag{S.2.90}$$

$$= \frac{1}{|\lambda_j|}\text{sign}(\lambda_j)\nabla_{\mathbf{W}}\lambda_j \tag{S.2.91}$$

Hence

$$\nabla J(\mathbf{W}) = \sum_{\lambda_i\in\mathbb{C}} \nabla_{\mathbf{W}}\log\lambda_i + \sum_{\lambda_j\in\mathbb{R}} \nabla_{\mathbf{W}}\log|\lambda_j| \tag{S.2.92}$$

$$= \sum_{\lambda_i\in\mathbb{C}} \frac{1}{\lambda_i}\nabla_{\mathbf{W}}\lambda_i + \sum_{\lambda_i\in\mathbb{R}} \frac{1}{|\lambda_i|}\text{sign}(\lambda_i)\nabla_{\mathbf{W}}\lambda_i \tag{S.2.93}$$

$$= \sum_{\lambda_i\in\mathbb{C}} \frac{\mathbf{v}_i\mathbf{u}_i^\top}{\lambda_i} + \sum_{\lambda_i\in\mathbb{R}} \frac{\text{sign}(\lambda_i)\mathbf{v}_i\mathbf{u}_i^\top}{|\lambda_i|} \tag{S.2.94}$$

$$= \sum_{\lambda_i\in\mathbb{C}} \frac{\mathbf{v}_i\mathbf{u}_i^\top}{\lambda_i} + \sum_{\lambda_i\in\mathbb{R}} \frac{\mathbf{v}_i\mathbf{u}_i^\top}{\lambda_i} \tag{S.2.95}$$

$$= \sum_i \frac{\mathbf{v}_i\mathbf{u}_i^\top}{\lambda_i}. \tag{S.2.96}$$

(d) Show that

$$\nabla J(\mathbf{W}) = (\mathbf{W}^{-1})^\top. \tag{2.12}$$

Solution. This follows from Exercise 1.3 where we have found that

$$\mathbf{W}^{-1} = \sum_i \frac{1}{\lambda_i}\mathbf{u}_i\mathbf{v}_i^\top. \tag{S.2.97}$$

Indeed:

$$\nabla J(\mathbf{W}) = \sum_i \frac{\mathbf{v}_i\mathbf{u}_i^\top}{\lambda_i} = \sum_i \frac{1}{\lambda_i}(\mathbf{u}_i\mathbf{v}_i^\top)^\top = \left(\mathbf{W}^{-1}\right)^\top. \tag{S.2.98}$$

2.5 Descent directions for matrix-valued functions

Assume we would like to minimise a matrix valued function $J(\mathbf{W})$ by gradient descent, i.e. the update equation is

$$\mathbf{W} \leftarrow \mathbf{W} - \epsilon \nabla J(\mathbf{W}), \tag{2.13}$$

where ϵ is the step-length. The gradient $\nabla J(\mathbf{W})$ was defined in Exercise 2.3. It was there pointed out that the gradient defines a first order approximation to the perturbed objective function $J(\mathbf{W} + \epsilon \mathbf{H})$. With (2.5),

$$J(\mathbf{W} - \epsilon \nabla J(\mathbf{W})) = J(\mathbf{W}) - \epsilon \operatorname{tr}(\nabla J(\mathbf{W})^\top \nabla J(\mathbf{W})) + O(\epsilon^2) \tag{2.14}$$

For any (nonzero) matrix $\mathbf{M}$, it holds that

$$\begin{aligned}
\operatorname{tr}(\mathbf{M}^\top \mathbf{M}) &= \sum_i (\mathbf{M}^\top \mathbf{M})_{ii} && (2.15)\\
&= \sum_i \sum_j (\mathbf{M}^\top)_{ij} (\mathbf{M})_{ji} && (2.16)\\
&= \sum_i \sum_j M_{ji} M_{ji} && (2.17)\\
&= \sum_{ij} (M_{ji})^2 && (2.18)\\
&> 0, && (2.19)
\end{aligned}$$

which means that $\operatorname{tr}(\nabla J(\mathbf{W})^\top \nabla J(\mathbf{W})) > 0$ if the gradient is nonzero, and hence

$$J(\mathbf{W} - \epsilon \nabla J(\mathbf{W})) < J(\mathbf{W}) \tag{2.20}$$

for small enough ϵ. Consequently, $\nabla J(\mathbf{W})$ is a descent direction. Show that $\mathbf{A}^\top \mathbf{A} \nabla J(\mathbf{W}) \mathbf{B}\mathbf{B}^\top$ for non-zero matrices $\mathbf{A}$ and $\mathbf{B}$ is also a descent direction or leaves the leaves the objective invariant.

Solution. As in the introduction to the question, we appeal to (2.5) to obtain

$$\begin{aligned}
J(\mathbf{W} - \epsilon \nabla J(\mathbf{W}) \mathbf{A}^\top \mathbf{A} \mathbf{B}\mathbf{B}^\top) &= J(\mathbf{W}) - \epsilon \operatorname{tr}(\nabla J(\mathbf{W})^\top \mathbf{A}^\top \mathbf{A} \nabla J(\mathbf{W}) \mathbf{B}\mathbf{B}^\top) + O(\epsilon^2) && \text{(S.2.99)}\\
&= J(\mathbf{W}) - \epsilon \operatorname{tr}(\mathbf{B}^\top \nabla J(\mathbf{W})^\top \mathbf{A}^\top \mathbf{A} \nabla J(\mathbf{W}) \mathbf{B}) + O(\epsilon^2), && \text{(S.2.100)}
\end{aligned}$$

where $\operatorname{tr}(\mathbf{B}^\top \nabla J(\mathbf{W})^\top \mathbf{A}^\top \mathbf{A} \nabla J(\mathbf{W}) \mathbf{B})$ takes the form $\operatorname{tr}(\mathbf{M}^\top \mathbf{M})$ with $\mathbf{M} = \mathbf{A} \nabla J(\mathbf{W}) \mathbf{B}$. With (2.19), we thus have $\operatorname{tr}(\mathbf{B}^\top \nabla J(\mathbf{W})^\top \mathbf{A}^\top \mathbf{A} \nabla J(\mathbf{W}) \mathbf{B}) > 0$ if $\mathbf{A} \nabla J(\mathbf{W}) \mathbf{B}$ is non-zero, and hence

$$J(\mathbf{W} - \epsilon \mathbf{A}^\top \mathbf{A} \nabla J(\mathbf{W}) \mathbf{B}\mathbf{B}^\top) < J(\mathbf{W}) \tag{S.2.101}$$

for small enough ϵ. We have equality if $\mathbf{A} \nabla J(\mathbf{W}) \mathbf{B} = 0$, e.g. if the columns of $\mathbf{B}$ are all in the null space of ∇J.

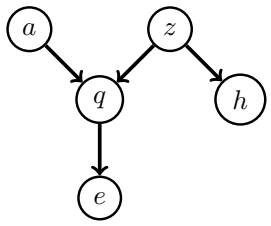
Chapter 3

Directed Graphical Models

Exercises

3.1 Directed graph concepts

Consider the following directed graph:



(a) List all trails in the graph (of maximal length)

Solution. We have

$$(a, q, e) \qquad (a, q, z, h) \qquad (h, z, q, e)$$

and the corresponding ones with swapped start and end nodes.

(b) List all directed paths in the graph (of maximal length)

Solution. (a, q, e) (z, q, e) (z, h)

(c) What are the descendants of z?

Solution. $\text{desc}(z) = \{q, e, h\}$

(d) What are the non-descendants of q?

Solution. $\text{nondesc}(q) = \{a, z, h, e\} \setminus \{e\} = \{a, z, h\}$

(e) Which of the following orderings are topological to the graph?

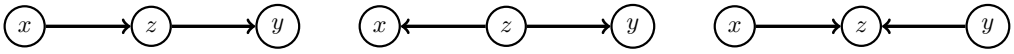
- (a,z,h,q,e)
- (a,z,e,h,q)
- (z,a,q,h,e)
- (z,q,e,a,h)

Solution.

- (a,z,h,q,e): yes
- (a,z,e,h,q): no (q is a parent of e and thus has to come before e in the ordering)
- (z,a,q,h,e): yes
- (z,q,e,a,h): no (a is a parent of q and thus has to come before q in the ordering)

3.2 Canonical connections

We here derive the independencies that hold in the three canonical connections that exist in DAGs, shown in Figure 3.1.



(a) Serial connection (b) Diverging connection (c) Converging connection

Figure 3.1: The three canonical connections in DAGs.

(a) For the serial connection, use the ordered Markov property to show that $x \perp\!\!\!\perp y \,|\, z$.

Solution. The only topological ordering is x, z, y. The predecessors of y are $\text{pre}_y = \{x, z\}$ and its parents $\text{pa}_y = \{z\}$. The ordered Markov property

$$y \perp\!\!\!\perp (\text{pre}_y \setminus \text{pa}_y) \,|\, \text{pa}_y \tag{S.3.1}$$

thus becomes $y \perp\!\!\!\perp (\{x, z\} \setminus z) \,|\, z$. Hence we have

$$y \perp\!\!\!\perp x \,|\, z, \tag{S.3.2}$$

which is the same as $x \perp\!\!\!\perp y \,|\, z$ since the independency relationship is symmetric.

This means that if the state or value of z is known (i.e. if the random variable z is "instantiated"), evidence about x will not change our belief about y, and vice versa. We say that the z node is "closed" and that the trail between x and y is "blocked" by the instantiated z. In other words, knowing the value of z blocks the flow of evidence *between* x and y.

(b) For the serial connection, show that the marginal $p(x, y)$ does generally not factorise into $p(x)p(y)$, i.e. that $x \perp\!\!\!\perp y$ does not hold.

Solution. There are several ways to show the result. One is to present an example where the independency does not hold. Consider for instance the following model

$$x \sim \mathcal{N}(x; 0, 1) \tag{S.3.3}$$
$$z = x + n_z \tag{S.3.4}$$
$$y = z + n_y \tag{S.3.5}$$

where $n_z \sim \mathcal{N}(n_z; 0, 1)$ and $n_y \sim \mathcal{N}(n_y; 0, 1)$, both being statistically independent from x. Here $\mathcal{N}(\cdot; 0, 1)$ denotes the Gaussian pdf with mean 0 and variance 1, and $x \sim \mathcal{N}(x; 0, 1)$ means that we sample x from the distribution $\mathcal{N}(x; 0, 1)$. Hence $p(z|x) = \mathcal{N}(z; x, 1)$, $p(y|z) = \mathcal{N}(y; z, 1)$ and $p(x, y, z) = p(x)p(z|x)p(y|z) = \mathcal{N}(x; 0, 1)\mathcal{N}(z; x, 1)\mathcal{N}(y; z, 1)$.

Whilst we could manipulate the pdfs to show the result, it's here easier to work with the generative model in Equations (S.3.3) to (S.3.5). Eliminating z from the equations, by plugging the definition of z into (S.3.5) we have

$$y = x + n_z + n_y, \tag{S.3.6}$$

which describes the marginal distribution of (x, y). We see that $\mathbb{E}[xy]$ is

$$\begin{aligned}\mathbb{E}[xy] &= \mathbb{E}[x^2 + xn_z + xn_y] && \text{(S.3.7)}\\ &= \mathbb{E}[x^2] + \mathbb{E}[x]\mathbb{E}[n_z] + \mathbb{E}[x]\mathbb{E}[n_y] && \text{(S.3.8)}\\ &= 1 + 0 + 0 && \text{(S.3.9)}\end{aligned}$$

where we have use the linearity of expectation, that x is independent from n_z and n_y, and that x has zero mean. If x and y were independent (or only uncorrelated), we had $\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y] = 0$. However, since $\mathbb{E}[xy] \neq \mathbb{E}[x]\mathbb{E}[y]$, x and y are not independent.

In plain English, this means that if the state of z is unknown, then evidence or information about x will influence our belief about y, and the other way around. Evidence can flow through z between x and y. We say that the z node is "open" and the trail between x and y is "active".

(c) For the diverging connection, use the ordered Markov property to show that $x \perp\!\!\!\perp y \mid z$.

Solution. A topological ordering is z, x, y. The predecessors of y are $\text{pre}_y = \{x, z\}$ and its parents $\text{pa}_y = \{z\}$. The ordered Markov property

$$y \perp\!\!\!\perp (\text{pre}_y \setminus \text{pa}_y) \mid \text{pa}_y \tag{S.3.10}$$

thus becomes again

$$y \perp\!\!\!\perp x \mid z, \tag{S.3.11}$$

which is, since the independence relationship is symmetric, the same as $x \perp\!\!\!\perp z \mid z$.

As in the serial connection, if the state or value z is known, evidence about x will not change our belief about y, and vice versa. Knowing z closes the z node, which blocks the trail between x and y.

(d) For the diverging connection, show that the marginal $p(x, y)$ does generally not factorise into $p(x)p(y)$, i.e. that $x \perp\!\!\!\perp y$ does not hold.

Solution. As for the serial connection, it suffices to give an example where $x \perp\!\!\!\perp y$ does not hold. We consider the following generative model

$$\begin{aligned}z &\sim \mathcal{N}(z; 0, 1) && \text{(S.3.12)}\\ x &= z + n_x && \text{(S.3.13)}\\ y &= z + n_y && \text{(S.3.14)}\end{aligned}$$

where $n_x \sim \mathcal{N}(n_x; 0, 1)$ and $n_y \sim \mathcal{N}(n_y; 0, 1)$, and they are independent of each other and the other variables. We have $\mathbb{E}[x] = \mathbb{E}[z + n_x] = \mathbb{E}[z] + \mathbb{E}[n_x] = 0$. On the other hand

$$\begin{aligned}\mathbb{E}[xy] &= \mathbb{E}[(z + n_x)(z + n_y)] && \text{(S.3.15)}\\ &= \mathbb{E}[z^2 + z(n_x + n_y) + n_x n_y] && \text{(S.3.16)}\\ &= \mathbb{E}[z^2] + \mathbb{E}[z(n_x + n_y)] + \mathbb{E}[n_x n_y] && \text{(S.3.17)}\\ &= 1 + 0 + 0 && \text{(S.3.18)}\end{aligned}$$

Hence, $\mathbb{E}[xy] \neq \mathbb{E}[x]\mathbb{E}[y]$ and we do not have that $x \perp\!\!\!\perp y$ holds.

In a diverging connection, as in the serial connection, if the state of z is unknown, then evidence or information about x will influence our belief about y, and the other way around. Evidence can flow through z between x and y. We say that the z node is open and the trail between x and y is active.

(e) For the converging connection, show that $x \perp\!\!\!\perp y$.

Solution. We can here again use the ordered Markov property with the ordering y, x, z. Since $\text{pre}_x = \{y\}$ and $\text{pa}_x = \varnothing$, we have

$$x \perp\!\!\!\perp (\text{pre}_x \setminus \text{pa}_x) \mid \text{pa}_x = x \perp\!\!\!\perp y. \tag{S.3.19}$$

Alternatively, we can use the basic definition of directed graphical models, i.e.

$$p(x, y, z) = k(x)k(y)k(z \mid x, y) \tag{S.3.20}$$

together with the result that the kernels (factors) are valid (conditional) pdfs/pmfs and equal to the conditionals/marginals with respect to the joint distribution $p(x, y, z)$, i.e.

$$k(x) = p(x) \tag{S.3.21}$$
$$k(y) = p(y) \tag{S.3.22}$$
$$k(z|x, y) = p(z|x, y) \quad \text{(not needed in the proof below)} \tag{S.3.23}$$

Integrating out z gives

$$p(x, y) = \int p(x, y, z)\mathrm{d}z \tag{S.3.24}$$
$$= \int k(x)k(y)k(z \mid x, y)\mathrm{d}z \tag{S.3.25}$$
$$= k(x)k(y) \underbrace{\int k(z \mid x, y)\mathrm{d}z}_{1} \tag{S.3.26}$$
$$= p(x)p(y) \tag{S.3.27}$$

Hence $p(x, y)$ factorises into its marginals, which means that $x \perp\!\!\!\perp y$.

Hence, when we do not have evidence about z, evidence about x will not change our belief about y, and vice versa. For the converging connection, if no evidence about z is available, the z node is closed, which blocks the trail between x and y.

(f) For the converging connection, show that $x \perp\!\!\!\perp y \mid z$ does generally not hold.

Solution. We give a simple example where $x \perp\!\!\!\perp y \mid z$ does not hold.

Consider

$$x \sim \mathcal{N}(x; 0, 1) \tag{S.3.28}$$
$$y \sim \mathcal{N}(y; 0, 1) \tag{S.3.29}$$
$$z = xy + n_z \tag{S.3.30}$$

where $n_z \sim \mathcal{N}(n_z; 0, 1)$, independent from the other variables. From the last equation, we have

$$xy = z - n_z \tag{S.3.31}$$

We thus have

$$\mathbb{E}[xy \mid z] = \mathbb{E}[z - n_z \mid z] \tag{S.3.32}$$
$$= z - 0 \tag{S.3.33}$$

On the other hand, $\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y] = 0$. Since $\mathbb{E}[xy \mid z] \neq \mathbb{E}[xy]$, $x \perp\!\!\!\perp y \mid z$ cannot hold.

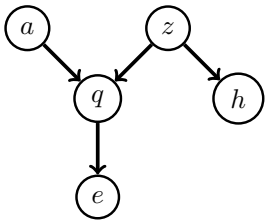
The intuition here is that if you know the value of the product xy, even if subject to noise, knowing the value of x allows you to guess the value of y and vice versa.

More generally, for converging connections, if evidence or information about z is available, evidence about x will influence the belief about y, and vice versa. We say that information about z opens the z-node, and evidence can flow between x and y.

Note: information about z means that z *or one of its descendents* is observed, see exercise 3.9.

3.3 Ordered and local Markov properties, d-separation

We continue with the investigation of the graph from Exercise 3.1 shown below for reference.



(a) The ordering (z, h, a, q, e) is topological to the graph. What are the independencies that follow from the ordered Markov property?

Solution. A distribution that factorises over the graph satisfies the independencies

$$x_i \perp\!\!\!\perp (\mathrm{pre}_i \setminus \mathrm{pa}_i) \mid \mathrm{pa}_i \text{ for all } i$$

for all orderings of the variables that are topological to the graph. The ordering comes into play via the predecessors $\mathrm{pre}_i = \{x_1, \ldots, x_{i-1}\}$ of the variables x_i; the graph via the parent sets pa_i.

For the graph and the specified topological ordering, the predecessor sets are

$$\mathrm{pre}_z = \varnothing, \mathrm{pre}_h = \{z\}, \mathrm{pre}_a = \{z, h\}, \mathrm{pre}_q = \{z, h, a\}, \mathrm{pre}_e = \{z, h, a, q\}$$

The parent sets only depend on the graph and not the topological ordering. They are:

$$\mathrm{pa}_z = \varnothing, \mathrm{pa}_h = \{z\}, \mathrm{pa}_a = \varnothing, \mathrm{pa}_q = \{a, z\}, \mathrm{pa}_e = \{q\},$$

The ordered Markov property reads $x_i \perp\!\!\!\perp (\text{pre}_i \setminus \text{pa}_i) \mid \text{pa}_i$ where the x_i refer to the ordered variables, e.g. $x_1 = z, x_2 = h, x_3 = a$, etc.

With

$$\text{pre}_h \setminus \text{pa}_h = \varnothing \quad \text{pre}_a \setminus \text{pa}_a = \{z, h\} \quad \text{pre}_q \setminus \text{pa}_q = \{h\} \quad \text{pre}_e \setminus \text{pa}_e = \{z, h, a\}$$

we thus obtain

$$h \perp\!\!\!\perp \varnothing \mid z \qquad a \perp\!\!\!\perp \{z, h\} \qquad q \perp\!\!\!\perp h \mid \{a, z\} \qquad e \perp\!\!\!\perp \{z, h, a\} \mid q$$

The relation $h \perp\!\!\!\perp \varnothing \mid z$ should be understood as "there is no variable from which h is independent given z" and should thus be dropped from the list. Note that we can possibly obtain more independence relations for variables that occur later in the topological ordering. This is because the set pre $\setminus$ pa can only increase when the predecessor set pre becomes larger.

(b) What are the independencies that follow from the local Markov property?

Solution. The non-descendants are

$$\text{nondesc}(a) = \{z, h\} \quad \text{nondesc}(z) = \{a\} \quad \text{nondesc}(h) = \{a, z, q, e\}$$

$$\text{nondesc}(q) = \{a, z, h\} \quad \text{nondesc}(e) = \{a, q, z, h\}$$

With the parent sets as before, the independencies that follow from the local Markov property are $x_i \perp\!\!\!\perp (\text{nondesc}(x_i) \setminus \text{pa}_i) \mid \text{pa}_i$, i.e.

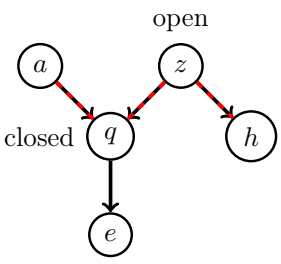
$$a \perp\!\!\!\perp \{z, h\} \qquad z \perp\!\!\!\perp a \qquad h \perp\!\!\!\perp \{a, q, e\} \mid z \qquad q \perp\!\!\!\perp h \mid \{a, z\} \qquad e \perp\!\!\!\perp \{a, z, h\} \mid q$$

(c) The independency relations obtained via the ordered and local Markov property include $q \perp\!\!\!\perp h \mid \{a, z\}$. Verify the independency using d-separation.

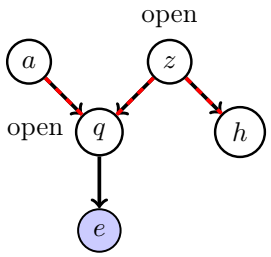
Solution. The only trail from q to h goes through z which is in a tail-tail configuration. Since z is part of the conditioning set, the trail is blocked and the result follows.

(d) Use d-separation to check whether $a \perp\!\!\!\perp h \mid e$ holds.

Solution. The trail from a to h is shown below in red together with the default states of the nodes along the trail.



Conditioning on e opens the q node since q in a collider configuration on the path.



The trail from a to h is thus active, which means that the relationship does not hold because $a \not\perp\!\!\!\perp h \mid e$ for some distributions that factorise over the graph.

(e) Assume all variables in the graph are binary. How many numbers do you need to specify, or learn from data, in order to fully specify the probability distribution?

Solution. The graph defines a set of probability mass functions (pmf) that factorise as

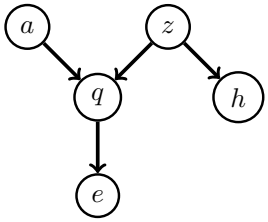
$$p(a, z, q, h, e) = p(a)p(z)p(q|a, z)p(h|z)p(e|q)$$

To specify a member of the set, we need to specify the (conditional) pmfs on the right-hand side. The (conditional) pmfs can be seen as tables, and the number of elements that we need to specified in the tables are:
- 1 for $p(a)$
- 1 for $p(z)$
- 4 for $p(q|a, z)$
- 2 for $p(h|z)$
- 2 for $p(e|q)$

In total, there are 10 numbers to specify. This is in contrast to $2^5 - 1 = 31$ for a distribution without independencies. Note that the number of parameters to specify could be further reduced by making parametric assumptions.

3.4 More on ordered and local Markov properties, d-separation

We continue with the investigation of the graph below



(a) Why can the ordered or local Markov property not be used to check whether $a \perp\!\!\!\perp h \mid e$ may hold?

Solution. The independencies that follow from the ordered or local Markov property require conditioning on parent sets. However, e is not a parent of any node so that the above independence assertion cannot be checked via the ordered or local Markov property.

(b) The independency relations obtained via the ordered and local Markov property include $a \perp\!\!\!\perp \{z, h\}$. Verify the independency using d-separation.

Solution. All paths from a to z or h pass through the node q that forms a head-head connection along that trail. Since neither q nor its descendant e is part of the conditioning set, the trail is blocked and the independence relation follows.

(c) Determine the Markov blanket of z.

Solution. The Markov blanket is given by the parents, children, and co-parents. Hence: $\mathrm{MB}(z) = \{a, q, h\}$.

(d) Verify that $q \perp\!\!\!\perp h \mid \{a, z\}$ holds by manipulating the probability distribution induced by the graph.

Solution. A basic definition of conditional statistical independence $x_1 \perp\!\!\!\perp x_2 \mid x_3$ is that the (conditional) joint $p(x_1, x_2 \mid x_3)$ equals the product of the (conditional) marginals $p(x_1 \mid x_3)$ and $p(x_2 \mid x_3)$. In other words, for discrete random variables,

$$x_1 \perp\!\!\!\perp x_2 \mid x_3 \quad \Longleftrightarrow \quad p(x_1, x_2 \mid x_3) = \left(\sum_{x_2} p(x_1, x_2 \mid x_3)\right)\left(\sum_{x_1} p(x_1, x_2 \mid x_3)\right) \tag{S.3.34}$$

We thus answer the question by showing that (use integrals in case of continuous random variables)

$$p(q, h|a, z) = \left(\sum_{h} p(q, h|a, z)\right)\left(\sum_{q} p(q, h|a, z)\right) \tag{S.3.35}$$

First, note that the graph defines a set of probability density or mass functions that factorise as

$$p(a, z, q, h, e) = p(a)p(z)p(q|a, z)p(h|z)p(e|q)$$

We then use the sum-rule to compute the joint distribution of (a, z, q, h), i.e. the distribution of all the variables that occur in $p(q, h|a, z)$

$$\begin{aligned} p(a, z, q, h) &= \sum_{e} p(a, z, q, h, e) && \text{(S.3.36)} \\ &= \sum_{e} p(a)p(z)p(q|a, z)p(h|z)p(e|q) && \text{(S.3.37)} \\ &= p(a)p(z)p(q|a, z)p(h|z)\underbrace{\sum_{e} p(e|q)}_{1} && \text{(S.3.38)} \\ &= p(a)p(z)p(q|a, z)p(h|z), && \text{(S.3.39)} \end{aligned}$$

where $\sum_e p(e|q) = 1$ because (conditional) pdfs/pmfs are normalised so that the integrate/sum to one. We further have

$$p(a,z) = \sum_{q,h} p(a,z,q,h) \tag{S.3.40}$$
$$= \sum_{q,h} p(a)p(z)p(q|a,z)p(h|z) \tag{S.3.41}$$
$$= p(a)p(z)\sum_q p(q|a,z)\sum_h p(h|z) \tag{S.3.42}$$
$$= p(a)p(z) \tag{S.3.43}$$

so that

$$p(q,h|a,z) = \frac{p(a,z,q,h)}{p(a,z)} \tag{S.3.44}$$
$$= \frac{p(a)p(z)p(q|a,z)p(h|z)}{p(a)p(z)} \tag{S.3.45}$$
$$= p(q|a,z)p(h|z). \tag{S.3.46}$$

We further see that $p(q|a,z)$ and $p(h|z)$ are the marginals of $p(q,h|a,z)$, i.e.

$$p(q|a,z) = \sum_h p(q,h|a,z) \tag{S.3.47}$$
$$p(h|z) = \sum_q p(q,h|a,z). \tag{S.3.48}$$

This means that

$$p(q,h|a,z) = \left(\sum_h p(q,h|a,z)\right)\left(\sum_q p(q,h|a,z)\right), \tag{S.3.49}$$

which shows that $q \perp\!\!\!\perp h|a,z$.

We see that using the graph to determine the independency is easier than manipulating the pmf/pdf.

3.5 Chest clinic (based on Barber, 2012, Exercise 3.3)

The directed graphical model in Figure 3.2 is about the diagnosis of lung disease (t=tuberculosis or l=lung cancer). In this model, a visit to some place "a" is thought to increase the probability of tuberculosis.

(a) Explain which of the following independence relationships hold for all distributions that factorise over the graph.

1. $t \perp\!\!\!\perp s \mid d$

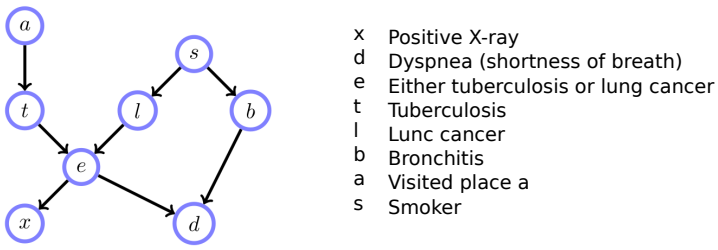


Figure 3.2: Graphical model for Exercise 3.5 (Barber Figure 3.15).

Solution.

- There are two trails from t to s: (t, e, l, s) and (t, e, d, b, s).
- The trail (t, e, l, s) features a collider node e that is opened by the conditioning variable d. The trail is thus active and we do not need to check the second trail because for independence all trails needed to be blocked.
- The independence relationship does thus generally not hold.

2. $l \perp\!\!\!\perp b \mid s$

Solution.

- There are two trails from l to b: (l, s, b) and (l, e, d, b)
- The trail (l, s, b) is blocked by s (s is in a tail-tail configuration and part of the conditioning set)
- The trail (l, e, d, b) is blocked by the collider configuration for node d.
- All trails are blocked so that the independence relation holds.

(b) Can we simplify $p(l|b, s)$ to $p(l|s)$?

Solution. Since $l \perp\!\!\!\perp b \mid s$, we have $p(l|b, s) = p(l|s)$.

3.6 More on the chest clinic (based on Barber, 2012, Exercise 3.3)

Consider the directed graphical model in Figure 3.2.

(a) Explain which of the following independence relationships hold for all distributions that factorise over the graph.

1. $a \perp\!\!\!\perp s \mid l$

Solution.

- There are two trails from a to s: (a, t, e, l, s) and (a, t, e, d, b, s)
- The trail (a, t, e, l, s) features a collider node e that blocks the trail (the trail is also blocked by l).
- The trail (a, t, e, d, b, s) is blocked by the collider node d.

- All trails are blocked so that the independence relation holds.

2. $a \perp\!\!\!\perp s \mid l, d$

Solution.

- There are two trails from a to s: (a, t, e, l, s) and (a, t, e, d, b, s)
- The trail (a, t, e, l, s) features a collider node e that is opened by the conditioning variable d but the l node is closed by the conditioning variable l: the trail is blocked
- The trail (a, t, e, d, b, s) features a collider node d that is opened by conditioning on d. On this trail, e is not in a head-head (collider) configuration) so that all nodes are open and the trail active.
- Hence, the independence relation does generally not hold.

(b) Let g be a (deterministic) function of x and t. Is the expected value $\mathbb{E}[g(x, t) \mid l, b]$ equal to $\mathbb{E}[g(x, t) \mid l]$?

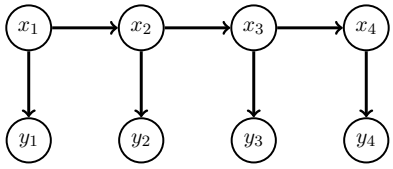
Solution. The question boils down to checking whether $x, t \perp\!\!\!\perp b \mid l$. For the independence relation to hold, all trails from both x and t to b need to be blocked by l.

- For x, we have the trails (x, e, l, s, b) and (x, e, d, b)
- Trail (x, e, l, s, b) is blocked by l
- Trail (x, e, d, b) is blocked by the collider configuration of node d.
- For t, we have the trails (t, e, l, s, b) and (t, e, d, b)
- Trail (t, e, l, s, b) is blocked by l.
- Trail (t, e, d, b) is blocked by the collider configuration of node d.

As all trails are blocked we have $x, t \perp\!\!\!\perp b \mid l$ and $\mathbb{E}[g(x, t) \mid l, b] = \mathbb{E}[g(x, t) \mid l]$.

3.7 Hidden Markov models

This exercise is about directed graphical models that are specified by the following DAG:



These models are called "hidden" Markov models because we typically assume to only observe the y_i and not the x_i that follow a Markov model.

(a) Show that all probabilistic models specified by the DAG factorise as

$$p(x_1, y_1, x_2, y_2, \ldots, x_4, y_4) = p(x_1)p(y_1|x_1)p(x_2|x_1)p(y_2|x_2)p(x_3|x_2)p(y_3|x_3)p(x_4|x_3)p(y_4|x_4)$$

Solution. From the definition of directed graphical models it follows that

$$p(x_1, y_1, x_2, y_2, \ldots, x_4, y_4) = \prod_{i=1}^{4} p(x_i|\text{pa}(x_i)) \prod_{i=1}^{4} p(y_i|\text{pa}(y_i)).$$

The result is then obtained by noting that the parent of y_i is given by x_i for all i, and that the parent of x_i is x_{i-1} for $i = 2, 3, 4$ and that x_1 does not have a parent ($\text{pa}(x_1) = \varnothing$).

(b) Derive the independencies implied by the ordered Markov property with the topological ordering $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$

Solution.

$$y_i \perp\!\!\!\perp x_1, y_1, \ldots, x_{i-1}, y_{i-1} \mid x_i \qquad x_i \perp\!\!\!\perp x_1, y_1, \ldots, x_{i-2}, y_{i-2}, y_{i-1} \mid x_{i-1}$$

(c) Derive the independencies implied by the ordered Markov property with the topological ordering $(x_1, x_2, \ldots, x_4, y_1, \ldots, y_4)$.

Solution. For the x_i, we use that for $i \geq 2$: $\text{pre}(x_i) = \{x_1, \ldots, x_{i-1}\}$ and $\text{pa}(x_i) = x_{i-1}$. For the y_i, we use that $\text{pre}(y_1) = \{x_1, \ldots, x_4\}$, that $\text{pre}(y_i) = \{x_1, \ldots, x_4, y_1, \ldots, y_{i-1}\}$ for $i > 1$, and that $\text{pa}(y_i) = x_i$. The ordered Markov property then gives:

$$\begin{aligned} x_3 &\perp\!\!\!\perp x_1 \mid x_2 & x_4 &\perp\!\!\!\perp \{x_1, x_2\} \mid x_3 \\ y_1 &\perp\!\!\!\perp \{x_2, x_3, x_4\} \mid x_1 & y_2 &\perp\!\!\!\perp \{x_1, x_3, x_4, y_1\} \mid x_2 \\ y_3 &\perp\!\!\!\perp \{x_1, x_2, x_4, y_1, y_2\} \mid x_3 & y_4 &\perp\!\!\!\perp \{x_1, x_2, x_3, y_1, y_2, y_3\} \mid x_4 \end{aligned}$$

(d) Does $y_4 \perp\!\!\!\perp y_1 \mid y_3$ hold?

Solution. The trail $y_1 - x_1 - x_2 - x_3 - x_4 - y_4$ is active: none of the nodes is in a collider configuration, so that their default state is open and conditioning on y_3 does not block any of the nodes on the trail.

While $x_1 - x_2 - x_3 - x_4$ forms a Markov chain, where e.g. $x_4 \perp\!\!\!\perp x_1 \mid x_3$ holds, this not so for the distribution of the y's.

3.8 Alternative characterisation of independencies

We have seen that $x \perp\!\!\!\perp y|z$ is characterised by $p(x, y|z) = p(x|z)p(y|z)$ or, equivalently, by $p(x|y, z) = p(x|z)$. Show that further equivalent characterisations are

$$p(x, y, z) = p(x|z)p(y|z)p(z) \quad \text{and} \tag{3.1}$$

$$p(x, y, z) = a(x, z)b(y, z) \quad \text{for some non-neg. functions } a(x, z) \text{ and } b(x, z). \tag{3.2}$$

The characterisation in Equation (3.2) is particularly important for undirected graphical models.

Solution. We first show the equivalence of $p(x, y|z) = p(x|z)p(y|z)$ and $p(x, y, z) = p(x|z)p(y|z)p(z)$: By the product rule, we have

$$p(x, y, z) = p(x, y|z)p(z).$$

If $p(x, y|z) = p(x|z)p(y|z)$, it follows that $p(x, y, z) = p(x|z)p(y|z)p(z)$. To show the opposite direction assume that $p(x, y, z) = p(x|z)p(y|z)p(z)$ holds. By comparison with the decomposition in the product rule, it follows that we must have $p(x, y|z) = p(x|z)p(y|z)$ whenever $p(z) > 0$ (it suffices to consider this case because for z where $p(z) = 0$, $p(x, y|z)$ may not be uniquely defined in the first place).

Equation (3.1) implies (3.2) with $a(x, z) = p(x|z)$ and $b(y, z) = p(y|z)p(z)$. We now show the inverse. Let us assume that $p(x, y, z) = a(x, z)b(y, z)$. By the product rule, we have

$$p(x, y|z)p(z) = a(x, z)b(y, z). \tag{S.3.50}$$

(S.3.51)

Summing over y gives

$$\sum_y p(x, y|z)p(z) = p(z) \sum_y p(x, y|z) \tag{S.3.52}$$

$$= p(z)p(x|z) \tag{S.3.53}$$

Moreover

$$\sum_y p(x, y|z)p(z) = \sum_y a(x, z)b(y, z) \tag{S.3.54}$$

$$= a(x, z) \sum_y b(y, z) \tag{S.3.55}$$

so that

$$a(x, z) = \frac{p(z)p(x|z)}{\sum_y b(y, z)} \tag{S.3.56}$$

Since the sum of $p(x|z)$ over x equals one we have

$$\sum_x a(x, z) = \frac{p(z)}{\sum_y b(y, z)}. \tag{S.3.57}$$

Now, summing $p(x, y|z)p(z)$ over x yields

$$\sum_x p(x, y|z)p(z) = p(z) \sum_x p(x, y|z). \tag{S.3.58}$$

$$= p(y|z)p(z) \tag{S.3.59}$$

We also have

$$\sum_x p(x, y|z)p(z) = \sum_x a(x, z)b(y, z) \tag{S.3.60}$$

$$= b(y, z) \sum_x a(x, z) \tag{S.3.61}$$

$$\overset{(S.3.57)}{=} b(y, z) \frac{p(z)}{\sum_y b(y, z)} \tag{S.3.62}$$

so that

$$p(y|z)p(z) = p(z)\frac{b(y,z)}{\sum_y b(y,z)} \tag{S.3.63}$$

We thus have

$$\begin{aligned} p(x,y,z) &= a(x,z)b(y,z) && \text{(S.3.64)}\\ &\overset{\text{(S.3.56)}}{=} \frac{p(z)p(x|z)}{\sum_y b(y,z)}b(y,z) && \text{(S.3.65)}\\ &= p(x|z)p(z)\frac{b(y,z)}{\sum_y b(y,z)} && \text{(S.3.66)}\\ &\overset{\text{(S.3.63)}}{=} p(x|z)p(y|z)p(z) && \text{(S.3.67)} \end{aligned}$$

which is Equation (3.1).

3.9 More on independencies

This exercise is on further properties and characterisations of statistical independence.

(a) Without using d-separation, show that $x \perp\!\!\!\perp \{y, w\} \mid z$ implies that $x \perp\!\!\!\perp y \mid z$ and $x \perp\!\!\!\perp w \mid z$.
Hint: use the definition of statistical independence in terms of the factorisation of pmfs/pdfs.

Solution. We consider the joint distribution $p(x, y, w|z)$. By assumption

$$p(x,y,w|z) = p(x|z)p(y,w|z) \tag{S.3.68}$$

We have to show that $x \perp\!\!\!\perp y|z$ and $x \perp\!\!\!\perp w|z$. For simplicity, we assume that the variables are discrete valued. If not, replace the sum below with an integral.

To show that $x \perp\!\!\!\perp y|z$, we marginalise $p(x, y, w|z)$ over w to obtain

$$\begin{aligned} p(x,y|z) &= \sum_w p(x,y,w|z) && \text{(S.3.69)}\\ &= \sum_w p(x|z)p(y,w|z) && \text{(S.3.70)}\\ &= p(x|z)\sum_w p(y,w|z) && \text{(S.3.71)} \end{aligned}$$

Since $\sum_w p(y, w|z)$ is the marginal $p(y|z)$, we have

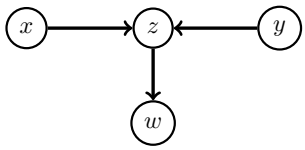
$$p(x,y|z) = p(x|z)p(y|z), \tag{S.3.72}$$

which means that $x \perp\!\!\!\perp y|z$.

To show that $x \perp\!\!\!\perp w|z$, we similarly marginalise $p(x, y, w|z)$ over y to obtain $p(x, w|z) = p(x|z)p(w|z)$, which means that $x \perp\!\!\!\perp w|z$.

(b) For the directed graphical model below, show that the following two statements hold without using d-separation:

$$x \perp\!\!\!\perp y \quad \text{and} \tag{3.3}$$
$$x \not\perp\!\!\!\perp y \mid w \tag{3.4}$$



The exercise shows that not only conditioning on a collider node but also on one of its descendents activates the trail between x and y. You can use the result that $x \perp\!\!\!\perp y|w \Leftrightarrow p(x, y, w) = a(x, w)b(y, w)$ for some non-negative functions $a(x, w)$ and $b(y, w)$.

Solution. The graphical model corresponds to the factorisation

$$p(x, y, z, w) = p(x)p(y)p(z|x, y)p(w|z).$$

For the marginal $p(x, y)$ we have to sum (integrate) over all (z, w)

$$p(x, y) = \sum_{z,w} p(x, y, z, w) \tag{S.3.73}$$
$$= \sum_{z,w} p(x)p(y)p(z|x, y)p(w|z) \tag{S.3.74}$$
$$= p(x)p(y) \sum_{z,w} p(z|x, y)p(w|z) \tag{S.3.75}$$
$$= p(x)p(y) \underbrace{\sum_{z} p(z|x, y)}_{1} \underbrace{\sum_{w} p(w|z)}_{1} \tag{S.3.76}$$
$$= p(x)p(y) \tag{S.3.77}$$

Since $p(x, y) = p(x)p(y)$ we have $x \perp\!\!\!\perp y$.

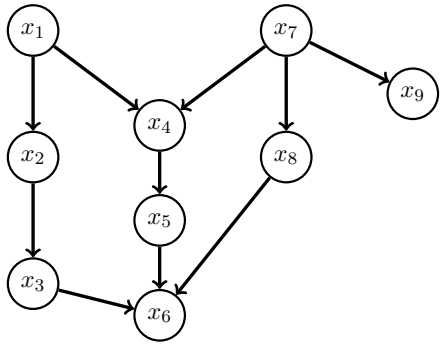
For $x \not\perp\!\!\!\perp y|w$, compute $p(x, y, w)$ and use the result $x \perp\!\!\!\perp y|w \Leftrightarrow p(x, y, w) = a(x, w)b(y, w)$.

$$p(x, y, w) = \sum_{z} p(x, y, z, w) \tag{S.3.78}$$
$$= \sum_{z} p(x)p(y)p(z|x, y)p(w|z) \tag{S.3.79}$$
$$= p(x) \underbrace{p(y) \sum_{z} p(z|x, y)p(w|z)}_{k(x,y,w)} \tag{S.3.80}$$

Since $p(x, y, w)$ cannot be factorised as $a(x, w)b(y, w)$, the relation $x \perp\!\!\!\perp y|w$ cannot generally hold.

3.10 Independencies in directed graphical models

Consider the following directed acyclic graph.



For each of the statements below, determine whether it holds for all probabilistic models that factorise over the graph. Provide a justification for your answer.

(a) $p(x_7|x_2) = p(x_7)$

Solution. Yes, it holds. x_2 is a non-descendant of x_7, $\text{pa}(x_7) = \varnothing$, and hence, by the local Markov property, $x_7 \perp\!\!\!\perp x_2$, so that $p(x_7|x_2) = p(x_7)$.

(b) $x_1 \not\perp\!\!\!\perp x_3$

Solution. No, does not hold. x_1 and x_3 are d-connected, which only implies independence for *some* and not all distributions that factorise over the graph. The graph generally only allows us to read out independencies and not dependencies.

(c) $p(x_1, x_2, x_4) \propto \phi_1(x_1, x_2)\phi_2(x_1, x_4)$ for some non-negative functions ϕ_1 and ϕ_2.

Solution. Yes, it holds. The statement is equivalent to $x_2 \perp\!\!\!\perp x_4 \mid x_1$. There are three trails from x_2 to x_4, which are all blocked:

1. $x_2 - x_1 - x_4$: this trail is blocked because x_1 is in a tail-tail connection and it is observed, which closes the node.
2. $x_2 - x_3 - x_6 - x_5 - x_4$: this trail is blocked because x_3, x_6, x_5 is in a collider configuration, and x_6 is not observed (and it does not have any descendants).
3. $x_2 - x_3 - x_6 - x_8 - x_7 - x_4$: this trail is blocked because x_3, x_6, x_8 is in a collider configuration, and x_6 is not observed (and it does not have any descendants).

Hence, by the global Markov property (d-separation), the independency holds.

(d) $x_2 \perp\!\!\!\perp x_9 \mid \{x_6, x_8\}$

Solution. No, does not hold. Conditioning on x_6 opens the collider node x_4 on the trail $x_2 - x_1 - x_4 - x_7 - x_9$, so that the trail is active.

(e) $x_8 \perp\!\!\!\perp \{x_2, x_9\} \mid \{x_3, x_5, x_6, x_7\}$

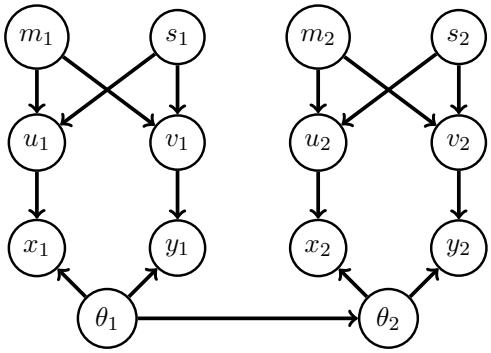
Solution. Yes, it holds. $\{x_3, x_5, x_6, x_7\}$ is the Markov blanket of x_8, so that x_8 is independent of remaining nodes given the Markov blanket.

(f) $\mathbb{E}[x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_8 \mid x_7] = 0$ if $\mathbb{E}[x_8 \mid x_7] = 0$

Solution. Yes, it holds. $\{x_2, x_3, x_4, x_5\}$ are non-descendants of x_8, and x_7 is the parent of x_8, so that $x_8 \perp\!\!\!\perp \{x_2, x_3, x_4, x_5\} \mid x_7$. This means that $\mathbb{E}[x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_8 \mid x_7] = \mathbb{E}[x_2 \cdot x_3 \cdot x_4 \cdot x_5 \mid x_7]\mathbb{E}[x_8 \mid x_7] = 0$.

3.11 Independencies in directed graphical models

Consider the following directed acyclic graph:



For each of the statements below, determine whether it holds for all probabilistic models that factorise over the graph. Provide a justification for your answer.

(a) $x_1 \perp\!\!\!\perp x_2$

Solution. Does not hold. The trail $x_1 - \theta_1 - \theta_2 - x_2$ is active (unblocked) because none of the nodes is in a collider configuration or in the conditioning set.

(b) $p(x_1, y_1, \theta_1, u_1) \propto \phi_A(x_1, \theta_1, u_1)\phi_B(y_1, \theta_1, u_1)$ for some non-negative functions ϕ_A and ϕ_B

Solution. Holds. The statement is equivalent to $x_1 \perp\!\!\!\perp y_1 \mid \{\theta_1, u_1\}$. The conditioning set $\{\theta_1, u_1\}$ blocks all trails from x_1 to y_1 because they are both only in serial configurations in all trails from x_1 to y_1, hence the independency holds by the

global Markov property. Alternative justification: the conditioning set is the Markov blanket of x_1, and x_1 and y_1 are not neighbours which implies the independency.

(c) $v_2 \perp\!\!\!\perp \{u_1, v_1, u_2, x_2\} \mid \{m_2, s_2, y_2, \theta_2\}$

Solution. Holds. The conditioning set is the Markov blanket of v_2 (the set of parents, children, and co-parents): the set of parents is $\text{pa}(v_2) = \{m_2, s_2\}$, y_2 is the only child of v_2, and θ_2 is the only other parent of y_2. And v_2 is independent of all other variables given its Markov blanket.

(d) $\mathbb{E}[m_2 \mid m_1] = \mathbb{E}[m_2]$

Solution. Holds. There are four trails from m_1 to m_2, namely via x_1, via y_1, via x_2, via y_2. In all trails the four variables are in a collider configuration, so that each of the trails is blocked. By the global Markov property (d-separation), this means that $m_1 \perp\!\!\!\perp m_2$ which implies that $\mathbb{E}[m_2 \mid m_1] = \mathbb{E}[m_2]$.

Alternative justification 1: m_2 is a non-descendent of m_1 and $\text{pa}(m_2) = \varnothing$. By the directed local Markov property, a variable is independent from its non-descendents given the parents, hence $m_2 \perp\!\!\!\perp m_1$.

Alternative justification 2: We can choose a topological ordering where m_1 and m_2 are the first two variables. Moreover, their parent sets are both empty. By the directed ordered Markov, we thus have $m_1 \perp\!\!\!\perp m_2$.

Chapter 4

Undirected Graphical Models

Exercises

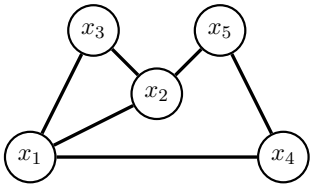
4.1 Visualising and analysing Gibbs distributions via undirected graphs

We here consider the Gibbs distribution

$$p(x_1, \ldots, x_5) \propto \phi_{12}(x_1, x_2)\phi_{13}(x_1, x_3)\phi_{14}(x_1, x_4)\phi_{23}(x_2, x_3)\phi_{25}(x_2, x_5)\phi_{45}(x_4, x_5)$$

(a) Visualise it as an undirected graph.

Solution. We draw a node for each random variable x_i. There is an edge between two nodes if the corresponding variables co-occur in a factor.



(b) What are the neighbours of x_3 in the graph?

Solution. The neighbours are all the nodes for which there is a single connecting edge. Thus: $\text{ne}(x_3) = \{x_1, x_2\}$. (Note that sometimes, we may denote $\text{ne}(x_3)$ by ne_3.)

(c) Do we have $x_3 \perp\!\!\!\perp x_4 \mid x_1, x_2$?

Solution. Yes. The conditioning set $\{x_1, x_2\}$ equals ne_3, which is also the Markov blanket of x_3. This means that x_3 is conditionally independent of all the other variables given $\{x_1, x_2\}$, i.e. $x_3 \perp\!\!\!\perp x_4, x_5 \mid x_1, x_2$, which implies that $x_3 \perp\!\!\!\perp x_4 \mid x_1, x_2$. (One can also use graph separation to answer the question.)

(d) What is the Markov blanket of x_4?

Solution. The Markov blanket of a node in a undirected graphical model equals the set of its neighbours: $\text{MB}(x_4) = \text{ne}(x_4) = \text{ne}_4 = \{x_1, x_5\}$. This implies, for example, that $x_4 \perp\!\!\!\perp x_2, x_3 \mid x_1, x_5$.

(e) On which minimal set of variables A do we need to condition to have $x_1 \perp\!\!\!\perp x_5 \mid A$?

Solution. We first identify all trails from x_1 to x_5. There are three such trails: (x_1, x_2, x_5), (x_1, x_3, x_2, x_5), and (x_1, x_4, x_5). Conditioning on x_2 blocks the first two trails, conditioning on x_4 blocks the last. We thus have: $x_1 \perp\!\!\!\perp x_5 \mid x_2, x_4$, so that $A = \{x_2, x_4\}$.

4.2 Factorisation and independencies for undirected graphical models

Consider the undirected graphical model defined by the graph in Figure 4.1.

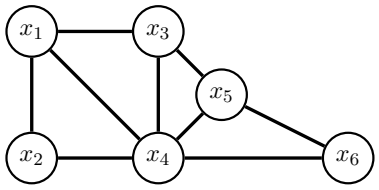


Figure 4.1: Graph for Exercise 4.2

(a) What is the set of Gibbs distributions that is induced by the graph?

Solution. The graph in Figure 4.1 has four maximal cliques:

$$(x_1, x_2, x_4) \quad (x_1, x_3, x_4) \quad (x_3, x_4, x_5) \quad (x_4, x_5, x_6)$$

The Gibbs distributions are thus

$$p(x_1, \ldots, x_6) \propto \phi_1(x_1, x_2, x_4)\phi_2(x_1, x_3, x_4)\phi_3(x_3, x_4, x_5)\phi_4(x_4, x_5, x_6)$$

(b) Let p be a pdf that factorises according to the graph. Does $p(x_3|x_2, x_4) = p(x_3|x_4)$ hold?

Solution. $p(x_3|x_2, x_4) = p(x_3|x_4)$ means that $x_3 \perp\!\!\!\perp x_2 \mid x_4$. We can use the graph to check whether this generally holds for pdfs that factorise according to the graph. There are multiple trails from x_3 to x_2, including the trail (x_3, x_1, x_2), which is not blocked by x_4. From the graph, we thus cannot conclude that $x_3 \perp\!\!\!\perp x_2 \mid x_4$, and $p(x_3|x_2, x_4) = p(x_3|x_4)$ will generally not hold (the relation may hold for some carefully defined factors ϕ_i).

(c) Explain why $x_2 \perp\!\!\!\perp x_5 \mid x_1, x_3, x_4, x_6$ holds for all distributions that factorise over the graph.

Solution. Distributions that factorise over the graph satisfy the pairwise Markov property. Since x_2 and x_5 are not neighbours, and x_1, x_3, x_4, x_6 are the remaining nodes in the graph, the independence relation follows from the pairwise Markov property.

(d) Assume you would like to approximate $\mathbb{E}(x_1x_2x_5 \mid x_3, x_4)$, i.e. the expected value of the product of x_1, x_2, and x_5 given x_3 and x_4, with a sample average. Do you need to have joint observations for all five variables $x_1, \ldots, x_5$?

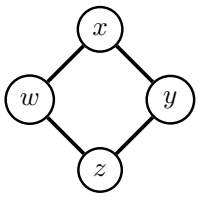
Solution. In the graph, all trails from $\{x_1, x_2\}$ to x_5 are blocked by $\{x_3, x_4\}$, so that $x_1, x_2 \perp\!\!\!\perp x_5 \mid x_3, x_4$. We thus have

$$\mathbb{E}(x_1 x_2 x_5 \mid x_3, x_4) = \mathbb{E}(x_1 x_2 \mid x_3, x_4)\mathbb{E}(x_5 \mid x_3, x_4).$$

Hence, we only need joint observations of (x_1, x_2, x_3, x_4) and (x_3, x_4, x_5). Variables (x_1, x_2) and x_5 do not need to be jointly measured.

4.3 Factorisation and independencies for undirected graphical models

Consider the undirected graphical model defined by the following graph, sometimes called a diamond configuration.



(a) How do the pdfs/pmfs of the undirected graphical model factorise?

Solution. The maximal cliques are (x, w), (w, z), (z, y) and (x, y). The undirected graphical model thus consists of pdfs/pmfs that factorise as follows

$$p(x, w, z, y) \propto \phi_1(x, w)\phi_2(w, z)\phi_3(z, y)\phi_4(x, y) \tag{S.4.1}$$

(b) List all independencies that hold for the undirected graphical model.

Solution. We can generate the independencies by conditioning on progressively larger sets. Since there is a trail between any two nodes, there are no unconditional independencies. If we condition on a single variable, there is still a trail that connects the remaining ones. Let us thus consider the case where we condition on two nodes. By graph separation, we have

$$w \perp\!\!\!\perp y \mid x, z \qquad x \perp\!\!\!\perp z \mid w, y \tag{S.4.2}$$

These are all the independencies that hold for the model, since conditioning on three nodes does not lead to any independencies in a model with four variables.

4.4 Factorisation from the Markov blankets I

Assume you know the following Markov blankets for all variables $x_1, \ldots, x_4, y_1, \ldots y_4$ of a

pdf or pmf $p(x_1, \ldots, x_4, y_1, \ldots, y_4)$.

$$\mathrm{MB}(x_1) = \{x_2, y_1\} \quad \mathrm{MB}(x_2) = \{x_1, x_3, y_2\} \quad \mathrm{MB}(x_3) = \{x_2, x_4, y_3\} \quad \mathrm{MB}(x_4) = \{x_3, y_4\} \tag{4.1}$$

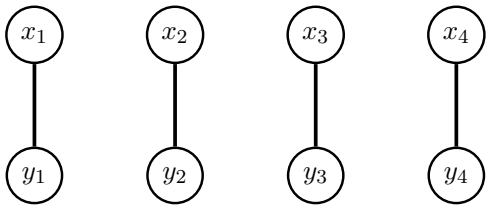
$$\mathrm{MB}(y_1) = \{x_1\} \qquad \mathrm{MB}(y_2) = \{x_2\} \qquad \mathrm{MB}(y_3) = \{x_3\} \qquad \mathrm{MB}(y_4) = \{x_4\} \tag{4.2}$$

Assuming that p is positive for all possible values of its variables, how does p factorise?

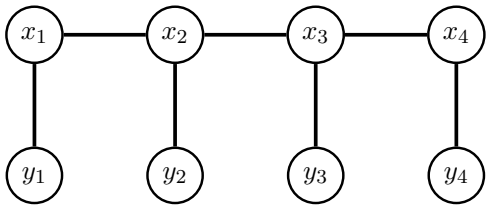
Solution. In undirected graphical models, the Markov blanket for a variable is the same as the set of its neighbours. Hence, when we are given all Markov blankets we know what local Markov property p must satisfy. For positive distributions we have an equivalence between p satisfying the local Markov property and p factorising over the graph. Hence, to specify the factorisation of p it suffices to construct the undirected graph H based on the Markov blankets and then read out the factorisation.

We need to build a graph where the neighbours of each variable equals the indicated Markov blanket. This can be easily done by starting with an empty graph and connecting each variable to the variables in its Markov blanket.

We see that each y_i is only connected to x_i. Including those Markov blankets we get the following graph:



Connecting the x_i to their neighbours according to the Markov blanket thus gives:



The graph has maximal cliques of size two, namely the $x_i - y_i$ for $i = 1, \ldots, 4$, and the $x_i - x_{i+1}$ for $i = 1, \ldots, 3$. Given the equivalence between the local Markov property and factorisation for positive distributions, we know that p must factorise as

$$p(x_1, \ldots, x_4, y_1, \ldots, y_4) = \frac{1}{Z} \prod_{i=1}^{3} m_i(x_i, x_{i+1}) \prod_{i=1}^{4} g_i(x_i, y_i), \tag{S.4.3}$$

where $m_i(x_i, x_{i+1}) > 0$, $g(x_i, y_i) > 0$ are positive factors (potential functions).

The graphical model corresponds to an undirected version of a hidden Markov model where the x_i are the unobserved (latent, hidden) variables and the y_i are the observed ones. Note that the x_i form a Markov chain.

4.5 Factorisation from the Markov blankets II

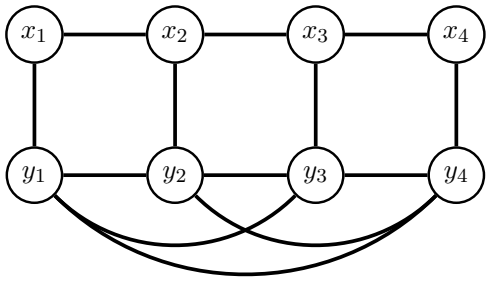
We consider the same setup as in Exercise 4.4 but we now assume that we do not know all Markov blankets but only

$$\mathrm{MB}(x_1) = \{x_2, y_1\} \quad \mathrm{MB}(x_2) = \{x_1, x_3, y_2\} \quad \mathrm{MB}(x_3) = \{x_2, x_4, y_3\} \quad \mathrm{MB}(x_4) = \{x_3, y_4\} \tag{4.3}$$

Without inserting more independencies than those specified by the Markov blankets, draw the graph over which p factorises and state the factorisation. (Again assume that p is positive for all possible values of its variables).

Solution. We take the same approach as in Exercise 4.4. In particular, the Markov blankets of a variable are its neighbours in the graph. But since we are not given all Markov blankets and are not allowed to insert additional independencies, we must assume that each y_i is connected to all the other y's. For example, if we didn't connect y_1 and y_4 we would assert the additional independency $y_1 \perp\!\!\!\perp y_4 \mid x_1, x_2, x_3, x_4, y_2, y_3$.

We thus have a graph as follows:



The factorisation thus is

$$p(x_1, \ldots, x_4, y_1, \ldots, y_4) = \frac{1}{Z} g(y_1, \ldots, y_4) \prod_{i=1}^{3} m_i(x_i, x_{i+1}) \prod_{i=1}^{4} g_i(x_i, y_i), \tag{S.4.4}$$

where the $m_i(x_i, x_{i+1})$, $g_i(x_i, y_i)$ and $g(y_1, \ldots, y_4)$ are positive factors. Compared to the factorisation in Exercise 4.4, we still have the Markov structure for the x_i, but only a single factor for (y_1, y_2, y_3, y_4) to avoid inserting independencies beyond those specified by the given Markov blankets.

4.6 Undirected graphical model with pairwise potentials

We here consider Gibbs distributions where the factors only depend on two variables at a time. The probability density or mass functions over d random variables $x_1, \ldots, x_d$ then take the form

$$p(x_1, \ldots, x_d) \propto \prod_{i \leq j} \phi_{ij}(x_i, x_j)$$

Such models are sometimes called pairwise Markov networks.

(a) Let $p(x_1, \ldots, x_d) \propto \exp\left(-\frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x}\right)$ where $\mathbf{A}$ is symmetric and $\mathbf{x} = (x_1, \ldots, x_d)^\top$. What are the corresponding factors ϕ_{ij} for $i \leq j$?

Solution. Denote the (i, j)-th element of $\mathbf{A}$ by a_{ij}. We have

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \sum_{ij} a_{ij} x_i x_j \tag{S.4.5}$$

$$= \sum_{i<j} 2a_{ij} x_i x_j + \sum_i a_{ii} x_i^2 \tag{S.4.6}$$

where the second line follows from $\mathbf{A}^\top = \mathbf{A}$. Hence,

$$-\frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x} = -\frac{1}{2}\sum_{i<j} 2a_{ij} x_i x_j - \frac{1}{2}\sum_i a_{ii} x_i^2 - \sum_i b_i x_i \tag{S.4.7}$$

so that

$$\phi_{ij}(x_i, x_j) = \begin{cases} \exp\left(-a_{ij} x_i x_j\right) & \text{if } i < j \\ \exp\left(-\frac{1}{2} a_{ii} x_i^2 - b_i x_i\right) & \text{if } i = j \end{cases} \tag{S.4.8}$$

For $\mathbf{x} \in \mathbb{R}^d$, the distribution is a Gaussian with $\mathbf{A}$ equal to the inverse covariance matrix. For binary $\mathbf{x}$, the model is known as Ising model or Boltzmann machine. For $x_i \in \{-1, 1\}$, $x_i^2 = 1$ for all i, so that the a_{ii} are constants that can be absorbed into the normalisation constant. This means that for $x_i \in \{-1, 1\}$, we can work with matrices $\mathbf{A}$ that have zeros on the diagonal.

(b) For $p(x_1, \ldots, x_d) \propto \exp\left(-\frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x}\right)$, show that $x_i \perp\!\!\!\perp x_j \mid \{x_1, \ldots, x_d\} \setminus \{x_i, x_j\}$ if the (i, j)-th element of $\mathbf{A}$ is zero.

Solution. The previous question showed that we can write $p(x_1, \ldots, x_d) \propto \prod_{i \leq j} \phi_{ij}(x_i, x_j)$ with potentials as in Equation (S.4.8). Consider two variables x_i and x_j for fixed (i, j). They only appear in the factorisation via the potential ϕ_{ij}. If $a_{ij} = 0$, the factor ϕ_{ij} becomes a constant, and no other factor contains x_i and x_j, which means that there is no edge between x_i and x_j if $a_{ij} = 0$. By the pairwise Markov property it then follows that $x_i \perp\!\!\!\perp x_j \mid \{x_1, \ldots, x_d\} \setminus \{x_i, x_j\}$.

4.7 Restricted Boltzmann machine (based on Barber, 2012, Exercise 4.4)

The restricted Boltzmann machine is an undirected graphical model for binary variables $\mathbf{v} = (v_1, \ldots, v_n)^\top$ and $\mathbf{h} = (h_1, \ldots, h_m)^\top$ with a probability mass function equal to

$$p(\mathbf{v}, \mathbf{h}) \propto \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h}\right), \tag{4.4}$$

where $\mathbf{W}$ is a $n \times m$ matrix. Both the v_i and h_i take values in $\{0, 1\}$. The v_i are called the "visibles" variables since they are assumed to be observed while the h_i are the hidden variables since it is assumed that we cannot measure them.

(a) Use graph separation to show that the joint conditional $p(\mathbf{h}|\mathbf{v})$ factorises as

$$p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{m} p(h_i|\mathbf{v}).$$

Solution. Figure 4.2 on the left shows the undirected graph for $p(\mathbf{v}, \mathbf{h})$ with $n = 3, m = 2$. We note that the graph is bi-partite: there are only direct connections between the h_i and the v_i. Conditioning on $\mathbf{v}$ thus blocks all trails between the h_i (graph on the right). This means that the h_i are independent from each other given $\mathbf{v}$ so that

$$p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{m} p(h_i|\mathbf{v}).$$

Figure 4.2: Left: Graph for $p(\mathbf{v}, \mathbf{h})$. Right: Graph for $p(\mathbf{h}|\mathbf{v})$

(b) Show that

$$p(h_i = 1|\mathbf{v}) = \frac{1}{1 + \exp\left(-b_i - \sum_j W_{ji} v_j\right)} \tag{4.5}$$

where W_{ji} is the (ji)-th element of $\mathbf{W}$, so that $\sum_j W_{ji} v_j$ is the inner product (scalar product) between the i-th column of $\mathbf{W}$ and $\mathbf{v}$.

Solution. For the conditional pmf $p(h_i|\mathbf{v})$ any quantity that does not depend on h_i can be considered to be part of the normalisation constant. A general strategy is to first work out $p(h_i|\mathbf{v})$ up to the normalisation constant and then to normalise it afterwards.

We begin with $p(\mathbf{h}|\mathbf{v})$:

$$\begin{aligned}
p(\mathbf{h}|\mathbf{v}) &= \frac{p(\mathbf{h}, \mathbf{v})}{p(\mathbf{v})} && \text{(S.4.9)} \\
&\propto p(\mathbf{h}, \mathbf{v}) && \text{(S.4.10)} \\
&\propto \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h}\right) && \text{(S.4.11)} \\
&\propto \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{b}^\top \mathbf{h}\right) && \text{(S.4.12)} \\
&\propto \exp\left(\sum_i \sum_j v_j W_{ji} h_i + \sum_i b_i h_i\right) && \text{(S.4.13)}
\end{aligned}$$

As we are interested in $p(h_i|\mathbf{v})$ for a fixed i, we can drop all the terms not depending on that h_i, so that

$$p(h_i|\mathbf{v}) \propto \exp\left(\sum_j v_j W_{ji} h_i + b_i h_i\right) \tag{S.4.14}$$

Since h_i only takes two values, 0 and 1, normalisation is here straightforward. Call the unnormalised pmf $\tilde{p}(h_i|\mathbf{v})$,

$$\tilde{p}(h_i|\mathbf{v}) = \exp\left(\sum_j v_j W_{ji} h_i + b_i h_i\right). \tag{S.4.15}$$

We then have

$$p(h_i|\mathbf{v}) = \frac{\tilde{p}(h_i|\mathbf{v})}{\tilde{p}(h_i = 0|\mathbf{v}) + \tilde{p}(h_i = 1|\mathbf{v})} \tag{S.4.16}$$

$$= \frac{\tilde{p}(h_i|\mathbf{v})}{1 + \exp\left(\sum_j v_j W_{ji} + b_i\right)} \tag{S.4.17}$$

$$= \frac{\exp\left(\sum_j v_j W_{ji} h_i + b_i h_i\right)}{1 + \exp\left(\sum_j v_j W_{ji} + b_i\right)}, \tag{S.4.18}$$

so that

$$p(h_i = 1|\mathbf{v}) = \frac{\exp\left(\sum_j v_j W_{ji} + b_i\right)}{1 + \exp\left(\sum_j v_j W_{ji} + b_i\right)} \tag{S.4.19}$$

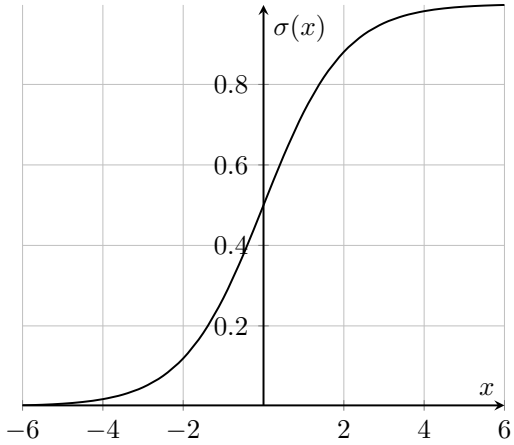
$$= \frac{1}{1 + \exp\left(-\sum_j v_j W_{ji} - b_i\right)}. \tag{S.4.20}$$

The probability $p(h = 0|\mathbf{v})$ equals $1 - p(h_i = 1|\mathbf{v})$, which is

$$p(h_i = 0|\mathbf{v}) = \frac{1 + \exp\left(\sum_j v_j W_{ji} + b_i\right)}{1 + \exp\left(\sum_j v_j W_{ji} + b_i\right)} - \frac{\exp\left(\sum_j v_j W_{ji} + b_i\right)}{1 + \exp\left(\sum_j v_j W_{ji} + b_i\right)} \tag{S.4.21}$$

$$= \frac{1}{1 + \exp\left(\sum_j W_{ji} v_j + b_i\right)} \tag{S.4.22}$$

The function $x \mapsto 1/(1 + \exp(-x))$ is called the logistic function. It is a sigmoid function and is thus sometimes denoted by $\sigma(x)$. For other versions of the sigmoid function, see https://en.wikipedia.org/wiki/Sigmoid_function.



With that notation, we have

$$p(h_i = 1|\mathbf{v}) = \sigma\left(\sum_j W_{ji} v_j + b_i\right).$$

(c) Use a symmetry argument to show that

$$p(\mathbf{v}|\mathbf{h}) = \prod_i p(v_i|\mathbf{h}) \quad \text{and} \quad p(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp\left(-a_i - \sum_j W_{ij} h_j\right)}$$

Solution. Since $\mathbf{v}^\top \mathbf{W}\mathbf{h}$ is a scalar we have $(\mathbf{v}^\top \mathbf{W}\mathbf{h})^\top = \mathbf{h}^\top \mathbf{W}^\top \mathbf{v} = \mathbf{v}^\top \mathbf{W}\mathbf{h}$, so that

$$p(\mathbf{v}, \mathbf{h}) \propto \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h}\right) \tag{S.4.23}$$

$$\propto \exp\left(\mathbf{h}^\top \mathbf{W}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h} + \mathbf{a}^\top \mathbf{v}\right). \tag{S.4.24}$$

To derive the result, we note that $\mathbf{v}$ and a now take the place of $\mathbf{h}$ and $\mathbf{b}$ from before, and that we now have $\mathbf{W}^\top$ rather than $\mathbf{W}$. In Equation (4.5), we thus replace h_i with v_i, b_i with a_i, and W_{ji} with W_{ij} to obtain $p(v_i = 1|\mathbf{h})$. In terms of the sigmoid function, we have

$$p(v_i = 1|\mathbf{h}) = \sigma\left(\sum_j W_{ij} h_j + a_i\right).$$

Note that while $p(\mathbf{v}|\mathbf{h})$ factorises, the marginal $p(\mathbf{v})$ does generally not. The marginal

$p(\mathbf{v})$ can here be obtained in closed form up to its normalisation constant.

$$p(\mathbf{v}) = \sum_{\mathbf{h}\in\{0,1\}^m} p(\mathbf{v},\mathbf{h}) \tag{S.4.25}$$

$$= \frac{1}{Z}\sum_{\mathbf{h}\in\{0,1\}^m} \exp\left(\mathbf{v}^\top\mathbf{W}\mathbf{h} + \mathbf{a}^\top\mathbf{v} + \mathbf{b}^\top\mathbf{h}\right) \tag{S.4.26}$$

$$= \frac{1}{Z}\sum_{\mathbf{h}\in\{0,1\}^m} \exp\left(\sum_{ij} v_i h_j W_{ij} + \sum_i a_i v_i + \sum_j b_j h_j\right) \tag{S.4.27}$$

$$= \frac{1}{Z}\sum_{\mathbf{h}\in\{0,1\}^m} \exp\left(\sum_{j=1}^m h_j\left[\sum_i v_i W_{ij} + b_j\right] + \sum_i a_i v_i\right) \tag{S.4.28}$$

$$= \frac{1}{Z}\sum_{\mathbf{h}\in\{0,1\}^m} \prod_{j=1}^m \exp\left(h_j\left[\sum_i v_i W_{ij} + b_j\right]\right)\exp\left(\sum_i a_i v_i\right) \tag{S.4.29}$$

$$= \frac{1}{Z}\exp\left(\sum_i a_i v_i\right)\sum_{\mathbf{h}\in\{0,1\}^m} \prod_{j=1}^m \exp\left(h_j\left[\sum_i v_i W_{ij} + b_j\right]\right) \tag{S.4.30}$$

$$= \frac{1}{Z}\exp\left(\sum_i a_i v_i\right)\sum_{h_1,\ldots,h_m} \prod_{j=1}^m \exp\left(h_j\left[\sum_i v_i W_{ij} + b_j\right]\right) \tag{S.4.31}$$

Importantly, each term in the product only depends on a single h_j, so that by sequentially applying the distributive law, we have

$$\sum_{h_1,\ldots,h_m}\prod_{j=1}^m \exp\left(h_j\left[\sum_i v_i W_{ij} + b_j\right]\right) = \left[\sum_{h_1,\ldots,h_{m-1}}\prod_{j=1}^{m-1} \exp\left(h_j\left[\sum_i v_i W_{ij} + b_j\right]\right)\right]\cdot \sum_{h_m}\exp\left(h_m\left[\sum_i v_i W_{im} + b_m\right]\right) \tag{S.4.32}$$

$$= \ldots \tag{S.4.33}$$

$$= \prod_{j=1}^m \left[\sum_{h_j}\exp\left(h_j\left[\sum_i v_i W_{ij} + b_j\right]\right)\right]$$

Since $h_j \in \{0,1\}$, we obtain

$$\sum_{h_j}\exp\left(h_j\left[\sum_i v_i W_{ij} + b_j\right]\right) = 1 + \exp\left(\sum_i v_i W_{ij} + b_j\right) \tag{S.4.34}$$

and thus

$$p(\mathbf{v}) = \frac{1}{Z}\exp\left(\sum_i a_i v_i\right)\prod_{j=1}^m\left[1 + \exp\left(\sum_i v_i W_{ij} + b_j\right)\right]. \tag{S.4.35}$$

Note that in the derivation of $p(\mathbf{v})$ we have not used the assumption that the visibles v_i are binary. The same expression would thus obtained if the visibles were defined in another space, e.g. the real numbers.

While $p(\mathbf{v})$ is written as a product, $p(\mathbf{v})$ does not factorise into terms that depend on subsets of the v_i. On the contrary, all v_i are present in all factors. Since $p(\mathbf{v})$ does not factorise, computing the normalising Z is expensive. For binary visibles $v_i \in \{0, 1\}$, Z equals

$$Z = \sum_{\mathbf{v}\in\{0,1\}^n} \exp\left(\sum_i a_i v_i\right) \prod_{j=1}^{m} \left[1 + \exp\left(\sum_i v_i W_{ij} + b_j\right)\right] \tag{S.4.36}$$

where we have to sum over all 2^n configurations of the visibles $\mathbf{v}$. This is computationally expensive, or even prohibitive if n is large ($2^{20} = 1048576$, $2^{30} > 10^9$). Note that different values of a_i, b_i, W_{ij} yield different values of Z. (This is a reason why Z is called the partition *function* when the a_i, b_i, W_{ij} are free parameters.)

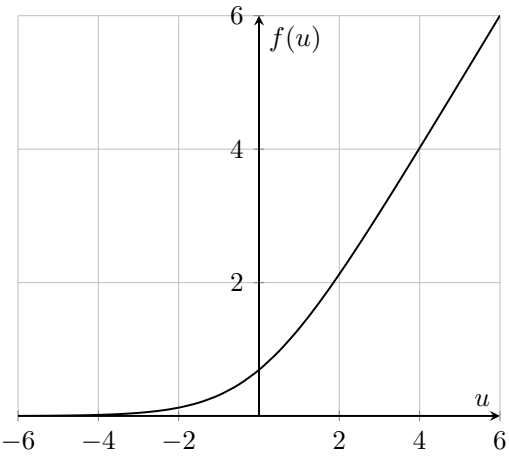
It is instructive to write $p(\mathbf{v})$ in the log-domain,

$$\log p(\mathbf{v}) = \log Z + \sum_{i=1}^{n} a_i v_i + \sum_{j=1}^{m} \log\left[1 + \exp\left(\sum_i v_i W_{ij} + b_j\right)\right], \tag{S.4.37}$$

and to introduce the nonlinearity $f(u)$,

$$f(u) = \log\left[1 + \exp(u)\right], \tag{S.4.38}$$

which is called the softplus function and plotted below. The softplus function is a smooth approximation of $\max(0, u)$, see e.g. https://en.wikipedia.org/wiki/Rectifier_(neural_networks)



With the softplus function $f(u)$, we can write $\log p(\mathbf{v})$ as

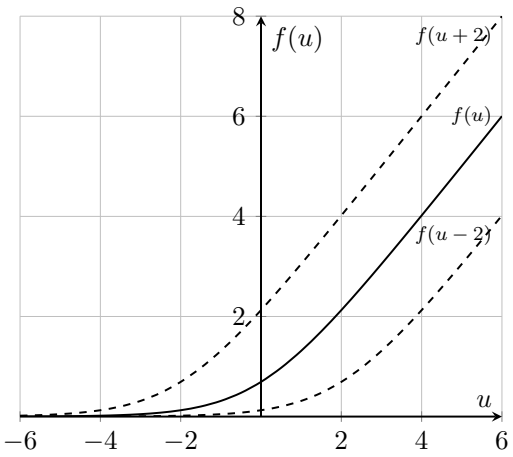
$$\log p(\mathbf{v}) = \log Z + \sum_{i=1}^{n} a_i v_i + \sum_{j=1}^{m} f\left(\sum_i v_i W_{ij} + b_j\right). \tag{S.4.39}$$

The parameter b_j plays the role of a threshold as shown in the figure below. The terms $f\left(\sum_i v_i W_{ij} + b_j\right)$ can be interpreted in terms of feature detection. The sum $\sum_i v_i W_{ij}$ is the inner product between $\mathbf{v}$ and the j-th column of $\mathbf{W}$, and the inner product is largest if $\mathbf{v}$ equals the j-th column. We can thus consider the columns of

$\mathbf{W}$ to be feature-templates, and the $f\left(\sum_i v_i W_{ij} + b_j\right)$ a way to measure how much of each feature is present in $\mathbf{v}$.

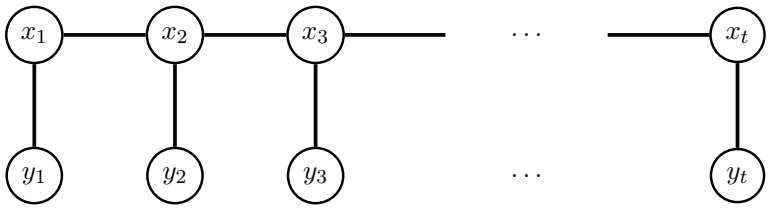
Further, $\sum_i v_i W_{ij} + b_j$ is also the input to the sigmoid function when computing $p(h_j = 1|\mathbf{v})$. Thus, the conditional probability for h_j to be one, i.e. "active", can be considered to be an indicator of the presence of the j-th feature (j-th column of $\mathbf{W}$) in the input $\mathbf{v}$.

If v is such that $\sum_i v_i W_{ij} + b_j$ is large for many j, i.e. if many features are detected, then $f\left(\sum_i v_i W_{ij} + b_j\right)$ will be non-zero for many j, and $\log p(\mathbf{v})$ will be large.



4.8 Hidden Markov models and change of measure

Consider the following undirected graph for a hidden Markov model where the y_i correspond to observed (visible) variables and the x_i to unobserved (hidden/latent) variables.



The graph implies the following factorisation

$$p(x_1, \ldots, x_t, y_1, \ldots, y_t) \propto \phi_1^y(x_1, y_1) \prod_{i=2}^{t} \phi_i^x(x_{i-1}, x_i) \phi_i^y(x_i, y_i), \tag{4.6}$$

where the ϕ_i^x and ϕ_i^y are non-negative factors.

Let us consider the situation where $\prod_{i=2}^{t} \phi_i^x(x_{i-1}, x_i)$ equals

$$f(\mathbf{x}) = \prod_{i=2}^{t} \phi_i^x(x_{i-1}, x_i) = f_1(x_1) \prod_{i=2}^{t} f_i(x_i|x_{i-1}), \tag{4.7}$$

with $\mathbf{x} = (x_1, \ldots, x_t)$ and where the f_i are (conditional) pdfs. We thus have

$$p(x_1, \ldots, x_t, y_1, \ldots, y_t) \propto f_1(x_1) \prod_{i=2}^{t} f_i(x_i|x_{i-1}) \prod_{i=1}^{t} \phi_i^y(x_i, y_i). \tag{4.8}$$

(a) Provide a factorised expression for $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$

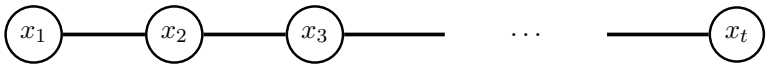
Solution. For fixed (observed) values of the y_i, $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$ factorises as

$$p(x_1, \ldots, x_t|y_1, \ldots, y_t) \propto f_1(x_1)g_1(x_1) \prod_{i=1}^{t} f_i(x_i|x_{i-1})g_i(x_i). \tag{S.4.40}$$

where $g_i(x_i)$ is $\phi_i^y(x_i, y_i)$ for a fixed value of y_i.

(b) Draw the undirected graph for $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$

Solution. Conditioning corresponds to removing nodes from an undirected graph. We thus have the following Markov chain for $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$.



(c) Show that if $\phi_i^y(x_i, y_i)$ equals the conditional pdf of y_i given x_i, i.e. $p(y_i|x_i)$, the marginal $p(x_1, \ldots, x_t)$, obtained by integrating out $y_1, \ldots, y_t$ from (4.8), equals $f(\mathbf{x})$.

Solution. In this setting all factors in (4.8) are conditional pdfs and we are dealing with a directed graphical model that factorises as

$$p(x_1, \ldots, x_t, y_1, \ldots, y_t) = f_1(x_1) \prod_{i=2}^{t} f_i(x_i|x_{i-1}) \prod_{i=1}^{t} p(y_i|x_i). \tag{S.4.41}$$

By integrating over the y_i, we have

$$\begin{aligned}
p(x_1, \ldots, x_t) &= \int p(x_1, \ldots, x_t, y_1, \ldots, y_t) \mathrm{d}y_1 \ldots \mathrm{d}y_t && \text{(S.4.42)} \\
&= f_1(x_1) \prod_{i=2}^{t} f_i(x_i|x_{i-1}) \int \prod_{i=1}^{t} p(y_i|x_i) \mathrm{d}y_1 \ldots \mathrm{d}y_t && \text{(S.4.43)} \\
&= f_1(x_1) \prod_{i=2}^{t} f_i(x_i|x_{i-1}) \prod_{i=1}^{t} \underbrace{\int p(y_i|x_i) \mathrm{d}y_i}_{1} && \text{(S.4.44)} \\
&= f_1(x_1) \prod_{i=2}^{t} f_i(x_i|x_{i-1}) && \text{(S.4.45)} \\
&= f(\mathbf{x}) && \text{(S.4.46)}
\end{aligned}$$

(d) Compute the normalising constant for $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$ and express it as an expectation over $f(\mathbf{x})$.

Solution. With

$$p(x_1, \ldots, x_t, y_1, \ldots, y_t) \propto f_1(x_1) \prod_{2=1}^{t} f_i(x_i|x_{i-1}) \prod_{i=1}^{t} \phi_i^y(x_i, y_i). \tag{S.4.47}$$

The normalising constant is given by

$$Z = \int f_1(x_1) \prod_{2=1}^{t} f_i(x_i|x_{i-1}) \prod_{i=1}^{t} g_i(x_i) \mathrm{d}x_1 \ldots \mathrm{d}x_t \tag{S.4.48}$$

$$= \mathbb{E}_f \left[\prod_{i=1}^{t} g_i(x_i) \right] \tag{S.4.49}$$

Since we can use ancestral sampling to sample from f, the above expectation can be easily computed via sampling.

(e) Express the expectation of a test function $h(\mathbf{x})$ with respect to $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$ as a reweighted expectation with respect to $f(\mathbf{x})$.

Solution. By definition, the expectation over a test function $h(\mathbf{x})$ is

$$\mathbb{E}_{p(x_1,\ldots,x_t|y_1,\ldots,y_t)}[h(\mathbf{x})] = \frac{1}{Z} \int h(\mathbf{x}) f_1(x_1) \prod_{2=1}^{t} f(x_i|x_{i-1}) \prod_{i=1}^{t} g_i(x_i) \mathrm{d}x_1 \ldots \mathrm{d}x_t \tag{S.4.50}$$

$$= \frac{\mathbb{E}_f \left[h(\mathbf{x}) \prod_i g_i(x_i) \right]}{\mathbb{E}_f \left[\prod_i g_i(x_i) \right]} \tag{S.4.51}$$

Both the numerator and denominator can be approximated using samples from f.

Since the $g_i(x_i) = \phi_i^y(x_i, y_i)$ involve the observed variables y_i, this has a nice interpretation: We can think we have two models for $\mathbf{x}$: $f(\mathbf{x})$ that does not involve the observations and $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$ that does. Note, however, that unless $\phi_i^y(x_i, y_i)$ is the conditional pdf $p(y_i|x_i)$, $f(\mathbf{x})$ is *not* the marginal $p(x_1, \ldots, x_t)$ that you would obtain by integrating out the y's from the joint model . We can thus generally think it is a base distribution that got "enhanced" by a change of measure in our expression for $p(x_1, \ldots, x_t|y_1, \ldots, y_t)$. If $\phi_i^y(x_i, y_i)$ is the conditional pdf $p(y_i|x_i)$, the change of measure corresponds to going from the prior to the posterior by multiplication with the likelihood (the terms g_i).

From the expression for the expectation, we can see that the "enhancing" leads to a corresponding introduction of weights in the expectation that depend via g_i on the observations. This can be particularly well seen when we approximate the expectation as a sample average over n samples $\mathbf{x}^{(k)} \sim f(\mathbf{x})$:

$$\mathbb{E}_{p(x_1,\ldots,x_t|y_1,\ldots,y_t)}[h(\mathbf{x})] \approx \sum_{k=1}^{n} W^{(k)} h(\mathbf{x}^{(k)}) \tag{S.4.52}$$

$$W^{(k)} = \frac{w^{(k)}}{\sum_{k=1}^{n} w^{(k)}} \tag{S.4.53}$$

$$w^{(k)} = \prod_i g_i(x_i^{(k)}) \tag{S.4.54}$$

where $x_i^{(k)}$ is the i-th dimension of the vector $\mathbf{x}^{(k)}$.

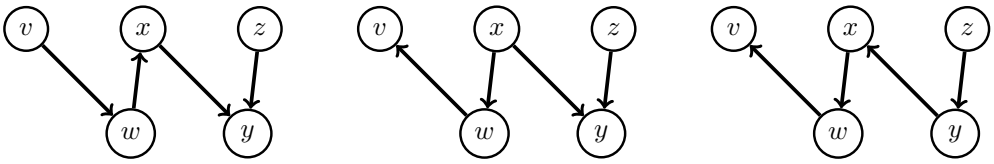
Chapter 5

Expressive Power of Graphical Models

Exercises

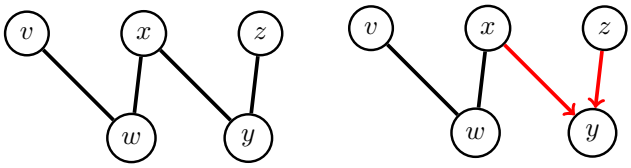
5.1 I-equivalence

(a) Which of three graphs represent the same set of independencies? Explain.



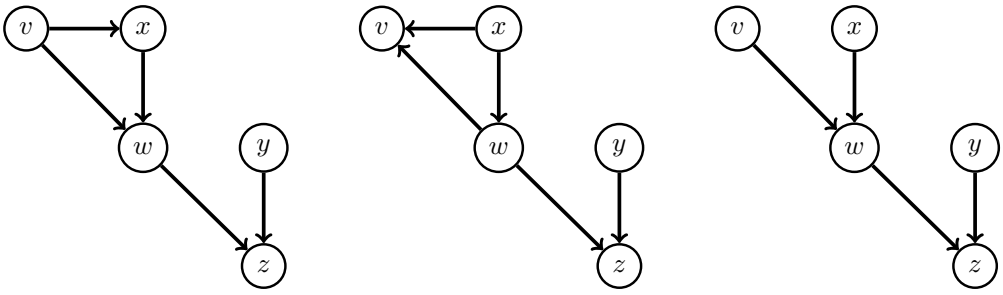
Graph 1 Graph 2 Graph 3

Solution. To check whether the graphs are I-equivalent, we have to check the skeletons and the immoralities. All have the same skeleton, but graph 1 and graph 2 also have the same immorality. The answer is thus: graph 1 and 2 encode the same independencies.



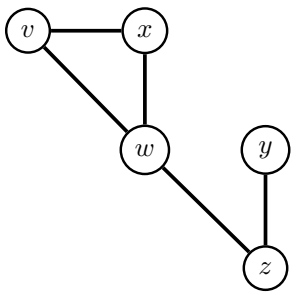
skeleton immorality

(b) Which of three graphs represent the same set of independencies? Explain.

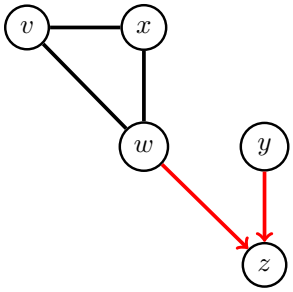


Graph 1 Graph 2 Graph 3

Solution. The skeleton of graph 3 is different from the skeleton of graphs 1 and 2, so that graph 3 cannot be I-equivalent to graph 1 or 2, and we do not need to further check the immoralities for graph 3. Graph 1 and 2 have the same skeleton, and they also have the same immorality. Hence, graph 1 and 2 are I-equivalent. Note that node w in graph 1 is in a collider configuration along trail $v - w - x$ but it is not an immorality because its parents are connected (covering edge); equivalently for node v in graph 2.

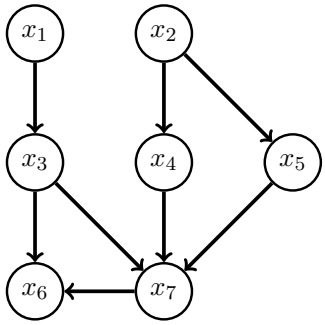


skeleton



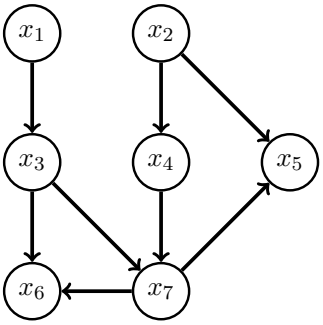
immorality

(c) Assume the graph below is a perfect map for a set of independencies $\mathcal{U}$.

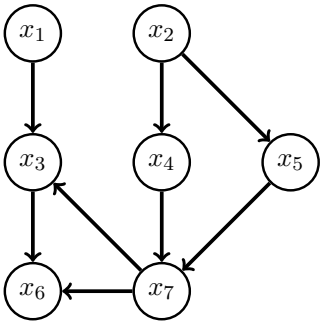


Graph 0

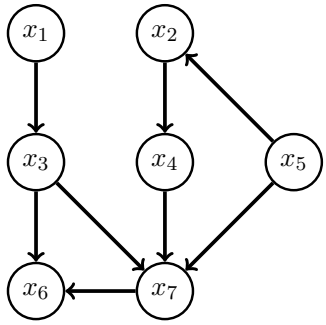
For each of the three graphs below, explain whether the graph is a perfect map, an I-map, or not an I-map for $\mathcal{U}$.



Graph 1



Graph 2



Graph 3

Solution.

- Graph 1 has an immorality $x_2 \to x_5 \leftarrow x_7$ which graph 0 does not have. The graph is thus not I-equivalent to graph 0 and can thus not be a perfect map. Moreover, graph 1 asserts that $x_2 \perp\!\!\!\perp x_7 | x_4$ which is not case for graph 0. Since graph 0 is a perfect map for $\mathcal{U}$, graph 1 asserts an independency that does not hold for $\mathcal{U}$ and can thus not be an I-map for $\mathcal{U}$.
- Graph 2 has an immorality $x_1 \to x_3 \leftarrow x_7$ which graph 0 does not have. Graph 2 thus asserts that $x_1 \perp\!\!\!\perp x_7$, which is not the case for graph 0. Hence, for the same reason as for graph 1, graph 2 is not an I-map for $\mathcal{U}$.
- Graph 3 has the same skeleton and set of immoralities as graph 0. It is thus I-equivalent to graph 0, and hence also a perfect map.

5.2 Minimal I-maps

(a) Assume that the graph G in Figure 5.1 is a perfect I-map for $p(a, z, q, e, h)$. Determine the minimal directed I-map using the ordering (e, h, q, z, a). Is the obtained graph I-equivalent to G?

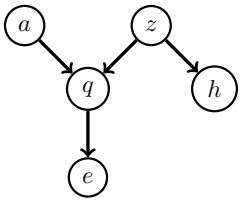


Figure 5.1: Perfect I-map G for Exercise 5.2, question (a).

Solution. Since the graph G is a perfect I-map for p, we can use G to check whether p satisfies a certain independency. This gives the following recipe to construct the minimal directed I-map:

1. Assume an ordering of the variables. Denote the ordered random variables by $x_1, \ldots, x_d$.

2. For each i, find a minimal subset of variables $\pi_i \subseteq \text{pre}_i$ such that

$$x_i \perp\!\!\!\perp \{\text{pre}_i \setminus \pi_i\} \mid \pi_i$$

   is in $\mathcal{I}(G)$ (only works if G is a perfect I-map for $\mathcal{I}(p)$)

3. Construct a graph with parents $\text{pa}_i = \pi_i$.

Note: For I-maps G that are not perfect, if the graph does not indicate that a certain independency holds, we have to check that the independency indeed does not hold for p. If we don't, we won't obtain a minimal I-map but just an I-map for $\mathcal{I}(p)$. This is because p may have independencies that are not encoded in the graph G.

Given the ordering (e, h, q, z, a), we build a graph where e is the root. From Figure 5.1 (and the perfect map assumption), we see that $h \perp\!\!\!\perp e$ does not hold. We thus set e as parent of h, see first graph in Figure 5.2. Then:

- We consider q: $\text{pre}_q = \{e, h\}$. There is no subset π_q of pre_q on which we could condition to make q independent of $\text{pre}_q \setminus \pi_q$, so that we set the parents of q in the graph to $\text{pa}_q = \{e, h\}$. (Second graph in Figure 5.2.)
- We consider z: $\text{pre}_z = \{e, h, q\}$. From the graph in Figure 5.1, we see that for $\pi_z = \{q, h\}$ we have $z \perp\!\!\!\perp \text{pre}_z \setminus \pi_z | \pi_z$. Note that $\pi_z = \{q\}$ does not work because $z \perp\!\!\!\perp e, h | q$ does not hold. We thus set $\text{pa}_z = \{q, h\}$. (Third graph in Figure 5.2.)
- We consider a: $\text{pre}_a = \{e, h, q, z\}$. This is the last node in the ordering. To find the minimal set π_a for which $a \perp\!\!\!\perp \text{pre}_a \setminus \pi_a | \pi_a$, we can determine its Markov blanket $\text{MB}(a)$. The Markov blanket is the set of parents (none), children (q), and co-parents of a (z) in Figure 5.1, so that $\text{MB}(a) = \{q, z\}$. We thus set $\text{pa}_a = \{q, z\}$.(Fourth graph in Figure 5.2.)

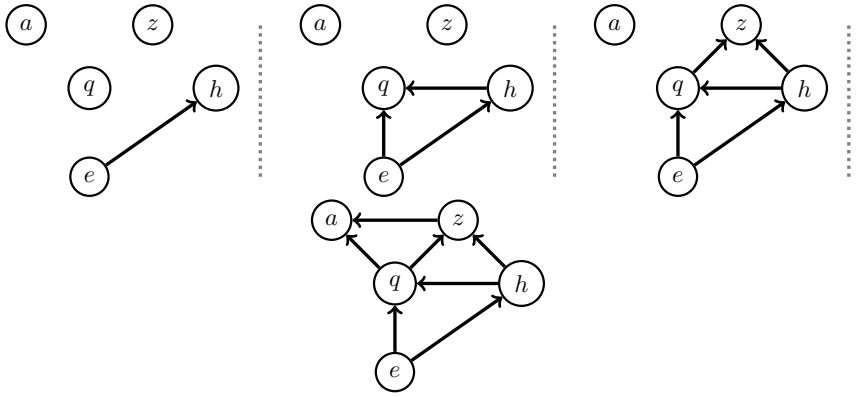


Figure 5.2: Exercise 5.2, Question (a):Construction of a minimal directed I-map for the ordering (e, h, q, z, a).

Since the skeleton in the obtained minimal I-map is different from the skeleton of G, we do not have I-equivalence. Note that the ordering (e, h, q, z, a) yields a denser graph (Figure 5.2) than the graph in Figure 5.1. Whilst a minimal I-map, the graph does e.g. not show that $a \perp\!\!\!\perp z$. Furthermore, the causal interpretation of the two graphs is different.

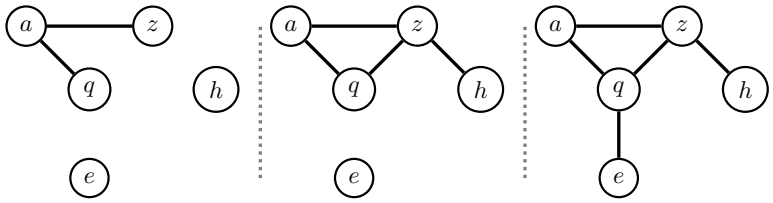
(b) For the collection of random variables (a, z, h, q, e) you are given the following Markov blankets for each variable:

- MB(a) = {q,z}
- MB(z) = {a,q,h}
- MB(h) = {z}
- MB(q) = {a,z,e}
- MB(e) = {q}

(i) Draw the undirected minimal I-map representing the independencies.

(ii) Indicate a Gibbs distribution that satisfies the independence relations specified by the Markov blankets.

Solution. Connecting each variable to all variables in its Markov blanket yields the desired undirected minimal I-map. Note that the Markov blankets are not mutually disjoint.



x,y After MB(a) After MB(z) After MB(q)

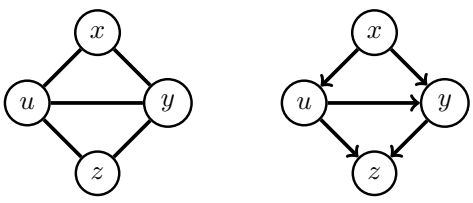
For positive distributions, the set of distributions that satisfy the local Markov property relative to a graph (as given by the Markov blankets) is the same as the set of Gibbs distributions that factorise according to the graph. Given the I-map, we can now easily find the Gibbs distribution

$$p(a, z, h, q, e) = \frac{1}{Z}\phi_1(a, z, q)\phi_2(q, e)\phi_3(z, h),$$

where the ϕ_i must take positive values on their domain. Note that we used the maximal clique (a, z, q).

5.3 I-equivalence between directed and undirected graphs

(a) Verify that the following two graphs are I-equivalent by listing and comparing the independencies that each graph implies.



Solution. First, note that both graphs share the same skeleton and the only reason that they are not fully connected is the missing edge between x and z.

For the DAG, there is also only one ordering that is topological to the graph: x, u, y, z. The missing edge between x and y corresponds to the only independency encoded by the graph: $z \perp\!\!\!\perp \text{pre}_z \setminus \text{pa}_z | \text{pa}_z$, i.e.

$$z \perp\!\!\!\perp x | u, y.$$

This is the same independency that we get from the directed local Markov property.

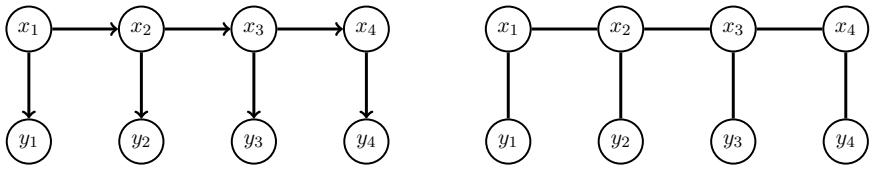
For the undirected graph,

$$z \perp\!\!\!\perp x | u, y$$

holds because u, y block all paths between z and x. All variables but z and x are connected to each other, so that no further independency can hold.

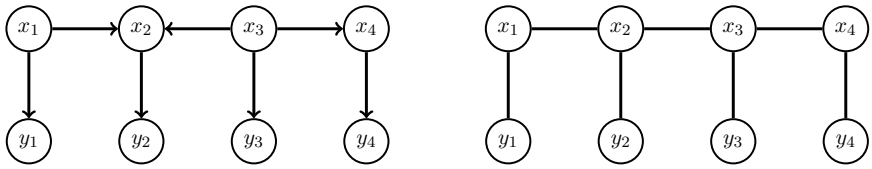
Hence both graphs only encode $z \perp\!\!\!\perp x | u, y$ and they are thus I-equivalent.

(b) Are the following two graphs, which are directed and undirected hidden Markov models, I-equivalent?



Solution. The skeleton of the two graphs is the same and there are no immoralities. Hence, the two graphs are I-equivalent.

(c) Are the following two graphs I-equivalent?



Solution. The two graphs are not I-equivalent because $x_1 - x_2 - x_3$ forms an immorality. Hence, the undirected graph encodes $x_1 \perp\!\!\!\perp x_3 | x_2$ which is not represented in the directed graph. On the other hand, the directed graph asserts $x_1 \perp\!\!\!\perp x_3$ which is not represented in the undirected graph.

5.4 Moralisation: Converting DAGs to undirected minimal I-maps

The following recipe constructs undirected minimal I-maps for $\mathcal{I}(p)$:

- Determine the Markov blanket for each variable x_i

- Construct a graph where the neighbours of x_i are given by its Markov blanket.

We can adapt the recipe to construct an undirected minimal I-map for the independencies $\mathcal{I}(G)$ encoded by a DAG G. What we need to do is to use G to read out the Markov blankets for the variables x_i rather than determining the Markov blankets from the distribution p.

Show that this procedure leads to the following recipe to convert DAGs to undirected minimal I-maps:

1. For all immoralities in the graph: add edges between *all* parents of the collider node.

2. Make all edges in the graph undirected.

The first step is sometimes called "moralisation" because we "marry" all the parents in the graph that are not already directly connected by an edge. The resulting undirected graph is called the moral graph of G, sometimes denoted by $\mathcal{M}(G)$.

Solution. The Markov blanket of a variable x is the set of its parents, children, and co-parents, as shown in the graph below in sub-figure (a). The parents and children are connected to x in the directed graph, but the co-parents are not directly connected to x. Hence, according to "Construct a graph where the neighbours of x_i are its Markov blanket.", we need to introduce edges between x and all its co-parents. This gives the intermediate graph in sub-figure (b).

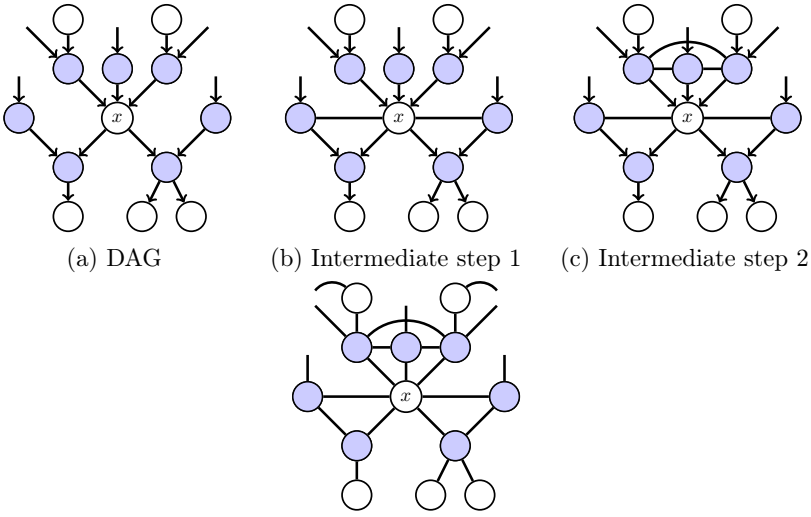
Now, considering the top-left parent of x, we see that for that node, the Markov blanket includes the other parents of x. This means that we need to connect all parents of x, which gives the graph in sub-figure (c). This is sometimes called "marrying" the parents of x. Continuing in this way, we see that we need to "marry" all parents in the graph that are not already married.

Finally, we need to make all edges in the graph undirected, which gives sub-figure (d).

A simpler approach is to note that the DAG specifies the factorisation $p(\mathbf{x}) = \prod_i p(x_i|\mathrm{pa}_i)$. We can consider each conditional $p(x_i|\mathrm{pa}_i)$ to be a factor $\phi_i(x_i, \mathrm{pa}_i)$ so that we obtain the Gibbs distribution $p(\mathbf{x}) = \prod_i \phi_i(x_i|\mathrm{pa}_i)$. Visualising the distribution by connecting all variables in the same factor $\phi_i(x_i|\mathrm{pa}_i)$ leads to the "marriage" of all parents of x_i. This corresponds to the first step in the recipe because x_i is in a collider configuration with respect to the parent nodes. Not all parents form an immorality but this does here not matter because those that do not form an immorality are already connected by a covering edge in the first place.

5.5 Moralisation exercise

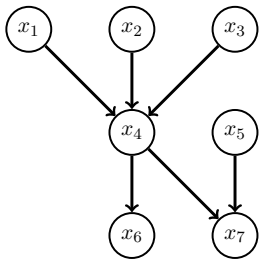
For the DAG G below find the minimal undirected I-map for $\mathcal{I}(G)$.



(a) DAG (b) Intermediate step 1 (c) Intermediate step 2

(d) Undirected graph

Figure 5.3: Answer to Exercise 5.4: Illustrating the moralisation process

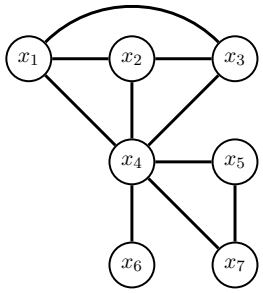


Solution. To derive an undirected minimal I-map from a directed one, we have to construct the moralised graph where the "unmarried" parents are connected by a covering edge. This is because each conditional $p(x_i|\text{pa}_i)$ corresponds to a factor $\phi_i(x_i, \text{pa}_i)$ and we need to connect all variables that are arguments of the same factor with edges.

Statistically, the reason for marrying the parents is as follows: An independency $x \perp\!\!\!\perp y|\{\text{child, other nodes}\}$ does not hold in the directed graph in case of collider connections but would hold in the undirected graph if we didn't marry the parents. Hence links between the parents must be added.

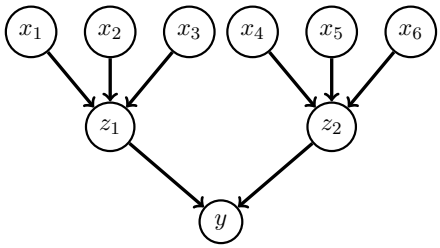
It is important to add edges between *all* parents of a node. Here, $p(x_4|x_1, x_2, x_3)$ corresponds to a factor $\phi(x_4, x_1, x_2, x_3)$ so that all four variables need to be connected. Just adding edges $x_1 - x_2$ and $x_2 - x_3$ would not be enough.

The moral graph, which is the requested minimal undirected I-map, is shown below.

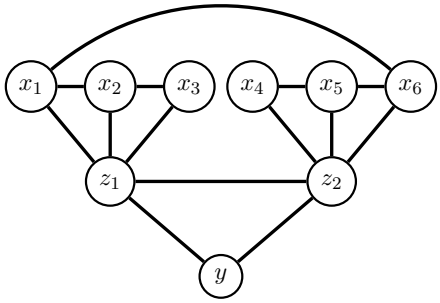


5.6 Moralisation exercise

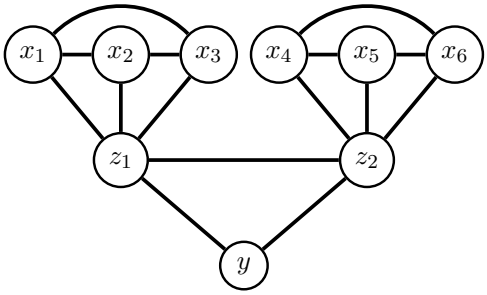
Consider the DAG G:



A friend claims that the undirected graph below is the moral graph $\mathcal{M}(G)$ of G. Is your friend correct? If not, state which edges needed to be removed or added, and explain, in terms of represented independencies, why the changes are necessary for the graph to become the moral graph of G.



Solution. The moral graph $\mathcal{M}(G)$ is an undirected minimal I-map of the independencies represented by G. Following the procedure of connecting "unmarried" parents of colliders, we obtain the following moral graph of G:



We can thus see that the friend's undirected graph is not the moral graph of G.

The edge between x_1 and x_6 can be removed. This is because for G, we have e.g. the independencies $x_1 \perp\!\!\!\perp x_6|z_1$, $x_1 \perp\!\!\!\perp x_6|z_2$, $x_1 \perp\!\!\!\perp x_6|z_1, z_2$ which is not represented by the drawn undirected graph.

We need to add edges between x_1 and x_3, and between x_4 and x_6. Otherwise, the undirected graph makes the wrong independency assertion that $x_1 \perp\!\!\!\perp x_3|x_2, z_1$ (and equivalent for x_4 and x_6).

5.7 Triangulation: Converting undirected graphs to directed minimal I-maps

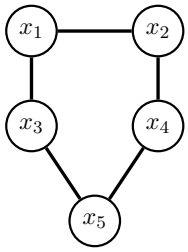
In Exercise 5.4 we adapted a recipe for constructing undirected minimal I-maps for $\mathcal{I}(p)$ to the case of $\mathcal{I}(G)$, where G is a DAG. The key difference was that we used the graph G to determine independencies rather than the distribution p.

We can similarly adapt the recipe for constructing a directed minimal I-map for $\mathcal{I}(p)$ to build a directed minimal I-map for $\mathcal{I}(H)$, where H is an undirected graph:

1. Choose an ordering of the random variables.
2. For all variables x_i, use H to determine a *minimal* subset π_i of the predecessors pre_i such that
$$x_i \perp\!\!\!\perp (\text{pre}_i \setminus \pi_i) \mid \pi_i$$
holds.
3. Construct a DAG with the π_i as parents pa_i of x_i.

Remarks: (1) Directed minimal I-maps obtained with different orderings are generally not I-equivalent. (2) The directed minimal I-maps obtained with the above method are always chordal graphs. Chordal graphs are graphs where the longest trail without shortcuts is a triangle (https://en.wikipedia.org/wiki/Chordal_graph). They are thus also called triangulated graphs. We obtain chordal graphs because if we had trails without shortcuts that involved more than 3 nodes, we would necessarily have an immorality in the graph. But immoralities encode independencies that an undirected graph cannot represent, which would make the DAG not an I-map for $\mathcal{I}(H)$ any more.

(a) Let H be the undirected graph below. Determine the directed minimal I-map for $\mathcal{I}(H)$ with the variable ordering x_1, x_2, x_3, x_4, x_5.



Solution. We use the ordering x_1, x_2, x_3, x_4, x_5 and follow the conversion procedure:

- x_2 is not independent from x_1 so that we set $\text{pa}_2 = \{x_1\}$. See first graph in Figure 5.4.
- Since x_3 is connected to both x_1 and x_2, we don't have $x_3 \perp\!\!\!\perp x_2, x_1$. We cannot make x_3 independent from x_2 by conditioning on x_1 because there are two paths from x_3 to x_2 and x_1 only blocks the upper one. Moreover, x_1 is a neighbour of x_3 so that conditioning on x_2 does make them independent. Hence we must set $\text{pa}_3 = \{x_1, x_2\}$. See second graph in Figure 5.4.
- For x_4, we see from the undirected graph, that $x_4 \perp\!\!\!\perp x_1 \mid x_3, x_2$. The graph further shows that removing either x_3 or x_2 from the conditioning set is not possible and conditioning on x_1 won't make x_4 independent from x_2 or x_3. We thus have $\text{pa}_4 = \{x_2, x_3\}$. See fourth graph in Figure 5.4.
- The same reasoning shows that $\text{pa}_5 = \{x_3, x_4\}$. See last graph in Figure 5.4.

This results in the triangulated directed graph in Figure 5.4 on the right.

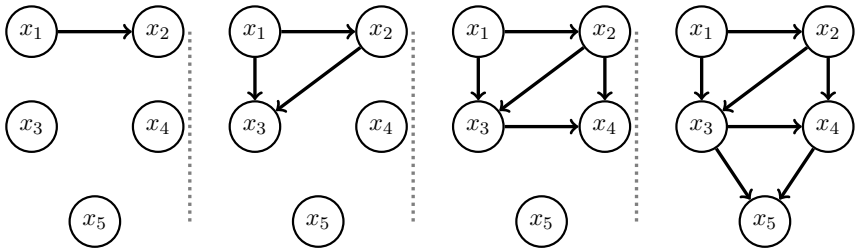


Figure 5.4: . Answer to Exercise 5.7, Question (a).

To see why triangulation is necessary consider the case where we didn't have the edge between x_2 and x_3 as in Figure 5.5. The directed graph would then imply that $x_3 \perp\!\!\!\perp x_2 \mid x_1$ (check!). But this independency assertion does not hold in the undirected graph so that the graph in Figure 5.5 is not an I-map.

(b) For the undirected graph from question (a) above, which variable ordering yields the directed minimal I-map below?

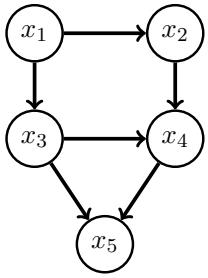
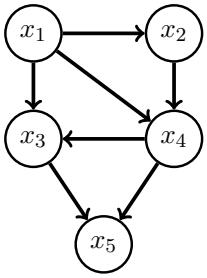


Figure 5.5: Not a directed I-map for the undirected graphical model defined by the graph in Exercise 5.7, Question (a).



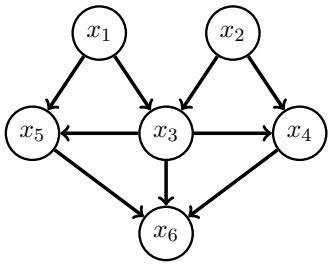
Solution. x_1 is the root of the DAG, so it comes first. Next in the ordering are the children of x_1: x_2, x_3, x_4. Since x_3 is a child of x_4, and x_4 a child of x_2, we must have x_1, x_2, x_4, x_3. Furthermore, x_3 must come before x_5 in the ordering since x_5 is a child of x_3, hence the ordering used must have been: x_1, x_2, x_4, x_3, x_5.

5.8 I-maps, minimal I-maps, and I-equivalency

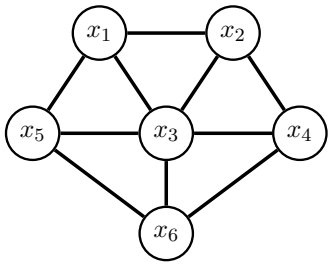
Consider the following probability density function for random variables $x_1, \ldots, x_6$.

$$p_a(x_1, \ldots, x_6) = p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_2)p(x_5|x_1)p(x_6|x_3, x_4, x_5)$$

For each of the two graphs below, explain whether it is a minimal I-map, not a minimal I-map but still an I-map, or not an I-map for the independencies that hold for p_a.

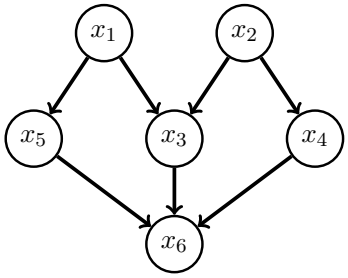


graph 1



graph 2

Solution. The pdf can be visualised as the following directed graph, which is a minimal I-map for it.



Graph 1 defines distributions that factorise as

$$p_b(\mathbf{x}) = p(x_1)p(x_2)p(x_3|x_1,x_2)p(x_4|x_2,x_3)p(x_5|x_1,x_3)p(x_6|x_3,x_4,x_5). \tag{S.5.1}$$

Comparing with $p_a(x_1,\ldots,x_6)$, we see that only the conditionals $p(x_4|x_2,x_3)$ and $p(x_5|x_1,x_3)$ are different. Specifically, their conditioning set includes x_3, which means that Graph 1 encodes fewer independencies than what $p_a(x_1,\ldots,x_6)$ satisfies. In particular $x_4 \perp\!\!\!\perp x_3|x_2$ and $x_5 \perp\!\!\!\perp x_3|x_1$ are not represented in the graph. This means that we could remove x_3 from the conditioning sets, or equivalently remove the edges $x_3 \to x_4$ and $x_3 \to x_5$ from the graph without introducing independence assertions that do not hold for p_a. This means graph 1 is an I-map but not a minimal I-map.

Graph 2 is not an I-map. To be an undirected minimal I-map, we had to connect variables x_5 and x_4 that are parents of x_6. Graph 2 wrongly claims that $x_5 \perp\!\!\!\perp x_4 \mid x_1, x_3, x_6$.

5.9 Limits of directed and undirected graphical models

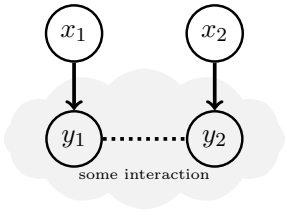
We here consider the probabilistic model $p(y_1,y_2,x_1,x_2) = p(y_1,y_2|x_1,x_2)p(x_1)p(x_2)$ where $p(y_1,y_2|x_1,x_2)$ factorises as

$$p(y_1,y_2|x_1,x_2) = p(y_1|x_1)p(y_2|x_2)\phi(y_1,y_2)n(x_1,x_2) \tag{5.1}$$

with $n(x_1,x_2)$ equal to

$$n(x_1,x_2) = \left(\int p(y_1|x_1)p(y_2|x_2)\phi(y_1,y_2)\mathrm{d}y_1\mathrm{d}y_2\right)^{-1}. \tag{5.2}$$

In the model, x_1 and x_2 are two independent inputs that each control the interacting variables y_1 and y_2 (see graph below). However, the nature of the interaction between y_1 and y_2 is not modelled. In particular, we do not assume a directionality, i.e. $y_1 \to y_2$, or $y_2 \to y_1$.



(a) Use the basic characterisations of statistical independence

$$u \perp\!\!\!\perp v|z \iff p(u,v|z) = p(u|z)p(v|z) \tag{5.3}$$
$$u \perp\!\!\!\perp v|z \iff p(u,v|z) = a(u,z)b(v,z) \qquad (a(u,z) \geq 0, b(v,z) \geq 0) \tag{5.4}$$

to show that $p(y_1, y_2, x_1, x_2)$ satisfies the following independencies

$$x_1 \perp\!\!\!\perp x_2 \qquad x_1 \perp\!\!\!\perp y_2 \mid y_1, x_2 \qquad x_2 \perp\!\!\!\perp y_1 \mid y_2, x_1$$

Solution. The pdf/pmf is

$$p(y_1, y_2, x_1, x_2) = p(y_1|x_1)p(y_2|x_2)\phi(y_1, y_2)n(x_1, x_2)p(x_1)p(x_2)$$

For $\mathbf{x_1 \perp\!\!\!\perp x_2}$
We compute $p(x_1, x_2)$ as

$$p(x_1, x_2) = \int p(y_1, y_2, x_1, x_2)\mathrm{d}y_1\mathrm{d}y_2 \tag{S.5.2}$$
$$= \int p(y_1|x_1)p(y_2|x_2)\phi(y_1, y_2)n(x_1, x_2)p(x_1)p(x_2)\mathrm{d}y_1\mathrm{d}y_2 \tag{S.5.3}$$
$$= n(x_1, x_2)p(x_1)p(x_2) \int p(y_1|x_1)p(y_2|x_2)\phi(y_1, y_2)\mathrm{d}y_1\mathrm{d}y_2 \tag{S.5.4}$$
$$\overset{(5.2)}{=} n(x_1, x_2)p(x_1)p(x_2)\frac{1}{n(x_1, x_2)} \tag{S.5.5}$$
$$= p(x_1)p(x_2). \tag{S.5.6}$$

Since $p(x_1)$ and $p(x_2)$ are the univariate marginals of x_1 and x_2, respectively, it follows from (5.3) that $x_1 \perp\!\!\!\perp x_2$.

For $\mathbf{x_1 \perp\!\!\!\perp y_2 \mid y_1, x_2}$
We rewrite $p(y_1, y_2, x_1, x_2)$ as

$$p(y_1, y_2, x_1, x_2) = p(y_1|x_1)p(y_2|x_2)\phi(y_1, y_2)n(x_1, x_2)p(x_1)p(x_2) \tag{S.5.7}$$
$$= [p(y_1|x_1)p(x_1)n(x_1, x_2)]\,[p(y_2|x_2)\phi(y_1, y_2)p(x_2)] \tag{S.5.8}$$
$$= \phi_A(x_1, y_1, x_2)\phi_B(y_2, y_1, x_2) \tag{S.5.9}$$

With (5.4), we have that $x_1 \perp\!\!\!\perp y_2 \mid y_1, x_2$. Note that $p(x_2)$ can be associated either with ϕ_A or with ϕ_B.

For $\mathbf{x_2 \perp\!\!\!\perp y_1 \mid y_2, x_1}$
We use here the same approach as for $x_1 \perp\!\!\!\perp y_2 \mid y_1, x_2$. (By symmetry considerations, we could immediately see that the relation holds but let us write it out for clarity). We rewrite $p(y_1, y_2, x_1, x_2)$ as

$$p(y_1, y_2, x_1, x_2) = p(y_1|x_1)p(y_2|x_2)\phi(y_1, y_2)n(x_1, x_2)p(x_1)p(x_2) \tag{S.5.10}$$
$$= [p(y_2|x_2)n(x_1, x_2)p(x_2)p(x_1))]\,[p(y_1|x_1)\phi(y_1, y_2)]) \tag{S.5.11}$$
$$= \tilde{\phi}_A(x_2, x_1, y_2)\tilde{\phi}_B(y_1, y_2, x_1) \tag{S.5.12}$$

With (5.4), we have that $x_2 \perp\!\!\!\perp y_1 \mid y_2, x_1$.

(b) Is there an undirected perfect map for the independencies satisfied by $p(y_1, y_2, x_1, x_2)$?

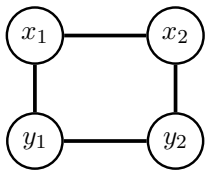
Solution. We write

$$p(y_1, y_2, x_1, x_2) = p(y_1|x_1)p(y_2|x_2)\phi(y_1, y_2)n(x_1, x_2)p(x_1)p(x_2)$$

as a Gibbs distribution

$$p(y_1, y_2, x_1, x_2) = \phi_1(y_1, x_1)\phi_2(y_2, x_2)\phi_3(y_1, y_2)\phi_4(x_1, x_2) \quad \text{with} \tag{S.5.13}$$
$$\phi_1(y_1, x_1) = p(y_1|x_1)p(x_1) \tag{S.5.14}$$
$$\phi_2(y_2, x_2) = p(y_2|x_2)p(x_2) \tag{S.5.15}$$
$$\phi_3(y_1, y_2) = \phi(y_1, y_2) \tag{S.5.16}$$
$$\phi_4(x_1, x_2) = n(x_1, x_2). \tag{S.5.17}$$

Visualising it as an undirected graph gives an I-map:



While the graph implies $x_1 \perp\!\!\!\perp y_2 \mid y_1, x_2$ and $x_2 \perp\!\!\!\perp y_1 \mid y_2, x_1$, the independency $x_1 \perp\!\!\!\perp x_2$ is not represented. Hence the graph is not a perfect map. Note further that removing any edge would result in a graph that is not an I-map for $\mathcal{I}(p)$ anymore. Hence the graph is a minimal I-map for $\mathcal{I}(p)$ but that we cannot obtain a perfect I-map.

(c) Is there a directed perfect map for the independencies satisfied by $p(y_1, y_2, x_1, x_2)$?

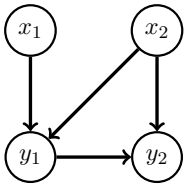
Solution. We construct directed minimal I-maps for $p(y_1, y_2, x_1, x_2) = p(y_1, y_2|x_1, x_2)p(x_1)p(x_2)$ for different orderings. We will see that they do not represent all independencies in $\mathcal{I}(p)$ and hence that they are not perfect I-maps.

To guarantee unconditional independence of x_1 and x_2, the two variables must come first in the orderings (either x_1 and then x_2 or the other way around).

If we use the ordering x_1, x_2, y_1, y_2, and that

- $x_1 \perp\!\!\!\perp x_2$
- $y_2 \perp\!\!\!\perp x_1|y_1, x_2$, which is $y_2 \perp\!\!\!\perp \text{pre}(y_2) \setminus \pi|\pi$ for $\pi = (y_1, x_2)$

are in $\mathcal{I}(p)$, we obtain the following directed minimal I-map:

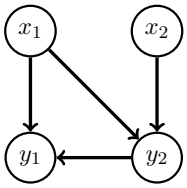


The graphs misses $x_2 \perp\!\!\!\perp y_1 \mid y_2, x_1$.

If we use the ordering x_1, x_2, y_2, y_1, and that

- $x_1 \perp\!\!\!\perp x_2$
- $y_1 \perp\!\!\!\perp x_2 | x_1, y_2$, which is $y_1 \perp\!\!\!\perp \text{pre}(y_1) \setminus \pi | \pi$ for $\pi = (x_1, y_2)$

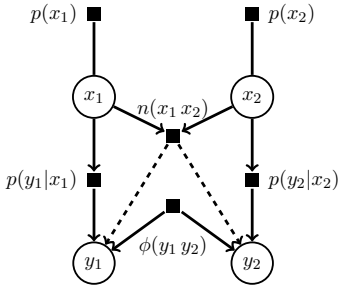
are in $\mathcal{I}(p)$, we obtain the following directed minimal I-map:



The graph misses $x_1 \perp\!\!\!\perp y_2 \mid y_1, x_2$.

Moreover, the graphs imply a directionality between y_1 and y_2, or a direct influence of x_1 on y_2, or of x_2 on y_1, in contrast to the original modelling goals.

(d) *(advanced)* The following factor graph represents $p(y_1, y_2, x_1, x_2)$:



Use the separation rules for factor graphs to verify that we can find all independence relations. The separation rules are (see Barber, 2012, Section 4.4.1), or the original paper by Frey (2003):

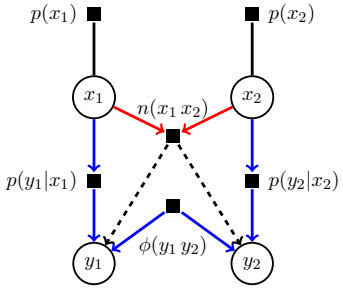
"If all paths are blocked, the variables are conditionally independent. A path is blocked if one or more of the following conditions is satisfied:

1. One of the variables in the path is in the conditioning set.
2. One of the variables or factors in the path has two incoming edges that are part of the path (variable or factor collider), and neither the variable or factor nor any of its descendants are in the conditioning set."

Remarks:

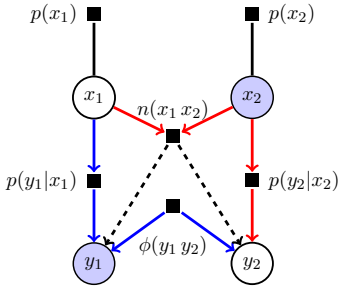
- "one or more of the following" should best be read as "one of the following".
- "incoming edges" means directed incoming edges
- the descendants of a variable or factor node are all the variables that you can reach by following a path (containing directed or directed edges, but for directed edges, all directions have to be consistent)
- In the graph we have dashed directed edges: they do count when you determine the descendants but they do not contribute to paths. For example, y_1 is a descendant of the $n(x_1, x_2)$ factor node but $x_1 - n - y_2$ is not a path.

Solution. $\mathbf{x_1 \perp\!\!\!\perp x_2}$
There are two paths from x_1 to x_2 marked with red and blue below:



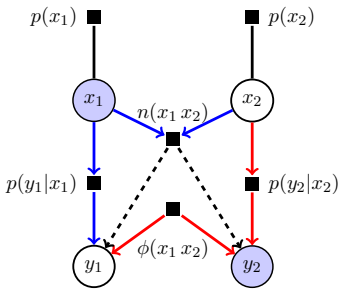
Both the blue and red path are blocked by condition 2.

$\mathbf{x_1 \perp\!\!\!\perp y_2 \mid y_1, x_2}$
There are two paths from x_1 to y_2 marked with red and blue below:



The observed variables are marked in blue. For the red path, the observed x_2 blocks the path (condition 1). Note that the $n(x_1, x_2)$ node would be open by condition 2. The blue path is blocked by condition 1 too. In directed graphical models, the y_1 node would be open, but here while condition 2 does not apply, condition 1 still applies (note the *one or more of ...* in the separation rules), so that the path is blocked.

$\mathbf{x_2 \perp\!\!\!\perp y_1 \mid y_2, x_1}$
There are two paths from x_2 to y_1 marked with red and blue below:



The same reasoning as before yields the result.

Finally note that x_1 and x_2 are not independent given y_1 or y_2 because the upper path through $n(x_1, x_2)$ is not blocked whenever y_1 or y_2 are observed (condition 2).

Credit: this example is discussed in the original paper by B. Frey (Figure 6).

Chapter 6

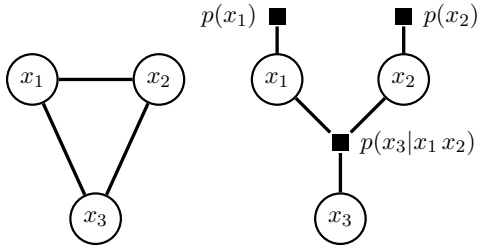
Factor Graphs and Message Passing

Exercises

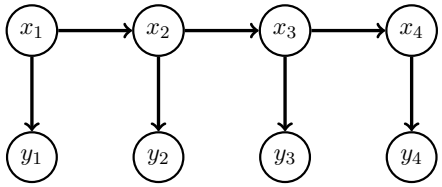
6.1 Conversion to factor graphs

(a) Draw an undirected graph and an undirected factor graph for $p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2)$

Solution.



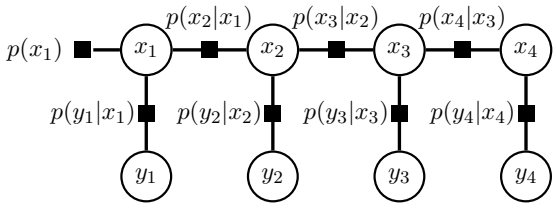
(b) Draw an undirected factor graph for the directed graphical model defined by the graph below.



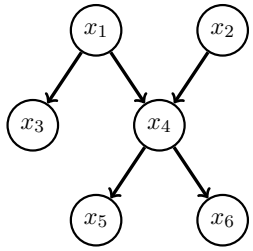
Solution. The graph specifies probabilistic models that factorise as

$$p(x_1, \ldots, x_4, y_1, \ldots, y_4) = p(x_1)p(y_1|x_1) \prod_{i=2}^{4} p(y_i|x_i)p(x_i|x_{i-1})$$

It is the graph for a hidden Markov model. The corresponding factor graph is shown below.



(c) Draw the moralised graph and an undirected factor graph for directed graphical models defined by the graph below (this kind of graph is called a polytree: there are no loops but a node may have more than one parent).

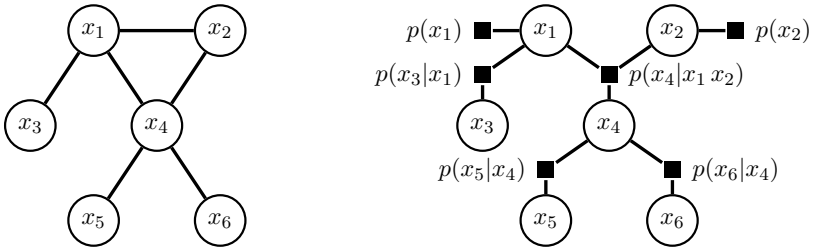


Solution. The moral graph is obtained by connecting the parents of the collider node x_4. See the graph on the left in the figure below.

For the factor graph, we note that the directed graph defines the following class of probabilistic models

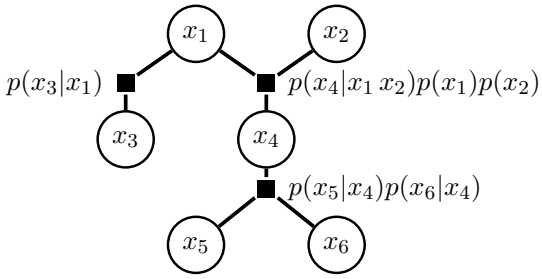
$$p(x_1, \dots x_6) = p(x_1)p(x_2)p(x_3|x_1)p(x_4|x_1, x_2)p(x_5|x_4)p(x_6|x_4)$$

This gives the factor graph on right in the figure below.



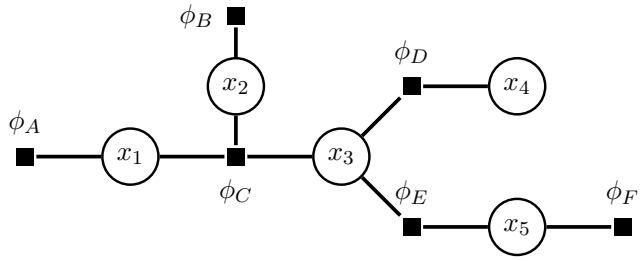
Note:

- The moral graph contains a loop while the factor graph does not. The factor graph is still a polytree. This can be exploited for inference.
- One may choose to group some factors together in order to obtain a factor graph with a particular structure (see factor graph below)



6.2 Sum-product message passing

We here consider the following factor tree:



Let all variables be binary, $x_i \in \{0, 1\}$, and the factors be defined as follows:

x_1	ϕ_A
0	2
1	4

x_2	ϕ_B
0	4
1	4

x_1	x_2	x_3	ϕ_C
0	0	0	4
1	0	0	2
0	1	0	2
1	1	0	6
0	0	1	2
1	0	1	6
0	1	1	6
1	1	1	4

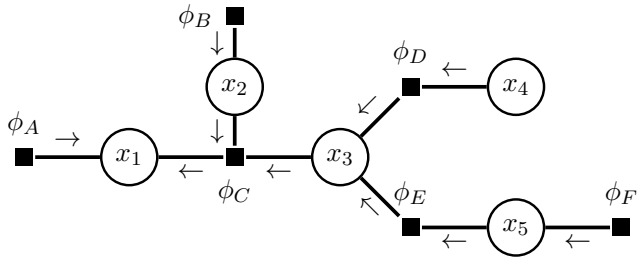
x_3	x_4	ϕ_D
0	0	8
1	0	2
0	1	2
1	1	6

x_3	x_5	ϕ_E
0	0	3
1	0	6
0	1	6
1	1	3

x_5	ϕ_F
0	1
1	8

(a) Mark the graph with arrows indicating all messages that need to be computed for the computation of $p(x_1)$.

Solution.



(b) Compute the messages that you have identified.

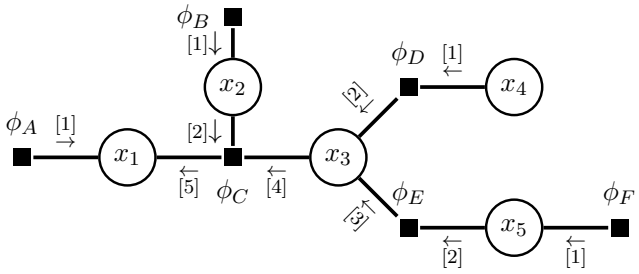
Assuming that the computation of the messages is scheduled according to a common clock, group the messages together so that all messages in the same group can be computed in parallel during a clock cycle.

Solution. Since the variables are binary, each message can be represented as a two-dimensional vector. We use the convention that the first element of the vector corresponds to the message for $x_i = 0$ and the second element to the message for $x_i = 1$. For example,

$$\boldsymbol{\mu_{\phi_A \to x_1}} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \tag{S.6.1}$$

means that the message $\mu_{\phi_A \to x_1}(x_1)$ equals 2 for $x_1 = 0$, i.e. $\mu_{\phi_A \to x_1}(0) = 2$.

The following figure shows a grouping (scheduling) of the computation of the messages.



Clock cycle 1:

$$\boldsymbol{\mu_{\phi_A \to x_1}} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \quad \boldsymbol{\mu_{\phi_B \to x_2}} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} \quad \boldsymbol{\mu_{x_4 \to \phi_D}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \boldsymbol{\mu_{\phi_F \to x_5}} = \begin{pmatrix} 1 \\ 8 \end{pmatrix} \tag{S.6.2}$$

Clock cycle 2:

$$\boldsymbol{\mu_{x_2 \to \phi_C}} = \boldsymbol{\mu_{\phi_B \to x_2}} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} \qquad \boldsymbol{\mu_{x_5 \to \phi_E}} = \boldsymbol{\mu_{\phi_F \to x_5}} = \begin{pmatrix} 1 \\ 8 \end{pmatrix} \tag{S.6.3}$$

Message $\mu_{\phi_D \to x_3}$ is defined as

$$\mu_{\phi_D \to x_3}(x_3) = \sum_{x_4} \phi_D(x_3, x_4) \mu_{x_4 \to \phi_D}(x_4) \tag{S.6.4}$$

so that

$$\begin{aligned}
\mu_{\phi_D \to x_3}(0) &= \sum_{x_4=0}^{1} \phi_D(0, x_4) \mu_{x_4 \to \phi_D}(x_4) && \text{(S.6.5)} \\
&= \phi_D(0,0) \mu_{x_4 \to \phi_D}(0) + \phi_D(0,1) \mu_{x_4 \to \phi_D}(1) && \text{(S.6.6)} \\
&= 8 \cdot 1 + 2 \cdot 1 && \text{(S.6.7)} \\
&= 10 && \text{(S.6.8)} \\
\mu_{\phi_D \to x_3}(1) &= \sum_{x_4=0}^{1} \phi_D(1, x_4) \mu_{x_4 \to \phi_D}(x_4) && \text{(S.6.9)} \\
&= \phi_D(1,0) \mu_{x_4 \to \phi_D}(0) + \phi_D(1,1) \mu_{x_4 \to \phi_D}(1) && \text{(S.6.10)} \\
&= 2 \cdot 1 + 6 \cdot 1 && \text{(S.6.11)} \\
&= 8 && \text{(S.6.12)}
\end{aligned}$$

and thus

$$\boldsymbol{\mu_{\phi_D \to x_3}} = \begin{pmatrix} 10 \\ 8 \end{pmatrix}. \tag{S.6.13}$$

The above computations can be written more compactly in matrix notation. Let $\boldsymbol{\phi_D}$ be the matrix that contains the outputs of $\phi_D(x_3, x_4)$

$$\boldsymbol{\phi_D} = \begin{pmatrix} \phi_D(x_3 = 0, x_4 = 0) & \phi_D(x_3 = 0, x_4 = 1) \\ \phi_D(x_3 = 1, x_4 = 0) & \phi_D(x_3 = 1, x_4 = 1) \end{pmatrix} = \begin{pmatrix} 8 & 2 \\ 2 & 6 \end{pmatrix}. \tag{S.6.14}$$

We can then write $\boldsymbol{\mu_{\phi_D \to x_3}}$ in terms of a matrix vector product,

$$\boldsymbol{\mu_{\phi_D \to x_3}} = \boldsymbol{\phi_D \mu_{x_4 \to \phi_D}}. \tag{S.6.15}$$

Clock cycle 3:
Representing the factor ϕ_E as matrix $\boldsymbol{\phi_E}$,

$$\boldsymbol{\phi_E} = \begin{pmatrix} \phi_E(x_3 = 0, x_5 = 0) & \phi_E(x_3 = 0, x_5 = 1) \\ \phi_E(x_3 = 1, x_5 = 0) & \phi_E(x_3 = 1, x_5 = 1) \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ 6 & 3 \end{pmatrix}, \tag{S.6.16}$$

we can write

$$\mu_{\phi_E \to x_3}(x_3) = \sum_{x_5} \phi_E(x_3, x_5) \mu_{x_5 \to \phi_E}(x_5) \tag{S.6.17}$$

as a matrix vector product,

$$\begin{aligned} \boldsymbol{\mu_{\phi_E \to x_3}} &= \boldsymbol{\phi_E \mu_{x_5 \to \phi_E}} && \text{(S.6.18)} \\ &= \begin{pmatrix} 3 & 6 \\ 6 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 8 \end{pmatrix} && \text{(S.6.19)} \\ &= \begin{pmatrix} 51 \\ 30 \end{pmatrix}. && \text{(S.6.20)} \end{aligned}$$

Clock cycle 4:
Variable node x_3 has received all incoming messages, and can thus output $\mu_{x_3 \to \phi_C}$,

$$\mu_{x_3 \to \phi_C}(x_3) = \mu_{\phi_D \to x_3}(x_3) \mu_{\phi_E \to x_3}(x_3). \tag{S.6.21}$$

Using $\odot$ to denote element-wise multiplication of two vectors, we have

$$\begin{aligned} \boldsymbol{\mu_{x_3 \to \phi_C}} &= \boldsymbol{\mu_{\phi_D \to x_3}} \odot \boldsymbol{\mu_{\phi_E \to x_3}} && \text{(S.6.22)} \\ &= \begin{pmatrix} 10 \\ 8 \end{pmatrix} \odot \begin{pmatrix} 51 \\ 30 \end{pmatrix} && \text{(S.6.23)} \\ &= \begin{pmatrix} 510 \\ 240 \end{pmatrix}. && \text{(S.6.24)} \end{aligned}$$

Clock cycle 5:
Factor node ϕ_C has received all incoming messages, and can thus output $\mu_{\phi_C \to x_1}$,

$$\mu_{\phi_C \to x_1}(x_1) = \sum_{x_2, x_3} \phi_C(x_1, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3). \tag{S.6.25}$$

Writing out the sum for $x_1 = 0$ and $x_1 = 1$ gives

$$\begin{aligned}
\mu_{\phi_C \to x_1}(0) &= \sum_{x_2,x_3} \phi_C(0,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3) && \text{(S.6.26)}\\
&= \phi_C(0,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(0,0)} + && \text{(S.6.27)}\\
&\quad \phi_C(0,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(1,0)} + && \text{(S.6.28)}\\
&\quad \phi_C(0,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(0,1)} + && \text{(S.6.29)}\\
&\quad \phi_C(0,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(1,1)} && \text{(S.6.30)}\\
&= 4\cdot 4\cdot 510 + && \text{(S.6.31)}\\
&\quad 2\cdot 4\cdot 510 + && \text{(S.6.32)}\\
&\quad 2\cdot 4\cdot 240 + && \text{(S.6.33)}\\
&\quad 6\cdot 4\cdot 240 && \text{(S.6.34)}\\
&= 19920 && \text{(S.6.35)}\\
\mu_{\phi_C \to x_1}(1) &= \sum_{x_2,x_3} \phi_C(1,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3) && \text{(S.6.36)}\\
&= \phi_C(1,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(0,0)} + && \text{(S.6.37)}\\
&\quad \phi_C(1,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(1,0)} + && \text{(S.6.38)}\\
&\quad \phi_C(1,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(0,1)} + && \text{(S.6.39)}\\
&\quad \phi_C(1,x_2,x_3)\mu_{x_2\to\phi_C}(x_2)\mu_{x_3\to\phi_C}(x_3)\,|_{(x_2,x_3)=(1,1)} && \text{(S.6.40)}\\
&= 2\cdot 4\cdot 510 + && \text{(S.6.41)}\\
&\quad 6\cdot 4\cdot 510 + && \text{(S.6.42)}\\
&\quad 6\cdot 4\cdot 240 + && \text{(S.6.43)}\\
&\quad 4\cdot 4\cdot 240 && \text{(S.6.44)}\\
&= 25920 && \text{(S.6.45)}
\end{aligned}$$

and hence

$$\boldsymbol{\mu_{\phi_C \to x_1}} = \begin{pmatrix} 19920 \\ 25920 \end{pmatrix} \tag{S.6.46}$$

After step 5, variable node x_1 has received all incoming messages and the marginal can be computed.

In addition to the messages needed for computation of $p(x_1)$ one can compute *all* messages in the graph in five clock cycles, see Figure 6.1. This means that *all* marginals, as well as the joints of those variables sharing a factor node, are available after five clock cycles.

(c) What is $p(x_1 = 1)$?

Solution. We compute the marginal $p(x_1)$ as

$$p(x_1) \propto \mu_{\phi_A \to x_1}(x_1)\mu_{\phi_C \to x_1}(x_1) \tag{S.6.47}$$

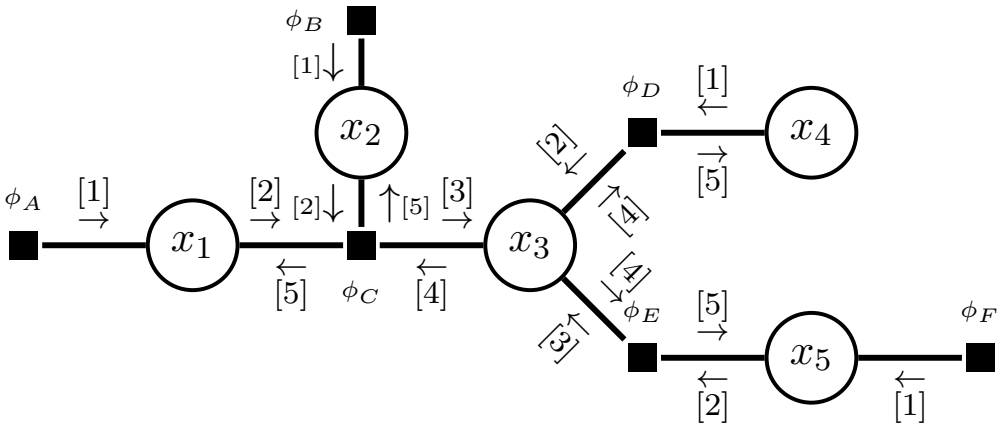


Figure 6.1: Answer to Exercise 6.2 Question (b): Computing all messages in five clock cycles. If we also computed the messages toward the leaf factor nodes, we needed six cycles, but they are not necessary for computation of the marginals so they are omitted.

which is in vector notation

$$\begin{pmatrix} p(x_1=0) \\ p(x_1=1) \end{pmatrix} \propto \boldsymbol{\mu_{\phi_A \to x_1}} \odot \boldsymbol{\mu_{\phi_C \to x_1}} \tag{S.6.48}$$

$$\propto \begin{pmatrix} 2 \\ 4 \end{pmatrix} \odot \begin{pmatrix} 19920 \\ 25920 \end{pmatrix} \tag{S.6.49}$$

$$\propto \begin{pmatrix} 39840 \\ 103680 \end{pmatrix}. \tag{S.6.50}$$

Normalisation gives

$$\begin{pmatrix} p(x_1=0) \\ p(x_1=1) \end{pmatrix} = \frac{1}{39840+103680} \begin{pmatrix} 39840 \\ 103680 \end{pmatrix} \tag{S.6.51}$$

$$= \begin{pmatrix} 0.2776 \\ 0.7224 \end{pmatrix} \tag{S.6.52}$$

so that $p(x_1 = 1) = 0.7224$.

Note the relatively large numbers in the messages that we computed. In other cases, one may obtain very small ones depending on the scale of the factors. This can cause numerical issues that can be addressed by working in the logarithmic domain.

(d) Draw the factor graph corresponding to $p(x_1, x_3, x_4, x_5|x_2 = 1)$ and provide the numerical values for all factors.

Solution. The pmf represented by the original factor graph is

$$p(x_1, \ldots, x_5) \propto \phi_A(x_1)\phi_B(x_2)\phi_C(x_1, x_2, x_3)\phi_D(x_3, x_4)\phi_E(x_3, x_5)\phi_F(x_5)$$

The conditional $p(x_1, x_3, x_4, x_5|x_2 = 1)$ is proportional to $p(x_1, \ldots, x_5)$ with x_2 fixed to $x_2 = 1$, i.e.

$$p(x_1, x_3, x_4, x_5|x_2 = 1) \propto p(x_1, x_2 = 1, x_3, x_4, x_5) \tag{S.6.53}$$

$$\propto \phi_A(x_1)\phi_B(x_2 = 1)\phi_C(x_1, x_2 = 1, x_3)\phi_D(x_3, x_4)\phi_E(x_3, x_5)\phi_F(x_5) \tag{S.6.54}$$

$$\propto \phi_A(x_1)\phi_C^{x_2}(x_1, x_3)\phi_D(x_3, x_4)\phi_E(x_3, x_5)\phi_F(x_5) \tag{S.6.55}$$

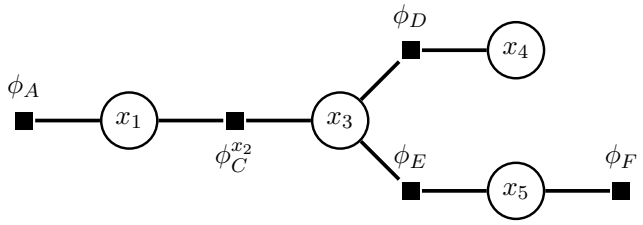
where $\phi_C^{x_2}(x_1, x_3) = \phi_C(x_1, x_2 = 1, x_3)$. The numerical values of $\phi_C^{x_2}(x_1, x_3)$ can be read from the table defining $\phi_C(x_1, x_2, x_3)$, extracting those rows where $x_2 = 1$,

	x_1	x_2	x_3	ϕ_C
	0	0	0	4
	1	0	0	2
→	0	1	0	2
→	1	1	0	6
	0	0	1	2
	1	0	1	6
→	0	1	1	6
→	1	1	1	4

so that

x_1	x_3	$\phi_C^{x_2}$
0	0	2
1	0	6
0	1	6
1	1	4

The factor graph for $p(x_1, x_3, x_4, x_5|x_2 = 1)$ is shown below. Factor ϕ_B has disappeared since it only depended on x_2 and thus became a constant. Factor ϕ_C is replaced by $\phi_C^{x_2}$ defined above. The remaining factors are the same as in the original factor graph.



(e) Compute $p(x_1 = 1|x_2 = 1)$, re-using messages that you have already computed for the evaluation of $p(x_1 = 1)$.

Solution. The message $\mu_{\phi_A \to x_1}$ is the same as in the original factor graph and $\mu_{x_3 \to \phi_C^{x_2}} = \mu_{x_3 \to \phi_C}$. This is because the outgoing message from x_3 corresponds to the effective factor obtained by summing out all variables in the sub-trees attached to x_3 (without the $\phi_C^{x_2}$ branch), and these sub-trees do not depend on x_2.

The message $\mu_{\phi_C^{x_2} \to x_1}$ needs to be newly computed. We have

$$\mu_{\phi_C^{x_2} \to x_1}(x_1) = \sum_{x_3} \phi_C^{x_2}(x_1, x_3)\mu_{x_3 \to \phi_C^{x_2}} \tag{S.6.56}$$

or in vector notation

$$\boldsymbol{\mu}_{\boldsymbol{\phi}_C^{\boldsymbol{x_2}} \to \boldsymbol{x_1}} = \boldsymbol{\phi}_C^{\boldsymbol{x_2}} \boldsymbol{\mu}_{\boldsymbol{x_3} \to \boldsymbol{\phi}_C^{\boldsymbol{x_2}}} \tag{S.6.57}$$

$$= \begin{pmatrix} \phi_C^{x_2}(x_1 = 0, x_3 = 0) & \phi_C^{x_2}(x_1 = 0, x_3 = 1) \\ \phi_C^{x_2}(x_1 = 1, x_3 = 0) & \phi_C^{x_2}(x_1 = 1, x_3 = 1) \end{pmatrix} \boldsymbol{\mu}_{\boldsymbol{x_3} \to \boldsymbol{\phi}_C^{\boldsymbol{x_2}}} \tag{S.6.58}$$

$$= \begin{pmatrix} 2 & 6 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 510 \\ 240 \end{pmatrix} \tag{S.6.59}$$

$$= \begin{pmatrix} 2460 \\ 4020 \end{pmatrix} \tag{S.6.60}$$

We thus obtain for the marginal posterior of x_1 given $x_2 = 1$:

$$\begin{pmatrix} p(x_1 = 0 | x_2 = 1) \\ p(x_1 = 1 | x_2 = 1) \end{pmatrix} \propto \boldsymbol{\mu}_{\boldsymbol{\phi_A} \to \boldsymbol{x_1}} \odot \boldsymbol{\mu}_{\boldsymbol{\phi}_C^{\boldsymbol{x_2}} \to \boldsymbol{x_1}} \tag{S.6.61}$$

$$\propto \begin{pmatrix} 2 \\ 4 \end{pmatrix} \odot \begin{pmatrix} 2460 \\ 4020 \end{pmatrix} \tag{S.6.62}$$

$$\propto \begin{pmatrix} 4920 \\ 16080 \end{pmatrix}. \tag{S.6.63}$$

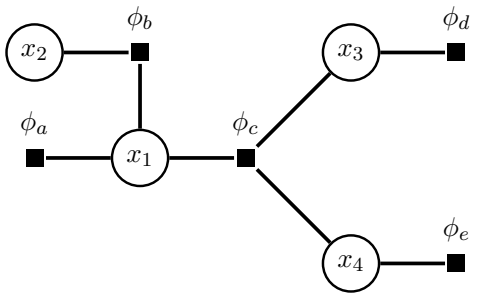
Normalisation gives

$$\begin{pmatrix} p(x_1 = 0 | x_2 = 1) \\ p(x_1 = 1 | x_2 = 1) \end{pmatrix} = \begin{pmatrix} 0.2343 \\ 0.7657 \end{pmatrix} \tag{S.6.64}$$

and thus $p(x_1 = 1 | x_2 = 1) = 0.7657$. The posterior probability is slightly larger than the prior probability, $p(x_1 = 1) = 0.7224$.

6.3 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

$$\mu_{\phi_e \to x_4}(x_4) = \phi_e(x_4) \tag{S.6.65}$$
$$\mu_{x_4 \to \phi_c}(x_4) = \mu_{\phi_e \to x_4}(x_4) \tag{S.6.66}$$

and hence

$$\phi_e(x_4) = \begin{cases} 1 & \text{if } x_4 = 0 \\ 3 & \text{if } x_4 = 1 \end{cases} \tag{S.6.67}$$

(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 $\mu_{x_3 \to \phi_c}(x_3)$:

$$\mu_{x_3 \to \phi_c}(x_3) = \mu_{\phi_d \to x_3}(x_3) \tag{S.6.68}$$
$$= \begin{cases} 1 & \text{if } x_3 = 0 \\ 2 & \text{if } x_3 = 1 \end{cases} \tag{S.6.69}$$

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

$$\mu_{\phi_c \to x_1}(x_1) = \sum_{x_3, x_4} \phi_c(x_1, x_3, x_4) \mu_{x_3 \to \phi_c}(x_3) \mu_{x_4 \to \phi_c}(x_4) \tag{S.6.70}$$

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:

$$\mu_{\phi_c \to x_1}(x_1 = 0) = \phi_c(0,0,0)\mu_{x_3 \to \phi_c}(0)\mu_{x_4 \to \phi_c}(0) \tag{S.6.71}$$
$$= 1 \cdot 1 \cdot 1 \tag{S.6.72}$$
$$= 1 \tag{S.6.73}$$

$$\mu_{\phi_c \to x_1}(x_1 = 1) = \phi_c(1,1,1)\mu_{x_3 \to \phi_c}(1)\mu_{x_4 \to \phi_c}(1) \tag{S.6.74}$$
$$= 1 \cdot 2 \cdot 3 \tag{S.6.75}$$
$$= 6 \tag{S.6.76}$$

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

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

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

$$p(x_1) \propto \mu_{\phi_a \to x_1}(x_1)\mu_{\phi_b \to x_1}(x_1)\mu_{\phi_c \to x_1}(x_1) \tag{S.6.77}$$

With

$$\mu_{\phi_b \to x_1}(x_1) = \sum_{x_2} \phi_b(x_1, x_2) \tag{S.6.78}$$

it follows that

$$\mu_{\phi_b \to x_1}(x_1 = 0) = \sum_{x_2} \phi_b(0, x_2) \tag{S.6.79}$$
$$= 5 + 2 \tag{S.6.80}$$
$$= 7 \tag{S.6.81}$$
$$\mu_{\phi_b \to x_1}(x_1 = 1) = \sum_{x_2} \phi_b(1, x_2) \tag{S.6.82}$$
$$= 2 + 6 \tag{S.6.83}$$
$$= 8 \tag{S.6.84}$$

Hence, we obtain

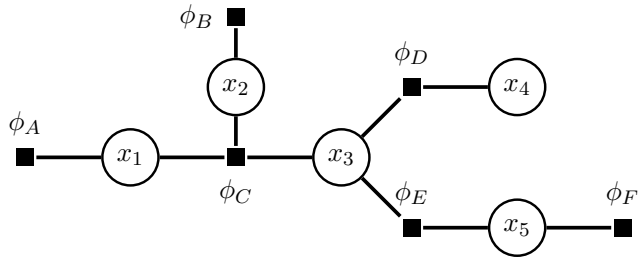
$$p(x_1 = 0) \propto 2 \cdot 7 \cdot 1 = 14 \tag{S.6.85}$$
$$p(x_1 = 1) \propto 1 \cdot 8 \cdot 6 = 48 \tag{S.6.86}$$

and normalisation yields the desired result

$$p(x_1 = 1) = \frac{48}{14 + 48} = \frac{48}{62} = \frac{24}{31} = 0.774 \tag{S.6.87}$$

6.4 Max-sum message passing

We here compute most probable states for the factor graph and factors below.



Let all variables be binary, $x_i \in \{0, 1\}$, and the factors be defined as follows:

x_1	ϕ_A
0	2
1	4

x_2	ϕ_B
0	4
1	4

x_1	x_2	x_3	ϕ_C
0	0	0	4
1	0	0	2
0	1	0	2
1	1	0	6
0	0	1	2
1	0	1	6
0	1	1	6
1	1	1	4

x_3	x_4	ϕ_D
0	0	8
1	0	2
0	1	2
1	1	6

x_3	x_5	ϕ_E
0	0	3
1	0	6
0	1	6
1	1	3

x_5	ϕ_F
0	1
1	8

(a) Will we need to compute the normalising constant Z to determine $\text{argmax}_{\mathbf{x}}\, p(x_1, \ldots, x_5)$?

Solution. This is not necessary since $\text{argmax}_{\mathbf{x}}\, p(x_1, \ldots, x_5) = \text{argmax}_{\mathbf{x}}\, cp(x_1, \ldots, x_5)$ for any constant c. Algorithmically, the backtracking algorithm is also invariant to any scaling of the factors.

(b) Compute $\text{argmax}_{x_1,x_2,x_3}\, p(x_1, x_2, x_3 | x_4 = 0, x_5 = 0)$ via max-sum message passing.

Solution. We first derive the factor graph and corresponding factors for $p(x_1, x_2, x_3 | x_4 = 0, x_5 = 0)$.

For fixed values of x_4, x_5, the two variables are removed from the graph, and the factors $\phi_D(x_3, x_4)$ and $\phi_E(x_3, x_5)$ are reduced to univariate factors $\phi_D^{x_4}(x_3)$ and $\phi_D^{x_5}(x_3)$ by retaining those rows in the table where $x_4 = 0$ and $x_5 = 0$, respectively:

x_3	$\phi_D^{x_4}$
0	8
1	2

x_3	$\phi_E^{x_5}$
0	3
1	6

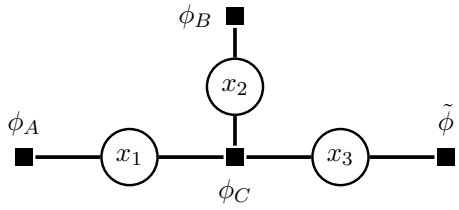
Since both factors only depend on x_3, they can be combined into a new factor $\tilde{\phi}(x_3)$ by element-wise multiplication.

x_3	$\tilde{\phi}$
0	24
1	12

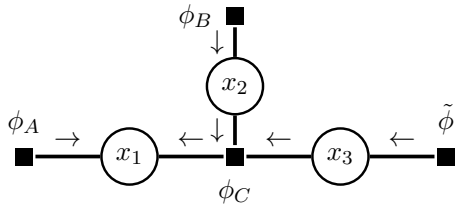
Moreover, since we work with an unnormalised model, we can rescale the factor so that the maximum value is one, so that

x_3	$\tilde{\phi}$
0	2
1	1

Factor $\phi_F(x_5)$ is a constant for fixed value of x_5 and can be ignored. The factor graph for $p(x_1, x_2, x_3|x_4 = 0, x_5 = 0)$ thus is



Let us fix x_1 as root towards which we compute the messages. The messages that we need to compute are shown in the following graph



Next, we compute the leaf (log) messages. We only have factor nodes as leaf nodes so that

$$\boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} \log \phi_A(x_1 = 0) \\ \log \phi_A(x_1 = 1) \end{pmatrix} = \begin{pmatrix} \log 2 \\ \log 4 \end{pmatrix} \tag{S.6.88}$$

and similarly

$$\boldsymbol{\lambda}_{\phi_B \to x_2} = \begin{pmatrix} \log \phi_B(x_2 = 0) \\ \log \phi_B(x_2 = 1) \end{pmatrix} = \begin{pmatrix} \log 4 \\ \log 4 \end{pmatrix} \quad \boldsymbol{\lambda}_{\tilde{\phi} \to x_3} = \begin{pmatrix} \log \tilde{\phi}(x_3 = 0) \\ \log \tilde{\phi}(x_3 = 1) \end{pmatrix} = \begin{pmatrix} \log 2 \\ \log 1 \end{pmatrix} \tag{S.6.89}$$

Since the variable nodes x_2 and x_3 only have one incoming edge each, we obtain

$$\boldsymbol{\lambda}_{x_2 \to \phi_C} = \boldsymbol{\lambda}_{\phi_B \to x_2} = \begin{pmatrix} \log 4 \\ \log 4 \end{pmatrix} \qquad \boldsymbol{\lambda}_{x_3 \to \phi_C} = \boldsymbol{\lambda}_{\tilde{\phi} \to x_3} = \begin{pmatrix} \log 2 \\ \log 1 \end{pmatrix} \tag{S.6.90}$$

The message $\lambda_{\phi_C \to x_1}(x_1)$ equals

$$\lambda_{\phi_C \to x_1}(x_1) = \max_{x_2, x_3} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3) \tag{S.6.91}$$

where we wrote the messages in non-vector notation to highlight their dependency on the variables x_2 and x_3. We now have to consider all combinations of x_2 and x_3

x_2	x_3	$\log \phi_C(x_1 = 0, x_2, x_3)$
0	0	$\log 4$
1	0	$\log 2$
0	1	$\log 2$
1	1	$\log 6$

x_2	x_3	$\log \phi_C(x_1 = 1, x_2, x_3)$
0	0	$\log 2$
1	0	$\log 6$
0	1	$\log 6$
1	1	$\log 4$

Furthermore

x_2	x_3	$\lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	$\log 4 + \log 2 = \log 8$
1	0	$\log 4 + \log 2 = \log 8$
0	1	$\log 4$
1	1	$\log 4$

Hence for $x_1 = 0$, we have

x_2	x_3	$\log \phi_C(x_1 = 0, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	$\log 4 + \log 8 = \log 32$
1	0	$\log 2 + \log 8 = \log 16$
0	1	$\log 2 + \log 4 = \log 8$
1	1	$\log 6 + \log 4 = \log 24$

The maximal value is $\log 32$ and for backtracking, we also need to keep track of the argmax which is here $\hat{x}_2 = \hat{x}_3 = 0$.

For $x_1 = 1$, we have

x_2	x_3	$\log \phi_C(x_1 = 1, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	$\log 2 + \log 8 = \log 16$
1	0	$\log 6 + \log 8 = \log 48$
0	1	$\log 6 + \log 4 = \log 24$
1	1	$\log 4 + \log 4 = \log 16$

The maximal value is $\log 48$ and the argmax is $(\hat{x}_2 = 1, \hat{x}_3 = 0)$.

So overall, we have

$$\boldsymbol{\lambda}_{\phi_C \to x_1} = \begin{pmatrix} \lambda_{\phi_C \to x_1}(x_1 = 0) \\ \lambda_{\phi_C \to x_1}(x_1 = 1) \end{pmatrix} = \begin{pmatrix} \log 32 \\ \log 48 \end{pmatrix} \tag{S.6.92}$$

and the argmax back-tracking function is

$$\lambda^*_{\phi_C \to x_1}(x_1) = \begin{cases} (\hat{x}_2 = 0, \hat{x}_3 = 0) & \text{if } x_1 = 0 \\ (\hat{x}_2 = 1, \hat{x}_3 = 0) & \text{if } x_1 = 1 \end{cases} \tag{S.6.93}$$

We now have all incoming messages to the assigned root node x_1. *Ignoring the normalising constant*, we obtain

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma^*(x_1 = 0) \\ \gamma^*(x_1 = 1) \end{pmatrix} = \boldsymbol{\lambda}_{\phi_A \to x_1} + \boldsymbol{\lambda}_{\phi_C \to x_1} \tag{S.6.94}$$

$$= \begin{pmatrix} \log 2 \\ \log 4 \end{pmatrix} + \begin{pmatrix} \log 32 \\ \log 48 \end{pmatrix} = \begin{pmatrix} \log 64 \\ \log 192 \end{pmatrix} \tag{S.6.95}$$

The value x_1 for which $\gamma^*(x_1)$ is largest is thus $\hat{x}_1 = 1$. Plugging $\hat{x}_1 = 1$ into the backtracking function $\lambda^*_{\phi_C \to x_1}(x_1)$ gives

$$(\hat{x}_1, \hat{x}_2, \hat{x}_3) = \operatorname*{argmax}_{x_1, x_2, x_3} p(x_1, x_2, x_3 | x_4 = 0, x_5 = 0) = (1, 1, 0). \tag{S.6.96}$$

In this low-dimensional example, we can verify the solution by computing the unnormalised pmf for all combinations of x_1, x_2, x_3. This is done in the following table where we start with the table for ϕ_C and then multiply-in the further factors ϕ_A, $\tilde{\phi}$ and ϕ_B.

x_1	x_2	x_3	ϕ_C	$\phi_C\phi_A$	$\phi_C\phi_A\tilde{\phi}$	$\phi_C\phi_A\tilde{\phi}\phi_B$
0	0	0	4	8	16	$16 \cdot 4$
1	0	0	2	8	16	$16 \cdot 4$
0	1	0	2	8	16	$16 \cdot 4$
1	1	0	6	24	48	$48 \cdot 4$
0	0	1	2	8	8	$8 \cdot 4$
1	0	1	6	24	24	$24 \cdot 4$
0	1	1	6	12	12	$12 \cdot 4$
1	1	1	4	16	16	$16 \cdot 4$

For example, for the column $\phi_c\phi_A$, we multiply each value of $\phi_C(x_1, x_2, x_3)$ by $\phi_A(x_1)$, so that the rows with $x_1 = 0$ get multiplied by 2, and the rows with $x_1 = 1$ by 4.

The maximal value in the final column is achieved for $x_1 = 1, x_2 = 1, x_3 = 0$, in line with the result above (and $48 \cdot 4 = 192$). Since $\phi_B(x_2)$ is a constant, being equal to 4 for all values of x_2, we could have ignored it in the computation. The formal reason for this is that since the model is unnormalised, we are allowed to rescale each factor by an arbitrary (factor-dependent) *constant*. This operation does not change the model. So we could divide ϕ_B by 4 which would give a value of 1, so that the factor can indeed be ignored.

(c) Compute $\operatorname{argmax}_{x_1,\ldots,x_5} p(x_1, \ldots, x_5)$ via max-sum message passing with x_1 as root.

Solution. As discussed in the solution to the answer above, we can drop factor $\phi_B(x_2)$ since it takes the same value for all x_2. Moreover, we can rescale the individual factors by a constant so they are more amenable to calculations by hand. We normalise them such that the largest value is one, which gives the following factors. Note that this is entirely optional.

x_1	ϕ_A
0	1
1	2

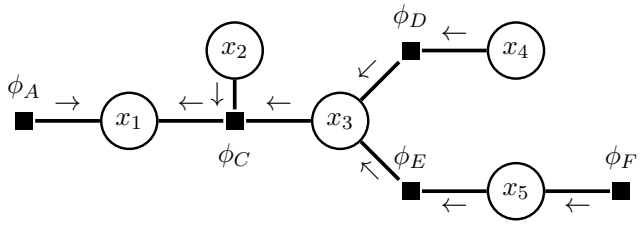
x_1	x_2	x_3	ϕ_C
0	0	0	2
1	0	0	1
0	1	0	1
1	1	0	3
0	0	1	1
1	0	1	3
0	1	1	3
1	1	1	2

x_3	x_4	ϕ_D
0	0	4
1	0	1
0	1	1
1	1	3

x_3	x_5	ϕ_E
0	0	1
1	0	2
0	1	2
1	1	1

x_5	ϕ_F
0	1
1	8

The factor graph without ϕ_B together with the messages that we need to compute is:



The leaf (log) messages are (using vector notation where the top element corresponds to $x_i = 0$ and the bottom one to $x_i = 1$):

$$\boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} 0 \\ \log 2 \end{pmatrix} \quad \boldsymbol{\lambda}_{x_2 \to \phi_C} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \boldsymbol{\lambda}_{x_4 \to \phi_D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \boldsymbol{\lambda}_{\phi_F \to x_5} = \begin{pmatrix} 0 \\ \log 8 \end{pmatrix} \tag{S.6.97}$$

The variable node x_5 only has one incoming edge so that $\boldsymbol{\lambda}_{x_5 \to \phi_E} = \boldsymbol{\lambda}_{\phi_F \to x_5}$. The message $\lambda_{\phi_E \to x_3}(x_3)$ equals

$$\lambda_{\phi_E \to x_3}(x_3) = \max_{x_5} \log \phi_E(x_3, x_5) + \lambda_{x_5 \to \phi_E}(x_5) \tag{S.6.98}$$

Writing out $\log \phi_E(x_3, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$ for all x_5 as a function of x_3 we have

x_5	$\log \phi_E(x_3 = 0, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$
0	$\log 1 + 0 = 0$
1	$\log 2 + \log 8 = \log 16$

x_5	$\log \phi_E(x_3 = 1, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$
0	$\log 2 + 0 = \log 2$
1	$\log 1 + \log 8 = \log 8$

Taking the maximum over x_5 as a function of x_3, we obtain

$$\boldsymbol{\lambda}_{\phi_E \to x_3} = \begin{pmatrix} \log 16 \\ \log 8 \end{pmatrix} \tag{S.6.99}$$

and the backtracking function that indicates the maximiser $\hat{x}_5 = \operatorname{argmax}_{x_5} \log \phi_E(x_3, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$ as a function of x_3 equals

$$\lambda^*_{\phi_E \to x_3}(x_3) = \begin{cases} \hat{x}_5 = 1 & \text{if } x_3 = 0 \\ \hat{x}_5 = 1 & \text{if } x_3 = 1 \end{cases} \tag{S.6.100}$$

We perform the same kind of operation for $\lambda_{\phi_D \to x_3}(x_3)$

$$\lambda_{\phi_D \to x_3}(x_3) = \max_{x_4} \log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4) \tag{S.6.101}$$

Since $\lambda_{x_4 \to \phi_D}(x_4) = 0$ for all x_4, the table with all values of $\log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4)$ is

x_3	x_4	$\log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4)$
0	0	$\log 4 + 0 = \log 4$
1	0	$\log 1 + 0 = 0$
0	1	$\log 1 + 0 = 0$
1	1	$\log 3 + 0 = \log 3$

Taking the maximum over x_4 as a function of x_3 we thus obtain

$$\boldsymbol{\lambda}_{\phi_D \to x_3} = \begin{pmatrix} \log 4 \\ \log 3 \end{pmatrix} \tag{S.6.102}$$

and the backtracking function that indicates the maximiser $\hat{x}_4 = \operatorname{argmax}_{x_4} \log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4)$ as a function of x_3 equals

$$\lambda^*_{\phi_D \to x_3}(x_3) = \begin{cases} \hat{x}_4 = 0 & \text{if } x_3 = 0 \\ \hat{x}_4 = 1 & \text{if } x_3 = 1 \end{cases} \tag{S.6.103}$$

For the message $\lambda_{x_3 \to \phi_C}(x_3)$ we add together the messages $\lambda_{\phi_E \to x_3}(x_3)$ and $\lambda_{\phi_D \to x_3}(x_3)$ which gives

$$\boldsymbol{\lambda}_{x_3 \to \phi_C} = \begin{pmatrix} \log 16 + \log 4 \\ \log 8 + \log 3 \end{pmatrix} = \begin{pmatrix} \log 64 \\ \log 24 \end{pmatrix} \tag{S.6.104}$$

Next we compute the message $\lambda_{\phi_C \to x_1}(x_1)$ by maximising over x_2 and x_3,

$$\lambda_{\phi_C \to x_1}(x_1) = \max_{x_2, x_3} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3) \tag{S.6.105}$$

Since $\lambda_{x_2 \to \phi_C}(x_2) = 0$, the problem becomes

$$\lambda_{\phi_C \to x_1}(x_1) = \max_{x_2, x_3} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_3 \to \phi_C}(x_3) \tag{S.6.106}$$

Building on the table for ϕ_C, we form a table with all values of $\log \phi_C(x_1, x_2, x_3) + \lambda_{x_3 \to \phi_C}(x_3)$

x_1	x_2	x_3	$\log \phi_C(x_1, x_2, x_3) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	0	$\log 2 + \log 64 = \mathbf{\log 128}$
1	0	0	$0 + \log 64 = \log 64$
0	1	0	$0 + \log 64 = \log 64$
1	1	0	$\log 3 + \log 64 = \mathbf{\log 192}$
0	0	1	$\log 24$
1	0	1	$\log 3 + \log 24 = \log 72$
0	1	1	$\log 3 + \log 24 = \log 72$
1	1	1	$\log 2 + \log 24 = \log 48$

The maximal value as a function of x_1 are highlighted in the table, which gives the message

$$\boldsymbol{\lambda}_{\phi_C \to x_1} = \begin{pmatrix} \log 128 \\ \log 192 \end{pmatrix} \tag{S.6.107}$$

and the backtracking function

$$\lambda^*_{\phi_C \to x_1}(x_1) = \begin{cases} (\hat{x}_2 = 0, \hat{x}_3 = 0) & \text{if } x_1 = 0 \\ (\hat{x}_2 = 1, \hat{x}_3 = 0) & \text{if } x_1 = 1 \end{cases} \tag{S.6.108}$$

We now have all incoming messages to the assigned root node x_1. *Ignoring the normalising constant*, we obtain

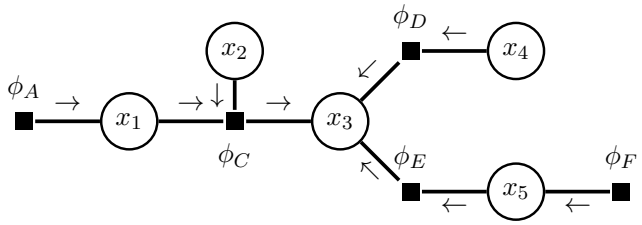
$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma^*(x_1 = 0) \\ \gamma^*(x_1 = 1) \end{pmatrix} = \begin{pmatrix} 0 + \log 128 \\ \log 2 + \log 192 \end{pmatrix} \tag{S.6.109}$$

We can now start the backtracking to compute the desired $\operatorname{argmax}_{x_1,\ldots,x_5} p(x_1, \ldots, x_5)$. Starting at the root we have $\hat{x}_1 = \operatorname{argmax}_{x_1} \gamma^*(x_1) = 1$. Plugging this value into the look-up table $\lambda^*_{\phi_C \to x_1}(x_1)$, we obtain $(\hat{x}_2 = 1, \hat{x}_3 = 0)$. With the look-up table $\lambda^*_{\phi_E \to x_3}(x_3)$ we find $\hat{x}_5 = 1$ and $\lambda^*_{\phi_D \to x_3}(x_3)$ gives $\hat{x}_4 = 0$ so that overall

$$\operatorname*{argmax}_{x_1,\ldots,x_5} p(x_1, \ldots, x_5) = (1, 1, 0, 0, 1). \tag{S.6.110}$$

(d) Compute $\operatorname{argmax}_{x_1,\ldots,x_5} p(x_1, \ldots, x_5)$ via max-sum message passing with x_3 as root.

Solution. With x_3 as root, we need the following messages:



The following messages are the same as when x_1 was the root:

$$\boldsymbol{\lambda}_{\phi_D \to x_3} = \begin{pmatrix} \log 4 \\ \log 3 \end{pmatrix} \quad \boldsymbol{\lambda}_{\phi_E \to x_3} = \begin{pmatrix} \log 16 \\ \log 8 \end{pmatrix} \quad \boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} 0 \\ \log 2 \end{pmatrix} \quad \boldsymbol{\lambda}_{x_2 \to \phi_C} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{S.6.111}$$

Since x_1 has only one incoming message, we further have

$$\boldsymbol{\lambda}_{x_1 \to \phi_C} = \boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} 0 \\ \log 2 \end{pmatrix}. \tag{S.6.112}$$

We next compute $\lambda_{\phi_C \to x_3}(x_3)$,

$$\lambda_{\phi_C \to x_3}(x_3) = \max_{x_1, x_2} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_1 \to \phi_C}(x_1) + \lambda_{x_2 \to \phi_C}(x_2). \tag{S.6.113}$$

We first form a table for $\log \phi_C(x_1, x_2, x_3) + \lambda_{x_1 \to \phi_C}(x_1) + \lambda_{x_2 \to \phi_C}(x_2)$ noting that $\lambda_{x_2 \to \phi_C}(x_2) = 0$

x_1	x_2	x_3	$\log \phi_C(x_1, x_2, x_3) + \lambda_{x_1 \to \phi_C}(x_1) + \lambda_{x_2 \to \phi_C}(x_2)$
0	0	0	$\log 2 + 0 = \log 2$
1	0	0	$0 + \log 2 = \log 2$
0	1	0	$0 + 0 = 0$
1	1	0	$\log 3 + \log 2 = \mathbf{\log 6}$
0	0	1	$0 + 0 = 0$
1	0	1	$\log 3 + \log 2 = \mathbf{\log 6}$
0	1	1	$\log 3 + 0 = \log 3$
1	1	1	$\log 2 + \log 2 = \log 4$

The maximal value as a function of x_3 are highlighted in the table, which gives the message

$$\boldsymbol{\lambda}_{\phi_C \to x_3} = \begin{pmatrix} \log 6 \\ \log 6 \end{pmatrix} \tag{S.6.114}$$

and the backtracking function

$$\lambda^*_{\phi_C \to x_3}(x_3) = \begin{cases} (\hat{x}_1 = 1, \hat{x}_2 = 1) & \text{if } x_3 = 0 \\ (\hat{x}_1 = 1, \hat{x}_2 = 0) & \text{if } x_3 = 1 \end{cases} \tag{S.6.115}$$

We have now all incoming messages for x_3 and can compute $\gamma^*(x_3)$ up the normalising constant $-\log Z$ (which is not needed if we are interested in the argmax only:

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma^*(x_3 = 0) \\ \gamma^*(x_3 = 1) \end{pmatrix} = \boldsymbol{\lambda}_{\phi_C \to x_3} + \boldsymbol{\lambda}_{\phi_D \to x_3} + \boldsymbol{\lambda}_{\phi_E \to x_3} \tag{S.6.116}$$

$$= \begin{pmatrix} \log 6 + \log 4 + \log 16 = \log 384 \\ \log 6 + \log 3 + \log 8 = \log 144 \end{pmatrix} \tag{S.6.117}$$

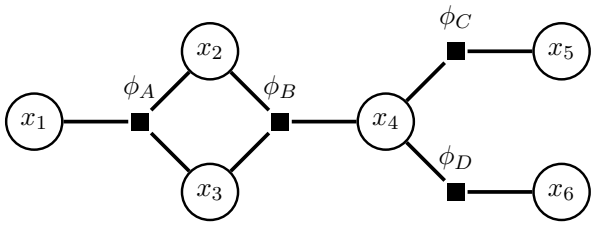
We can now start the backtracking which gives: $\hat{x}_3 = 0$, so that $\lambda^*_{\phi_C \to x_3}(0) = (\hat{x}_1 = 1, \hat{x}_2 = 1)$. The backtracking functions $\lambda^*_{\phi_E \to x_3}(x_3)$ and $\lambda^*_{\phi_D \to x_3}(x_3)$ are the same for question (c), which gives $\lambda^*_{\phi_E \to x_3}(0) = \hat{x}_5 = 1$ and $\lambda^*_{\phi_D \to x_3}(0) = \hat{x}_4 = 0$. Hence, overall, we find

$$\underset{x_1, \ldots, x_5}{\operatorname{argmax}}\, p(x_1, \ldots, x_5) = (1, 1, 0, 0, 1). \tag{S.6.118}$$

Note that this matches the result from question (c) where x_1 was the root. This is because the output of the max-sum algorithm is invariant to the choice of the root.

6.5 Choice of elimination order in factor graphs

Consider the following factor graph, which contains a loop:



Let all variables be binary, $x_i \in \{0, 1\}$, and the factors be defined as follows:

x_1	x_2	x_3	ϕ_A
0	0	0	4
1	0	0	2
0	1	0	2
1	1	0	6
0	0	1	2
1	0	1	6
0	1	1	6
1	1	1	4

x_2	x_3	x_4	ϕ_B
0	0	0	2
1	0	0	2
0	1	0	4
1	1	0	2
0	0	1	6
1	0	1	8
0	1	1	4
1	1	1	2

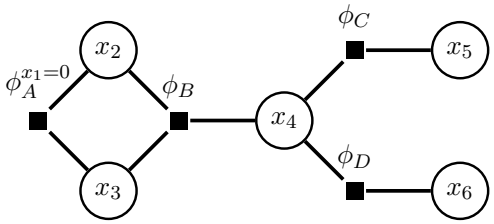
x_4	x_5	ϕ_C
0	0	8
1	0	2
0	1	2
1	1	6

x_4	x_6	ϕ_D
0	0	3
1	0	6
0	1	6
1	1	3

(a) Draw the factor graph corresponding to $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$ and give the tables defining the new factors $\phi_A^{x_1=0}(x_2, x_3)$ and $\phi_D^{x_6=1}(x_4)$ that you obtain.

Solution. First condition on $x_1 = 0$:

Factor node $\phi_A(x_1, x_2, x_3)$ depends on x_1, thus we create a new factor $\phi_A^{x_1=0}(x_2, x_3)$ from the table for ϕ_A using the rows where $x_1 = 0$.



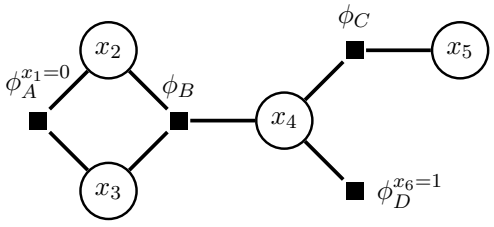
	x_1	x_2	x_3	ϕ_A
$\rightarrow$	0	0	0	4
	1	0	0	2
$\rightarrow$	0	1	0	2
	1	1	0	6
$\rightarrow$	0	0	1	2
	1	0	1	6
$\rightarrow$	0	1	1	6
	1	1	1	4

so that

x_2	x_3	$\phi_A^{x_1=0}$
0	0	4
1	0	2
0	1	2
1	1	6

Next condition on $x_6 = 1$:

Factor node $\phi_D(x_4, x_6)$ depends on x_6, thus we create a new factor $\phi_D^{x_6=1}(x_4)$ from the table for ϕ_D using the rows where $x_6 = 1$.



	x_4	x_6	ϕ_D
	0	0	3
	1	0	6
$\rightarrow$	0	1	6
$\rightarrow$	1	1	3

so that

x_4	$\phi_D^{x_6=1}$
0	6
1	3

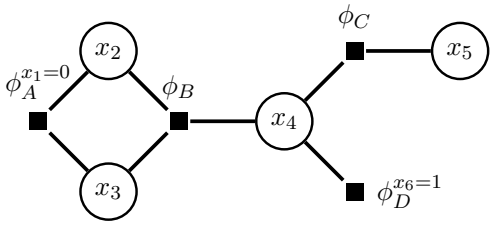
(b) Find $p(x_2 \mid x_1 = 0, x_6 = 1)$ using the elimination ordering (x_4, x_5, x_3):

(i) Draw the graph for $p(x_2, x_3, x_5 \mid x_1 = 0, x_6 = 1)$ by marginalising x_4
Compute the table for the new factor $\tilde{\phi}_4(x_2, x_3, x_5)$

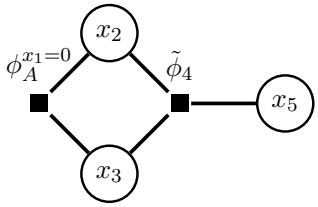
(ii) Draw the graph for $p(x_2, x_3 \mid x_1 = 0, x_6 = 1)$ by marginalising x_5
Compute the table for the new factor $\tilde{\phi}_{45}(x_2, x_3)$

(iii) Draw the graph for $p(x_2 \mid x_1 = 0, x_6 = 1)$ by marginalising x_3
Compute the table for the new factor $\tilde{\phi}_{453}(x_2)$

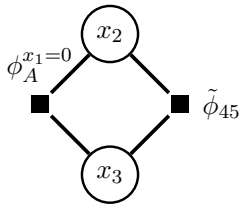
Solution. Starting with the factor graph for $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$



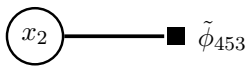
Marginalising x_4 combines the three factors ϕ_B, ϕ_C and $\phi_D^{x_6=1}$



Marginalising x_5 modifies the factor $\tilde{\phi}_4$



Marginalising x_3 combines the factors $\phi_A^{x_1=0}$ and $\tilde{\phi}_{45}$



We now compute the tables for the new factors $\tilde{\phi}_4$, $\tilde{\phi}_{45}$, $\tilde{\phi}_{453}$.
First find $\tilde{\phi}_4(x_2, x_3, x_5)$

x_2	x_3	x_4	ϕ_B
0	0	0	2
1	0	0	2
0	1	0	4
1	1	0	2
0	0	1	6
1	0	1	8
0	1	1	4
1	1	1	2

x_4	x_5	ϕ_C
0	0	8
1	0	2
0	1	2
1	1	6

x_4	$\phi_D^{x_6=1}$
0	6
1	3

so that $\phi_*(x_2, x_3, x_4, x_5) = \phi_B(x_2, x_3, x_4)\phi_C(x_4, x_5)\phi_D^{x_6=1}(x_4)$ equals

x_2	x_3	x_4	x_5	$\phi_*(x_2, x_3, x_4, x_5)$
0	0	0	0	2 * 8 * 6
1	0	0	0	2 * 8 * 6
0	1	0	0	4 * 8 * 6
1	1	0	0	2 * 8 * 6
0	0	1	0	6 * 2 * 3
1	0	1	0	8 * 2 * 3
0	1	1	0	4 * 2 * 3
1	1	1	0	2 * 2 * 3
0	0	0	1	2 * 2 * 6
1	0	0	1	2 * 2 * 6
0	1	0	1	4 * 2 * 6
1	1	0	1	2 * 2 * 6
0	0	1	1	6 * 6 * 3
1	0	1	1	8 * 6 * 3
0	1	1	1	4 * 6 * 3
1	1	1	1	2 * 6 * 3

and

x_2	x_3	x_5	$\sum_{x_4} \phi_B(x_2, x_3, x_4)\phi_C(x_4, x_5)\phi_D^{x_6=1}(x_4)$		$\tilde{\phi}_4$
0	0	0	(2 * 8 * 6) + (6 * 2 * 3)	=	132
1	0	0	(2 * 8 * 6) + (8 * 2 * 3)	=	144
0	1	0	(4 * 8 * 6) + (4 * 2 * 3)	=	216
1	1	0	(2 * 8 * 6) + (2 * 2 * 3)	=	108
0	0	1	(2 * 2 * 6) + (6 * 6 * 3)	=	132
1	0	1	(2 * 2 * 6) + (8 * 6 * 3)	=	168
0	1	1	(4 * 2 * 6) + (4 * 6 * 3)	=	120
1	1	1	(2 * 2 * 6) + (2 * 6 * 3)	=	60

Next find $\tilde{\phi}_{45}(x_2, x_3)$

x_2	x_3	x_5	$\tilde{\phi}_4$
0	0	0	132
1	0	0	144
0	1	0	216
1	1	0	108
0	0	1	132
1	0	1	168
0	1	1	120
1	1	1	60

so that

x_2	x_3	$\sum_{x_5} \tilde{\phi}_4(x_2, x_3, x_5)$		$\tilde{\phi}_{45}$
0	0	132 + 132	=	264
1	0	144 + 168	=	312
0	1	216 + 120	=	336
1	1	108 + 60	=	168

Finally find $\tilde{\phi}_{453}(x_2)$

x_2	x_3	$\phi_A^{x_1=0}$
0	0	4
1	0	2
0	1	2
1	1	6

x_2	x_3	$\tilde{\phi}_{45}$
0	0	264
1	0	312
0	1	336
1	1	168

so that

x_2	$\sum_{x_3} \tilde{\phi}_{45}(x_2, x_3)\phi_A^{x_1=0}(x_2, x_3)$		$\tilde{\phi}_{453}$
0	(4 * 264) + (2 * 336)	=	1728
1	(2 * 312) + (6 * 168)	=	1632

The normalising constant is $Z = 1728 + 1632$. Our conditional marginal is thus:

$$p(x_2 \mid x_1 = 0, x_6 = 1) = \begin{pmatrix} 1728/Z \\ 1632/Z \end{pmatrix} = \begin{pmatrix} 0.514 \\ 0.486 \end{pmatrix} \tag{S.6.119}$$

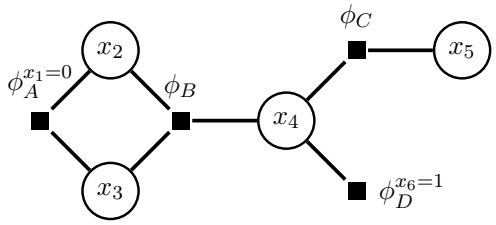
(c) Now determine $p(x_2 \mid x_1 = 0, x_6 = 1)$ with the elimination ordering (x_5, x_4, x_3):

(i) Draw the graph for $p(x_2, x_3, x_4, \mid x_1 = 0, x_6 = 1)$ by marginalising x_5
Compute the table for the new factor $\tilde{\phi}_5(x_4)$

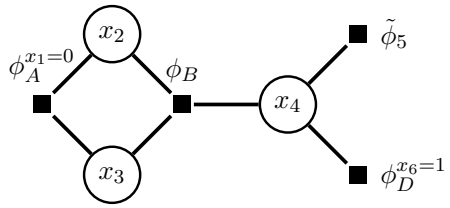
(ii) Draw the graph for $p(x_2, x_3 \mid x_1 = 0, x_6 = 1)$ by marginalising x_4
Compute the table for the new factor $\tilde{\phi}_{54}(x_2, x_3)$

(iii) Draw the graph for $p(x_2 \mid x_1 = 0, x_6 = 1)$ by marginalising x_3
Compute the table for the new factor $\tilde{\phi}_{543}(x_2)$

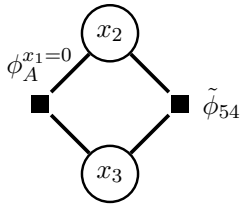
Solution. Starting with the factor graph for $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$



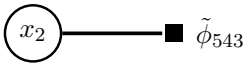
Marginalising x_5 modifies the factor ϕ_C



Marginalising x_4 combines the three factors ϕ_B, $\tilde{\phi}_5$ and $\phi_D^{x_6=1}$



Marginalising x_3 combines the factors $\phi_A^{x_1=0}$ and $\tilde{\phi}_{54}$



We now compute the tables for the new factors $\tilde{\phi}_5$, $\tilde{\phi}_{54}$, and $\tilde{\phi}_{543}$.

First find $\tilde{\phi}_5(x_4)$

x_4	x_5	ϕ_C
0	0	8
1	0	2
0	1	2
1	1	6

so that

x_4	$\sum_{x_5} \phi_C(x_4, x_5)$		$\tilde{\phi}_5$
0	$8 + 2$	$=$	10
1	$2 + 6$	$=$	8

Next find $\tilde{\phi}_{54}(x_2, x_3)$

x_2	x_3	x_4	ϕ_B
0	0	0	2
1	0	0	2
0	1	0	4
1	1	0	2
0	0	1	6
1	0	1	8
0	1	1	4
1	1	1	2

x_4	$\tilde{\phi}_5$
0	10
1	8

x_4	$\phi_D^{x_6=1}$
0	6
1	3

so that $\phi_*(x_2, x_3, x_4) = \phi_B(x_2, x_3, x_4)\tilde{\phi}_5(x_4)\phi_D^{x_6=1}(x_4)$ equals

x_2	x_3	x_4	$\phi_*(x_2, x_3, x_4)$
0	0	0	2 * 10 * 6
1	0	0	2 * 10 * 6
0	1	0	4 * 10 * 6
1	1	0	2 * 10 * 6
0	0	1	6 * 8 * 3
1	0	1	8 * 8 * 3
0	1	1	4 * 8 * 3
1	1	1	2 * 8 * 3

and

x_2	x_3	$\sum_{x_4} \phi_B(x_2, x_3, x_4)\tilde{\phi}_5(x_4)\phi_D^{x_6=1}(x_4)$		$\tilde{\phi}_{54}$
0	0	(2 * 10 * 6) + (6 * 8 * 3)	=	264
1	0	(2 * 10 * 6) + (8 * 8 * 3)	=	312
0	1	(4 * 10 * 6) + (4 * 8 * 3)	=	336
1	1	(2 * 10 * 6) + (2 * 8 * 3)	=	168

Finally find $\tilde{\phi}_{543}(x_2)$

x_2	x_3	$\phi_A^{x_1=0}$
0	0	4
1	0	2
0	1	2
1	1	6

x_2	x_3	$\tilde{\phi}_{54}$
0	0	264
1	0	312
0	1	336
1	1	168

so that

x_2	$\sum_{x_3} \tilde{\phi}_{54}(x_2, x_3)\phi_A^{x_1=0}(x_2, x_3)$		$\tilde{\phi}_{543}$
0	(4 * 264) + (2 * 336)	=	1728
1	(2 * 312) + (6 * 168)	=	1632

As with the ordering in the previous part, we should come to the same result for our conditional marginal distribution.The normalising constant is $Z = 1728 + 1632$, so that the conditional marginal is

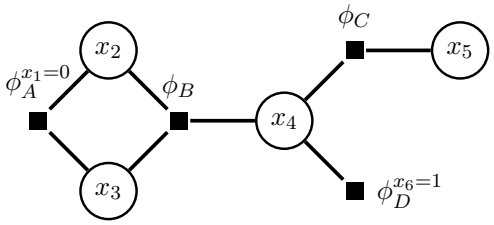
$$p(x_2 \mid x_1 = 0, x_6 = 1) = \begin{pmatrix} 1728/Z \\ 1632/Z \end{pmatrix} = \begin{pmatrix} 0.514 \\ 0.486 \end{pmatrix} \tag{S.6.120}$$

(d) Which variable ordering, (x_4, x_5, x_3) or (x_5, x_4, x_3) do you prefer?

Solution. The ordering (x_5, x_4, x_3) is cheaper and should be preferred over the ordering (x_4, x_5, x_3) .

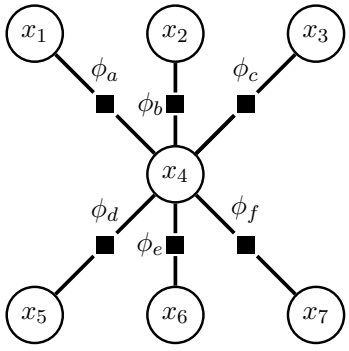
The reason for the difference in the cost is that x_4 has three neighbours in the factor graph for $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$. However, after elimination of x_5, which has only one neighbour, x_4 has only two neighbours left. Eliminating variables with more neighbours leads to larger (temporary) factors and hence a larger cost. We can see this from the tables that were generated during the computation (or numbers that we needed to add together): for the ordering (x_4, x_5, x_3), the largest table had 2^4 entries while for (x_5, x_4, x_3), it had 2^3 entries.

Choosing a reasonable variable ordering has a direct effect on the computational complexity of variable elimination. This effect becomes even more pronounced when the domain of our discrete variables has a size greater than 2 (binary variables), or if the variables are continuous.



6.6 Choice of elimination order in factor graphs

We would like to compute the marginal $p(x_1)$ by variable elimination for a joint pmf represented by the following factor graph. All variables x_i can take K different values.



(a) A friend proposes the elimination order $x_4, x_5, x_6, x_7, x_3, x_2$, i.e. to do x_4 first and x_2 last. Explain why this is computationally inefficient.

Solution. According to the factor graph, $p(x_1, \dots, x_7)$ factorises as

$$p(x_1, \dots, x_7) \propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\phi_f(x_7, x_4) \tag{S.6.121}$$

If we choose to eliminate x_4 first, i.e. compute

$$p(x_1, x_2, x_3, x_5, x_6, x_7) = \sum_{x_4} p(x_1, \dots, x_7) \tag{S.6.122}$$

$$\propto \sum_{x_4} \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\phi_f(x_7, x_4) \tag{S.6.123}$$

we cannot pull any of the factors out of the sum since each of them depends on x_4. This means the cost to sum out x_4 for all combinations of the six variables $(x_1, x_2, x_3, x_5, x_6, x_7)$ is K^7. Moreover, the new factor

$$\tilde{\phi}(x_1, x_2, x_3, x_5, x_6, x_7) = \sum_{x_4} \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\phi_f(x_7, x_4) \tag{S.6.124}$$

does not factorise anymore so that subsequent variable eliminations will be expensive too.

(b) Propose an elimination ordering that achieves $O(K^2)$ computational cost per variable elimination and explain why it does so.

Solution. Any ordering where x_4 is eliminated last will do. At any stage, elimination of one of the variables x_2, x_3, x_5, x_6, x_7 is then a $O(K^2)$ operation. This is because e.g.

$$p(x_1, \dots, x_6) = \sum_{x_7} p(x_1, \dots, x_7) \tag{S.6.125}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4) \underbrace{\sum_{x_7} \phi_f(x_7, x_4)}_{\tilde{\phi}_7(x_4)} \tag{S.6.126}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\phi_e(x_6, x_4)\tilde{\phi}_7(x_4) \tag{S.6.127}$$

where computing $\tilde{\phi}_7(x_4)$ for all values of x_4 is $O(K^2)$. Further,

$$p(x_1, \dots, x_5) = \sum_{x_6} p(x_1, \dots, x_6) \tag{S.6.128}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\tilde{\phi}_7(x_4) \sum_{x_6} \phi_e(x_6, x_4) \tag{S.6.129}$$

$$\propto \phi_a(x_1, x_4)\phi_b(x_2, x_4)\phi_c(x_3, x_4)\phi_d(x_5, x_4)\tilde{\phi}_7(x_4)\tilde{\phi}_6(x_4), \tag{S.6.130}$$

where computation of $\tilde{\phi}_6(x_4)$ for all values of x_4 is again $O(K^2)$. Continuing in this manner, one obtains

$$p(x_1, x_4) \propto \phi_a(x_1, x_4)\tilde{\phi}_2(x_4)\tilde{\phi}_3(x_4)\tilde{\phi}_5(x_4)\tilde{\phi}_6(x_4)\tilde{\phi}_7(x_4). \tag{S.6.131}$$

where each derived factor $\tilde{\phi}$ has $O(K^2)$ cost. Summing out x_4 and normalising the pmf is again a $O(K^2)$ operation.

Chapter 7

Inference for Hidden Markov Models

Exercises

7.1 Predictive distributions for hidden Markov models

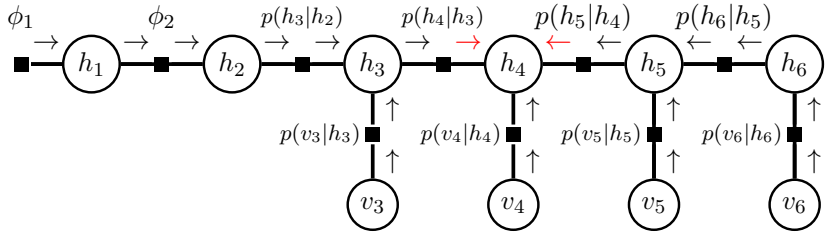
For the hidden Markov model

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

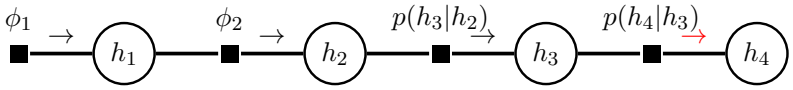
assume you have observations for v_i, $i = 1, \ldots, u < d$.

(a) Use message passing to compute $p(h_t|v_{1:u})$ for $u < t \leq d$. For the sake of concreteness, you may consider the case $d = 6, u = 2, t = 4$.

Solution. The factor graph for $d = 6, u = 2$, with messages that are required for the computation of $p(h_t|v_{1:u})$ for $t = 4$, is as follows.



The messages from the unobserved visibles v_i to their corresponding h_i, e.g. v_3 to h_3, are all one. Moreover, the message from the $p(h_5|h_4)$ node to h_4 equals one as well. This is because all involved factors, $p(v_i|h_i)$ and $p(h_i|h_{i-1})$, sum to one. Hence the factor graph reduces to a chain:



Since the variable nodes copy the messages in case of a chain, we only show the factor-to-variable messages.

The graph shows that we are essentially in the same situation as in filtering, with the difference that we use the factors $p(h_s|h_{s-1})$ for $s \geq u + 1$. Hence, we can use filtering to compute the messages until time $s = u$ and then compute the further messages with the $p(h_s|h_{s-1})$ as factors. This gives the following algorithm:

1. Compute $\alpha(h_u)$ by filtering.
2. For $s = u + 1, \ldots, t$, compute

$$\alpha(h_s) = \sum_{h_{s-1}} p(h_s|h_{s-1})\alpha(h_{s-1}) \tag{S.7.1}$$

3. The required predictive distribution is

$$p(h_t|v_{1:u}) = \frac{1}{Z}\alpha(h_t) \qquad Z = \sum_{h_t} \alpha(h_t) \tag{S.7.2}$$

For $s \geq u+1$, we have that

$$\sum_{h_s} \alpha(h_s) = \sum_{h_s} \sum_{h_{s-1}} p(h_s|h_{s-1})\alpha(h_{s-1}) \tag{S.7.3}$$

$$= \sum_{h_{s-1}} \alpha(h_{s-1}) \tag{S.7.4}$$

since $p(h_s|h_{s-1})$ is normalised. This means that the normalising constant Z above equals

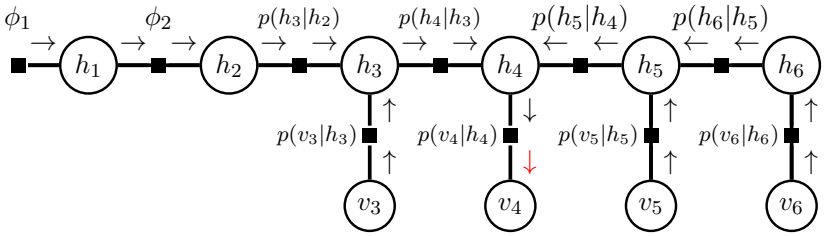
$$Z = \sum_{h_u} \alpha(h_u) = p(v_{1:u}) \tag{S.7.5}$$

which is the likelihood.

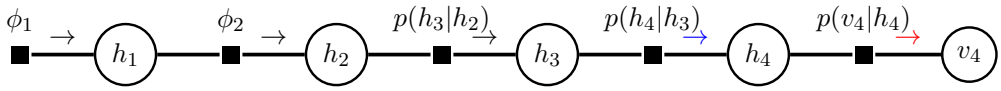
For filtering, we have seen that $\alpha(h_s) \propto p(h_s|v_{1:s})$, $s \leq u$. The $\alpha(h_s)$ for all $s > u$ are proportional to $p(h_s|v_{1:u})$. This may be seen by noting that the above arguments hold for any $t > u$.

(b) Use message passing to compute $p(v_t|v_{1:u})$ for $u < t \leq d$. For the sake of concreteness, you may consider the case $d = 6, u = 2, t = 4$.

Solution. The factor graph for $d = 6, u = 2$, with messages that are required for the computation of $p(v_t|v_{1:u})$ for $t = 4$, is as follows.



Due to the normalised factors, as above, the messages to the right of h_t are all one. Moreover the messages that go up from the v_i to the $h_i, i \neq t$, are also all one. Hence the graph simplifies to a chain.



The message in blue is proportional to $p(h_t|v_{1:u})$ computed in question (a). Thus assume that we have computed $p(h_t|v_{1:u})$. The predictive distribution on the level of the visibles thus is

$$p(v_t|v_{1:u}) = \sum_{h_t} p(v_t|h_t)p(h_t|v_{1:u}). \tag{S.7.6}$$

This follows from message passing since the last node (h_4 in the graph) just copies the (normalised) message and the next factor equals $p(v_t|h_t)$.

An alternative derivation follows from basic definitions and operations, together with the independencies in HMMs:

$$\text{(sum rule)} \qquad p(v_t|v_{1:u}) = \sum_{h_t} p(v_t, h_t|v_{1:u}) \tag{S.7.7}$$

$$\text{(product rule)} \qquad = \sum_{h_t} p(v_t|h_t, v_{1:u}) p(h_t|v_{1:u}) \tag{S.7.8}$$

$$(v_t \perp\!\!\!\perp v_{1:u} \mid h_t) \qquad = \sum_{h_t} p(v_t|h_t) p(h_t|v_{1:u}) \tag{S.7.9}$$

7.2 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

$$\hat{\mathbf{h}} = \underset{h_1,\ldots,h_t}{\operatorname{argmax}}\, p(h_{1:t}|v_{1:t}) \tag{7.1}$$

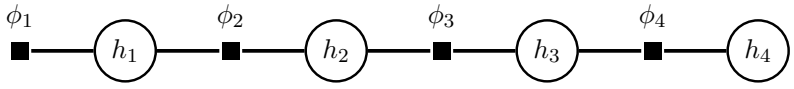
Assume that the latent variables h_i can take K different values, e.g. $h_i \in \{0, \ldots, K-1\}$. The resulting algorithm is known as Viterbi algorithm.

Solution. We first form the factors

$$\phi_1(h_1) = p(v_1|h_1)p(h_1) \qquad \phi_2(h_1, h_2) = p(v_2|h_2)p(h_2|h_1) \tag{S.7.10}$$

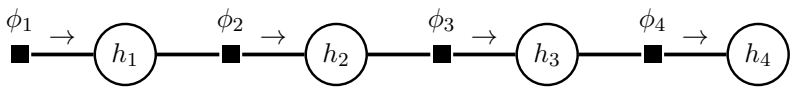
$$\ldots \qquad \phi_t(h_{t-1}, h_t) = p(v_t|h_t)p(h_t|h_{t-1}) \tag{S.7.11}$$

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 root. 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 the root 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

$$\gamma_{\phi_i\to h_i}(h_i) = \max_{h_{i-1}} \log \phi_i(h_{i-1}, h_i) + \gamma_{h_{i-1}\to\phi_i}(h_{i-1}) \tag{S.7.12}$$

$$= \max_{h_{i-1}} \log \phi_i(h_{i-1}, h_i) + \gamma_{\phi_{i-1}\to h_{i-1}}(h_{i-1}) \tag{S.7.13}$$

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

$$V_1(h_1) = \log p(v_1|h_1) + \log p(h_1) \tag{S.7.14}$$

$$V_i(h_i) = \max_{h_{i-1}} \log \phi_i(h_{i-1}, h_i) + V_{i-1}(h_{i-1}) \qquad i = 2, \ldots, t \tag{S.7.15}$$

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 $0, \ldots, K-1$, the above equations can be written as

$$v_{1,k} = \log p(v_1|k) + \log p(k) \qquad k = 0, \ldots, K-1 \tag{S.7.16}$$

$$v_{i,k} = \max_{m\in 0,\ldots,K-1} \log \phi_i(m, k) + v_{i-1,m} \qquad k = 0, \ldots, K-1, \quad i = 2, \ldots, t, \tag{S.7.17}$$

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

$$\gamma_i^*(h_i) = \operatorname*{argmax}_{h_{i-1}} \log \phi_i(h_{i-1}, h_i) + V_{i-1}(h_{i-1}) \tag{S.7.18}$$

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

$$\gamma_{i,k}^* = \operatorname*{argmax}_{m\in 0,\ldots,K-1} \log \phi_i(m, k) + v_{i-1,m} \tag{S.7.19}$$

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

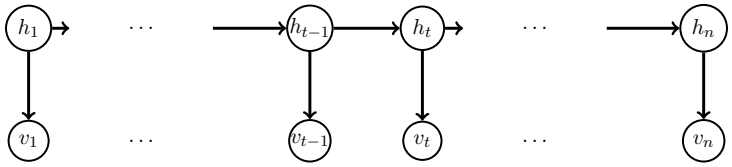
$$\hat{h}_t = \operatorname*{argmax}_k v_{t,k} \tag{S.7.20}$$

$$\hat{h}_i = \gamma_{i+1,\hat{h}_{i+1}}^* \qquad i = t-1, \ldots, 1 \tag{S.7.21}$$

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

7.3 Forward filtering backward sampling for hidden Markov models

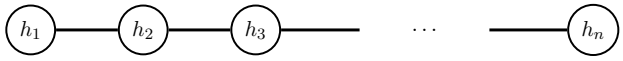
Consider the hidden Markov model specified by the following DAG.



We assume that have already run the alpha-recursion (filtering) and can compute $p(h_t|v_{1:t})$ for all t. The goal is now to generate samples $p(h_1, \ldots, h_n|v_{1:n})$, i.e. entire trajectories $(h_1, \ldots, h_n)$ from the posterior. Note that this is not the same as sampling from the n filtering distributions $p(h_t|v_{1:t})$. Moreover, compared to the Viterbi algorithm, the sampling approach generates samples from the full posterior rather than just returning the most probable state and its corresponding probability.

(a) Show that $p(h_1, \ldots, h_n|v_{1:n})$ forms a first-order Markov chain.

Solution. There are several ways to show this. The simplest is to notice that the undirected graph for the hidden Markov model is the same as the DAG but with the arrows removed as there are no colliders in the DAG. Moreover, conditioning corresponds to removing nodes from an undirected graph. This leaves us with a chain that connects the h_i.

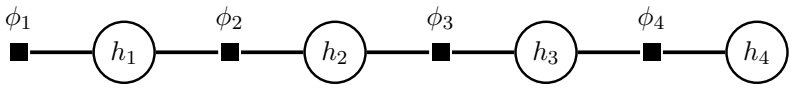


By graph separation, we see that $p(h_1, \ldots, h_n|v_{1:n})$ forms a first-order Markov chain so that e.g. $h_{1:t-1} \perp\!\!\!\perp h_{t+1:n}|h_t$ (past independent from the future given the present).

(b) Since $p(h_1, \ldots, h_n|v_{1:n})$ is a first-order Markov chain, it suffices to determine $p(h_{t-1}|h_t, v_{1:n})$, the probability mass function for h_{t-1} given h_t and all the data $v_{1:n}$. Use message passing to show that

$$p(h_{t-1}, h_t|v_{1:n}) \propto \alpha(h_{t-1})\beta(h_t)p(h_t|h_{t-1})p(v_t|h_t) \tag{7.2}$$

Solution. Since all visibles are in the conditioning set, i.e. assumed observed, we can represent the conditional model $p(h_1, \ldots, h_n|v_{1:n})$ as a chain factor tree, e.g. as follows in case of $n = 4$



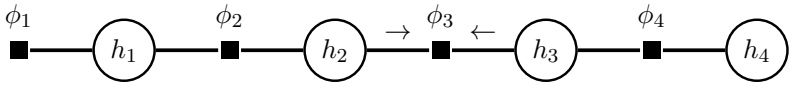
Combining the emission distributions $p(v_s|h_s)$ (and marginal $p(h_1)$) with the transition distributions $p(h_s|h_{s-1})$ we obtain the factors

$$\phi_1(h_1) = p(h_1)p(v_1|h_1) \tag{S.7.22}$$

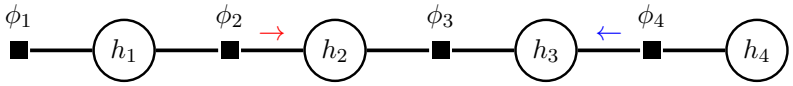
$$\phi_s(h_{s-1}, h_s) = p(h_s|h_{s-1})p(v_s|h_s) \quad \text{for } t = 2, \ldots, n \tag{S.7.23}$$

We see from the factor tree that h_{t-1} and h_t are neighbours, being attached to the same factor node $\phi_t(h_{t-1}, h_t)$, e.g. ϕ_3 in case of $p(h_2, h_3|v_{1:4})$.

By the rules of message passing, the joint $p(h_{t-1}, h_t|v_{1:n})$ is thus proportional to ϕ_t times the messages into ϕ_t. The following graph shows the messages for the case of $p(h_2, h_3|v_{1:4})$.



Since the variable nodes only receive single messages from any direction, they copy the messages so that the messages into ϕ_t are given by $\alpha(h_{t-1})$ and $\beta(h_t)$ shown below in red and blue, respectively.



Hence,

$$p(h_{t-1}, h_t|v_{1:n}) \propto \alpha(h_{t-1})\beta(h_t)\phi_t(h_{t-1}, h_t) \tag{S.7.24}$$
$$\propto \alpha(h_{t-1})\beta(h_t)p(h_t|h_{t-1})p(v_t|h_t) \tag{S.7.25}$$

which is the result that we want to show.

(c) Show that $p(h_{t-1}|h_t, v_{1:n}) = \frac{\alpha(h_{t-1})}{\alpha(h_t)}p(h_t|h_{t-1})p(v_t|h_t)$.

Solution. The conditional $p(h_{t-1}|h_t, v_{1:n})$ can be written as the ratio

$$p(h_{t-1}|h_t, v_{1:n}) = \frac{p(h_{t-1}, h_t|v_{1:n})}{p(h_t|v_{1:n})}. \tag{S.7.26}$$

Above, we have shown that the numerator satisfies

$$p(h_{t-1}, h_t|v_{1:n}) \propto \alpha(h_{t-1})\beta(h_t)p(h_t|h_{t-1})p(v_t|h_t). \tag{S.7.27}$$

The denominator $p(h_t|v_{1:n})$ is proportional to $\alpha(h_t)\beta(h_t)$ since it is the smoothing distribution than can be determined via the alpha-beta recursion.

Normally, we needed to sum the messages over all values of (h_{t-1}, h_t) to find the normalising constant of the numerator. For the denominator, we had to sum over all values of h_t. Next, I will argue qualitatively that this summation is not needed; the normalising constants are both equal to $p(v_{1:t})$. A more mathematical argument is given below.

We started with a factor graph and factors that represent the joint $p(h_{1:n}, v_{1:n})$. The conditional $p(h_{1:n}, v_{1:n})$ equals

$$p(h_{1:n}|v_{1:n}) = \frac{p(h_{1:n}, v_{1:n})}{p(v_{1:n})} \tag{S.7.28}$$

Message passing is variable elimination. Hence, when computing $p(h_t|v_{1:n})$ as $\alpha(h_t)\beta(h_t)$ from a factor graph for $p(h_{1:n}, v_{1:n})$, we only need to divide by $p(v_{1:n})$ for normalisation; explicitly summing out h_t is not needed. In other words,

$$p(h_t|v_{1:n}) = \frac{\alpha(h_t)\beta(h_t)}{p(v_{1:n})}. \tag{S.7.29}$$

Similarly, $p(h_{t-1}, h_t|v_{1:n})$ is also obtained from (S.7.28) by marginalisation/variable elimination. Again, when computing $p(h_{t-1}, h_t|v_{1:n})$ as $\alpha(h_{t-1})\beta(h_t)p(h_t|h_{t-1})p(v_t|h_t)$

from a factor graph for $p(h_{1:n}, v_{1:n})$, we do not need to explicitly sum over all values of h_t and h_{t-1} for normalisation. The definition of the factors in the factor graph together with (S.7.28) shows that we can simply divide by $p(v_{1:n})$. This gives

$$p(h_{t-1}, h_t|v_{1:n}) = \frac{1}{p(v_{1:n})}\alpha(h_{t-1})\beta(h_t)p(h_t|h_{t-1})p(v_t|h_t). \tag{S.7.30}$$

The desired conditional thus is

$$p(h_{t-1}|h_t, v_{1:n}) = \frac{p(h_{t-1}, h_t|v_{1:n})}{p(h_t|v_{1:n})} \tag{S.7.31}$$

$$= \frac{\alpha(h_{t-1})\beta(h_t)p(h_t|h_{t-1})p(v_t|h_t)}{\alpha(h_t)\beta(h_t)} \tag{S.7.32}$$

$$= \frac{\alpha(h_{t-1})p(h_t|h_{t-1})p(v_t|h_t)}{\alpha(h_t)} \tag{S.7.33}$$

which is the result that we wanted to show. Note that $\beta(h_t)$ cancels out and that $p(h_{t-1}|h_t, v_{1:n})$ only involves the α's, the (forward) transition distribution $p(h_t|h_{t-1})$ and the emission distribution at time t.

Alternative solution: An alternative, mathematically rigorous solution is as follows. The conditional $p(h_{t-1}|h_t, v_{1:n})$ can be written as the ratio

$$p(h_{t-1}|h_t, v_{1:n}) = \frac{p(h_{t-1}, h_t|v_{1:n})}{p(h_t|v_{1:n})}. \tag{S.7.34}$$

We first determine the denominator. From the properties of the alpha and beta recursion, we know that

$$\alpha(h_t) = p(h_t, v_{1:t}) \qquad \beta(h_t) = p(v_{t+1:n}|h_t) \tag{S.7.35}$$

Using that $v_{t+1:n} \perp\!\!\!\perp v_{1:t}|h_t$, we can thus express the denominator $p(h_t|v_{1:n})$ as

$$p(h_t|v_{1:n}) = \frac{p(h_t, v_{1:n})}{p(v_{1:n})} \tag{S.7.36}$$

$$= \frac{p(h_t, v_{1:t})p(v_{t+1:n}|h_t)}{p(v_{1:n})} \tag{S.7.37}$$

$$= \frac{\alpha(h_t)\beta(h_t)}{p(v_{1:n})} \tag{S.7.38}$$

For the numerator, we have

$$p(h_{t-1}, h_t|v_{1:n}) = \frac{p(h_{t-1}, h_t, v_{1:n})}{p(v_{1:n})} \tag{S.7.39}$$

$$= \frac{p(h_{t-1}, v_{1:t-1}, h_t, v_{t:n})}{p(v_{1:n})} \tag{S.7.40}$$

$$= \frac{p(h_{t-1}, v_{1:t-1})p(h_t, v_{t:n}|h_{t-1}, v_{1:t-1})}{p(v_{1:n})} \tag{S.7.41}$$

$$= \frac{p(h_{t-1}, v_{1:t-1})p(h_t, v_{t:n}|h_{t-1})}{p(v_{1:n})} \qquad (\text{using } h_t, v_{1:t} \perp\!\!\!\perp v_{1:t-1}|h_{t-1}) \tag{S.7.42}$$

$$= \frac{\alpha(h_{t-1})p(h_t, v_{t:n}|h_{t-1})}{p(v_{1:n})} \qquad (\text{using } \alpha(h_{t-1}) = p(h_{t-1}, v_{1:t-1})) \tag{S.7.43}$$

With the product rule, we have $p(h_t, v_{t:n}|h_{t-1}) = p(v_t|h_t, h_{t-1}, v_{t+1:n})p(h_t, v_{t+1:n}|h_{t-1})$ so that

$$p(h_{t-1}, h_t|v_{1:n}) = \frac{\alpha(h_{t-1})p(v_t|h_t, h_{t-1}, v_{t+1:n})p(h_t, v_{t+1:n}|h_{t-1})}{p(v_{1:n})} \tag{S.7.44}$$

$$= \frac{\alpha(h_{t-1})p(v_t|h_t)p(h_t, v_{t+1:n}|h_{t-1})}{p(v_{1:n})} \qquad (\text{using } v_t \perp\!\!\!\perp h_{t-1}, v_{t+1:n}|h_t) \tag{S.7.45}$$

$$= \frac{\alpha(h_{t-1})p(v_t|h_t)p(h_t|h_{t-1})p(v_{t+1:n}|h_{t-1}, h_t)}{p(v_{1:n})} \tag{S.7.46}$$

Hence

$$p(h_{t-1}, h_t|v_{1:n}) = \frac{\alpha(h_{t-1})p(v_t|h_t)p(h_t|h_{t-1})p(v_{t+1:n}|h_t)}{p(v_{1:n})} \qquad (\text{using } v_{t+1:n} \perp\!\!\!\perp h_{t-1}|h_t) \tag{S.7.47}$$

$$= \frac{\alpha(h_{t-1})p(v_t|h_t)p(h_t|h_{t-1})\beta(h_t)}{p(v_{1:n})} \qquad (\text{using } \beta(h_t) = p(v_{t+1:n}|h_t)) \tag{S.7.48}$$

The desired conditional thus is

$$p(h_{t-1}|h_t, v_{1:n}) = \frac{p(h_{t-1}, h_t|v_{1:n})}{p(h_t|v_{1:n})} \tag{S.7.49}$$

$$= \frac{\alpha(h_{t-1})\beta(h_t)p(h_t|h_{t-1})p(v_t|h_t)}{\alpha(h_t)\beta(h_t)} \tag{S.7.50}$$

$$= \frac{\alpha(h_{t-1})p(h_t|h_{t-1})p(v_t|h_t)}{\alpha(h_t)} \tag{S.7.51}$$

which is the result that we wanted to show.

We thus obtain the following algorithm to generate samples from $p(h_1, \ldots, h_n|v_{1:n})$:

1. Run the alpha-recursion (filtering) to determine all $\alpha(h_t)$ forward in time for $t = 1, \ldots, n$.
2. Sample h_n from $p(h_n|v_{1:n}) \propto \alpha(h_n)$
3. Go backwards in time using

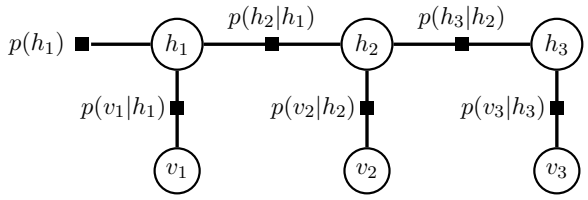
$$p(h_{t-1}|h_t, v_{1:n}) = \frac{\alpha(h_{t-1})}{\alpha(h_t)}p(h_t|h_{t-1})p(v_t|h_t) \tag{7.3}$$

   to generate samples $h_{t-1}|h_t, v_{1:n}$ for $t = n, \ldots, 2$.

This algorithm is known as forward filtering backward sampling (FFBS).

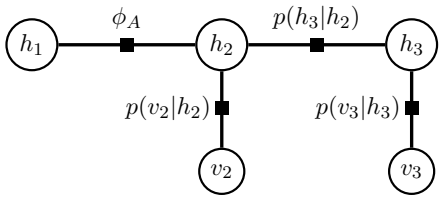
7.4 Prediction exercise

Consider a hidden Markov model with three visibles v_1, v_2, v_3 and three hidden variables h_1, h_2, h_3 which can be represented with the following factor graph:



This question is about computing the predictive probability $p(v_3 = 1|v_1 = 1)$.

(a) The factor graph below represents $p(h_1, h_2, h_3, v_2, v_3 \mid v_1 = 1)$. Provide an equation that defines ϕ_A in terms of the factors in the factor graph above.



Solution. $\phi_A(h_1, h_2) \propto p(v_1|h_1)p(h_1)p(h_2|h_1)$ with $v_1 = 1$.

(b) Assume further that all variables are binary, $h_i \in \{0, 1\}$, $v_i \in \{0, 1\}$; that $p(h_1 = 1) = 0.5$, and that the transition and emission distributions are, for all i, given by:

$p(h_{i+1}\|h_i)$	h_{i+1}	h_i
0	0	0
1	1	0
1	0	1
0	1	1

$p(v_i\|h_i)$	v_i	h_i
0.6	0	0
0.4	1	0
0.4	0	1
0.6	1	1

Compute the numerical values of the factor ϕ_A.

(c) Given the definition of the transition and emission probabilities, we have $\phi_A(h_1, h_2) = 0$ if $h_1 = h_2$. For $h_1 = 0, h_2 = 1$, we obtain

$$\begin{aligned}
\phi_A(h_1 = 0, h_2 = 1) &= p(v_1 = 1|h_1 = 0)p(h_1 = 0)p(h_2 = 1|h_1 = 0) && \text{(S.7.52)}\\
&= 0.4 \cdot 0.5 \cdot 1 && \text{(S.7.53)}\\
&= \frac{4}{10} \cdot \frac{1}{2} && \text{(S.7.54)}\\
&= \frac{2}{10} = 0.2 && \text{(S.7.55)}
\end{aligned}$$

For $h_1 = 1, h_2 = 0$, we obtain

$$\phi_A(h_1 = 1, h_2 = 0) = p(v_1 = 1|h_1 = 1)p(h_1 = 1)p(h_2 = 0|h_1 = 1) \tag{S.7.56}$$
$$= 0.6 \cdot 0.5 \cdot 1 \tag{S.7.57}$$
$$= \frac{6}{10} \cdot \frac{1}{2} \tag{S.7.58}$$
$$= \frac{3}{10} = 0.3 \tag{S.7.59}$$

Hence

$\phi_A(h_1, h_2)$	h_1	h_2
0	0	0
0.3	1	0
0.2	0	1
0	1	1

(d) Denote the message from variable node h_2 to factor node $p(h_3|h_2)$ by $\alpha(h_2)$. Use message passing to compute $\alpha(h_2)$ for $h_2 = 0$ and $h_2 = 1$. Report the values of any intermediate messages that need to be computed for the computation of $\alpha(h_2)$.

Solution. The message from h_1 to ϕ_A is one. The message from ϕ_A to h_2 is

$$\mu_{\phi_A \to h_2}(h_2 = 0) = \sum_{h_1} \phi_A(h_1, h_2 = 0) \tag{S.7.60}$$
$$= 0.3 \tag{S.7.61}$$
$$\mu_{\phi_A \to h_2}(h_2 = 1) = \sum_{h_1} \phi_A(h_1, h_2 = 1) \tag{S.7.62}$$
$$= 0.2 \tag{S.7.63}$$

Since v_2 is not observed and $p(v_2|h_2)$ normalised, the message from $p(v_2|h_2)$ to h_2 equals one.

This means that the message from h_2 to $p(h_3|h_2)$, which is $\alpha(h_2)$ equals $\mu_{\phi_A \to h_2}(h_2)$, i.e.

$$\alpha(h_2 = 0) = 0.3 \tag{S.7.64}$$
$$\alpha(h_2 = 1) = 0.2 \tag{S.7.65}$$

(e) With $\alpha(h_2)$ defined as above, use message passing to show that the predictive probability $p(v_3 = 1|v_1 = 1)$ can be expressed in terms of $\alpha(h_2)$ as

$$p(v_3 = 1|v_1 = 1) = \frac{x\alpha(h_2 = 1) + y\alpha(h_2 = 0)}{\alpha(h_2 = 1) + \alpha(h_2 = 0)} \tag{7.4}$$

and report the values of x and y.

Solution. Given the definition of $p(h_3|h_2)$, the message $\mu_{p(h_3|h_2)\to h_3}(h_3)$ is

$$\mu_{p(h_3|h_2)\to h_3}(h_3=0)=\alpha(h_2=1) \tag{S.7.66}$$
$$\mu_{p(h_3|h_2)\to h_3}(h_3=1)=\alpha(h_2=0) \tag{S.7.67}$$

The variable node h_3 copies the message so that we have

$$\mu_{p(v_3|h_3)\to v_3}(v_3=0)=\sum_{h_3}p(v_3=0|h_3)\mu_{p(h_3|h_2)\to h_3}(h_3) \tag{S.7.68}$$
$$=p(v_3=0|h_3=0)\alpha(h_2=1)+p(v_3=0|h_3=1)\alpha(h_2=0) \tag{S.7.69}$$
$$=0.6\alpha(h_2=1)+0.4\alpha(h_2=0) \tag{S.7.70}$$
$$\mu_{p(v_3|h_h3)\to v_3}(v_3=1)=\sum_{h_3}p(v_3=1|h_3))\mu_{p(h_3|h_2)\to h_3}(h_3) \tag{S.7.71}$$
$$=p(v_3=1|h_3=0)\alpha(h_2=1)+p(v_3=1|h_3=1)\alpha(h_2=0) \tag{S.7.72}$$
$$=0.4\alpha(h_2=1)+0.6\alpha(h_2=0) \tag{S.7.73}$$

We thus have

$$p(v_3=1|v_1=1)=\frac{0.4\alpha(h_2=1)+0.6\alpha(h_2=0)}{0.4\alpha(h_2=1)+0.6\alpha(h_2=0)+0.6\alpha(h_2=1)+0.4\alpha(h_2=0)} \tag{S.7.74}$$
$$=\frac{0.4\alpha(h_2=1)+0.6\alpha(h_2=0)}{\alpha(h_2=1)+\alpha(h_2=0)} \tag{S.7.75}$$

The requested x and y are thus: $x=0.4$, $y=0.6$.

(f) Compute the numerical value of $p(v_3=1|v_1=1)$.

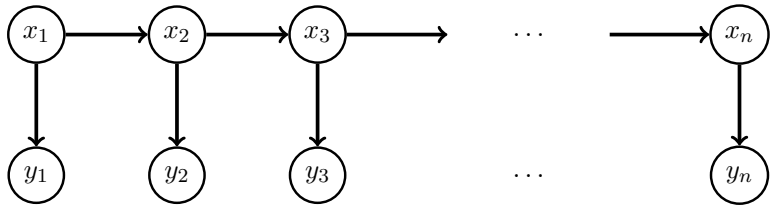
Solution. Inserting the numbers gives $\alpha(h_2=0)+\alpha(h_2=1)=5/10=1/2$ so that

$$p(v_3=1|v_1=1)=\frac{0.4\cdot 0.2+0.6\cdot 0.3}{\frac{1}{2}} \tag{S.7.76}$$
$$=2\cdot\left(\frac{4}{10}\cdot\frac{2}{10}+\frac{6}{10}\frac{3}{10}\right) \tag{S.7.77}$$
$$=\frac{4}{10}\cdot\frac{4}{10}+\frac{6}{10}\frac{6}{10} \tag{S.7.78}$$
$$=\frac{1}{100}(16+36) \tag{S.7.79}$$
$$=\frac{1}{100}52 \tag{S.7.80}$$
$$=\frac{52}{100}=0.52 \tag{S.7.81}$$

7.5 Hidden Markov models and change of measure

We take here a change of measure perspective on the alpha-recursion.

Consider the following directed graph for a hidden Markov model where the y_i correspond to observed (visible) variables and the x_i to unobserved (hidden/latent) variables.



The joint model for $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ thus is

$$p(\mathbf{x}, \mathbf{y}) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{n} p(y_i|x_i). \tag{7.5}$$

(a) Show that

$$p(x_1, \ldots, x_n, y_1, \ldots, y_t) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t} p(y_i|x_i) \tag{7.6}$$

for $t = 0, \ldots, n$. We take the case $t = 0$ to correspond to $p(x_1, \ldots, x_n)$,

$$p(x_1, \ldots, x_n) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}). \tag{7.7}$$

Solution. The result follows by integrating/summing out $y_{t+1} \ldots n$.

$$p(x_1, \ldots, x_n, y_1, \ldots, y_t) = \int p(x_1, \ldots, x_n, y_1, \ldots, y_n) \mathrm{d}y_{t+1} \ldots \mathrm{d}y_n \tag{S.7.82}$$

$$= \int p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{n} p(y_i|x_i) \mathrm{d}y_{t+1} \ldots \mathrm{d}y_n \tag{S.7.83}$$

$$= p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t} p(y_i|x_i) \int \prod_{i=t+1}^{n} p(y_i|x_i) \mathrm{d}y_{t+1} \ldots \mathrm{d}y_n \tag{S.7.84}$$

$$= p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t} p(y_i|x_i) \prod_{i=t+1}^{n} \underbrace{\int p(y_i|x_i) \mathrm{d}y_i}_{=1} \tag{S.7.85}$$

$$= p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t} p(y_i|x_i) \tag{S.7.86}$$

The result for $p(x_1, \ldots, x_n)$ is obtained when we integrate out all y's.

(b) Show that $p(x_1, \ldots, x_n|y_1, \ldots, y_t)$, $t = 0, \ldots, n$, factorises as

$$p(x_1, \ldots, x_n|y_1, \ldots, y_t) \propto p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t} g_i(x_i) \tag{7.8}$$

where $g_i(x_i) = p(y_i|x_i)$ for a fixed value of y_i, and that its normalising constant Z_t equals the likelihood $p(y_1, \ldots, y_t)$

Solution. The result follows from the basic definition of the conditional

$$p(x_1, \ldots, x_n|y_1, \ldots, y_t) = \frac{p(x_1, \ldots, x_n, y_1, \ldots, y_t)}{p(y_1, \ldots, y_t)} \tag{S.7.87}$$

together with the expression for $p(x_1, \ldots, x_n, y_1, \ldots, y_t)$ when the y_i are kept fixed.

(c) Denote $p(x_1, \ldots, x_n|y_1, \ldots, y_t)$ by $p_t(x_1, \ldots, x_n)$. The index $t \leq n$ thus indicates the time of the last y-variable we are conditioning on. Show the following recursion for $1 \leq t \leq n$:

$$p_{t-1}(x_1, \ldots, x_t) = \begin{cases} p(x_1) & \text{if } t = 1 \\ p_{t-1}(x_1, \ldots, x_{t-1}) p(x_t|x_{t-1}) & \text{otherwise} \end{cases} \quad \text{(extension)} \tag{7.9}$$

$$p_t(x_1, \ldots, x_t) = \frac{1}{Z_t} p_{t-1}(x_1, \ldots, x_t) g_t(x_t) \quad \text{(change of measure)} \tag{7.10}$$

$$Z_t = \int p_{t-1}(x_t) g_t(x_t) \mathrm{d}x_t \tag{7.11}$$

By iterating from $t = 1$ to $t = n$, we can thus recursively compute $p(x_1, \ldots, x_n|y_1, \ldots, y_n)$, including its normalising constant Z_n, which equals the likelihood $Z_n = p(y_1, \ldots, y_n)$

Solution. We start with (7.8) which shows that by definition of $p_t(x_1, \ldots, x_n)$ we have

$$p_t(x_1, \ldots, x_n) = p(x_1, \ldots, x_n|y_1, \ldots, y_t) \tag{S.7.88}$$

$$\propto p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t} g_i(x_i) \tag{S.7.89}$$

For $t = 1$, we thus have

$$p_1(x_1, \ldots, x_n) \propto p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) g_1(x_1) \tag{S.7.90}$$

Integrating out $x_2, \ldots, x_n$ gives

$$p_1(x_1) = \int p_1(x_1, \ldots, x_n) \mathrm{d}x_2 \ldots \mathrm{d}x_n \tag{S.7.91}$$

$$\propto \int p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) g_1(x_1) \mathrm{d}x_2 \ldots \mathrm{d}x_n \tag{S.7.92}$$

$$\propto p(x_1) g_1(x_1) \int \prod_{i=2}^{n} p(x_i|x_{i-1}) \mathrm{d}x_2 \ldots \mathrm{d}x_n \tag{S.7.93}$$

$$\propto p(x_1) g_1(x_1) \underbrace{\prod_{i=2}^{n} \int p(x_i|x_{i-1}) \mathrm{d}x_i}_{=1} \tag{S.7.94}$$

$$\propto p(x_1) g_1(x_1) \tag{S.7.95}$$

The normalising constant is

$$Z_1 = \int p(x_1) g_1(x_1) \mathrm{d}x_1 \tag{S.7.96}$$

This establishes the result for $t = 1$.

From (7.8), we further have

$$p_{t-1}(x_1, \ldots, x_n) = p(x_1, \ldots, x_n | y_1, \ldots, y_{t-1}) \tag{S.7.97}$$

$$\propto p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t-1} g_i(x_i) \tag{S.7.98}$$

Integrating out $x_{t+1}, \ldots, x_n$ thus gives

$$p_{t-1}(x_1, \ldots, x_t) = \int p_{t-1}(x_1, \ldots, x_n) \mathrm{d}x_{t+1} \ldots \mathrm{d}x_n \tag{S.7.99}$$

$$\propto \int p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \prod_{i=1}^{t-1} g_i(x_i) \mathrm{d}x_{t+1} \ldots \mathrm{d}x_n \tag{S.7.100}$$

$$\propto p(x_1) \prod_{i=2}^{t} p(x_i|x_{i-1}) \prod_{i=1}^{t-1} g_i(x_i) \int \prod_{i=t+1}^{n} p(x_i|x_{i-1}) \mathrm{d}x_{t+1} \ldots \mathrm{d}x_n \tag{S.7.101}$$

$$\propto p(x_1) \prod_{i=2}^{t} p(x_i|x_{i-1}) \prod_{i=1}^{t-1} g_i(x_i) \prod_{i=t+1}^{n} \int p(x_i|x_{i-1}) \mathrm{d}x_i \tag{S.7.102}$$

$$\propto p(x_1) \prod_{i=2}^{t} p(x_i|x_{i-1}) \prod_{i=1}^{t-1} g_i(x_i) \tag{S.7.103}$$

Noting that the product over the g_i does not involve x_t and that $p(x_t|x_{t-1})$ is a pdf, we have further

$$p_{t-1}(x_1, \ldots, x_{t-1}) = \int p_{t-1}(x_1, \ldots, x_t) \mathrm{d}x_t \tag{S.7.104}$$

$$\propto p(x_1) \prod_{i=2}^{t-1} p(x_i|x_{i-1}) \prod_{i=1}^{t-1} g_i(x_i) \tag{S.7.105}$$

Hence

$$p_{t-1}(x_1, \ldots, x_t) = p_{t-1}(x_1, \ldots, x_{t-1})p(x_t|x_{t-1}) \tag{S.7.106}$$

Note that we can have an equal sign since $p(x_t|x_{t-1})$ is a pdf and hence integrates to one. This is sometimes called the "extension" since the inputs for p_{t-1} are extended from $(x_1, \ldots, x_{t-1})$ to $x_1, \ldots, x_t$.

From (S.7.89), we further have

$$p_t(x_1, \ldots, x_n) \propto p_{t-1}(x_1, \ldots, x_n)g_t(x_t) \tag{S.7.107}$$

Integrating out $x_{t+1}, \ldots, x_n$ thus gives

$$p_t(x_1, \ldots, x_t) \propto p_{t-1}(x_1, \ldots, x_t)g_t(x_t) \tag{S.7.108}$$

This is a change of measure from $p_{t-1}(x_1, \ldots, x_t)$ to $p_t(x_1, \ldots, x_t)$. Note that $p_{t-1}(x_1, \ldots, x_t)$ only involves g_i, and hence observations y_i, up to index (time) $t-1$. The change of measure multiplies-in the additional factor $g_t(x_t) = p(y_t|x_t)$, and thereby incorporates the observation at index (time) t into the model.

The stated recursion is complete by computing the normalising constant Z_t for $p_t(x_1, \ldots, x_t)$, which equals

$$\begin{aligned} Z_t &= \int p_{t-1}(x_1, \ldots, x_t)g_t(x_t)\mathrm{d}x_1, \ldots \mathrm{d}x_t && \text{(S.7.109)} \\ &= \int g_t(x_t) \left[\int p_{t-1}(x_1, \ldots, x_t)\mathrm{d}x_1, \ldots \mathrm{d}x_{t-1} \right] \mathrm{d}x_t && \text{(S.7.110)} \\ &= \int g_t(x_t)p_{t-1}(x_t)\mathrm{d}x_t && \text{(S.7.111)} \end{aligned}$$

This recursion, and some slight generalisations, forms the basis for what is known as the "forward recursion" in particle filtering and sequential Monte Carlo. An excellent introduction to these topics is book (Chopin and Papaspiliopoulos, 2020).

(d) Use the recursion above to derive the following form of the alpha recursion:

$$\begin{aligned} p_{t-1}(x_{t-1}, x_t) &= p_{t-1}(x_{t-1})p(x_t|x_{t-1}) && \text{(extension)} && (7.12) \\ p_{t-1}(x_t) &= \int p_{t-1}(x_{t-1}, x_t)\mathrm{d}x_{t-1} && \text{(marginalisation)} && (7.13) \\ p_t(x_t) &= \frac{1}{Z_t} p_{t-1}(x_t)g_t(x_t) && \text{(change of measure)} && (7.14) \\ Z_t &= \int p_{t-1}(x_t)g_t(x_t)\mathrm{d}x_t && && (7.15) \end{aligned}$$

with $p_0(x_1) = p(x_1)$.

The term $p_t(x_t)$ corresponds to $\alpha(x_t)$ from the alpha-recursion after normalisation. Moreover, $p_{t-1}(x_t)$ is the predictive distribution for x_t given observations until time $t-1$. Multiplying $p_{t-1}(x_t)$ with $g_t(x_t)$ gives the new $\alpha(x_t)$. The term $g_t(x_t) = p(y_t|x_t)$ is sometimes called the "correction" term. We see here that the correction has the effect of a change of measure, changing the predictive distribution $p_{t-1}(x_t)$ into the filtering distribution $p_t(x_t)$.

Solution. Let $t > 1$. With (7.9), we have

$$p_{t-1}(x_{t-1}, x_t) = \int p_{t-1}(x_1, \ldots, x_t)\mathrm{d}x_1 \ldots \mathrm{d}x_{t-2} \tag{S.7.112}$$

$$= \int p_{t-1}(x_1, \ldots, x_{t-1})p(x_t|x_{t-1})\mathrm{d}x_1 \ldots \mathrm{d}x_{t-2} \tag{S.7.113}$$

$$= p(x_t|x_{t-1}) \int p_{t-1}(x_1, \ldots, x_{t-1})\mathrm{d}x_1 \ldots \mathrm{d}x_{t-2} \tag{S.7.114}$$

$$= p(x_t|x_{t-1})p_{t-1}(x_{t-1}) \tag{S.7.115}$$

which proves the "extension".

With (7.10), we have

$$p_t(x_t) = \int p_t(x_1, \ldots, x_t)\mathrm{d}x_1, \ldots \mathrm{d}x_{t-1} \tag{S.7.116}$$

$$= \frac{1}{Z_t} \int p_{t-1}(x_1, \ldots, x_t)g_t(x_t)\mathrm{d}x_1, \ldots \mathrm{d}x_{t-1} \tag{S.7.117}$$

$$= \frac{1}{Z_t} g_t(x_t) \int p_{t-1}(x_1, \ldots, x_t)\mathrm{d}x_1, \ldots \mathrm{d}x_{t-1} \tag{S.7.118}$$

$$= \frac{1}{Z_t} g_t(x_t)p_{t-1}(x_t) \tag{S.7.119}$$

which proves the "change of measure". Moreover, the normalising constant Z_t is the same as before. Hence completing the iteration until $t = n$ yields the likelihood $p(y_1, \ldots, y_n) = Z_n$ as a by-product of the recursion. The initialisation of the recursion with $p_0(x_1) = p(x_1)$ is also the same as above.

7.6 Kalman filtering

We here consider filtering for hidden Markov models with Gaussian transition and emission distributions. For simplicity, we assume one-dimensional hidden variables and observables. We denote the probability density function of a Gaussian random variable x with mean μ and variance σ^2 by $\mathcal{N}(x|\mu, \sigma^2)$,

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]. \tag{7.16}$$

The transition and emission distributions are assumed to be

$$p(h_s|h_{s-1}) = \mathcal{N}(h_s|A_s h_{s-1}, B_s^2) \tag{7.17}$$

$$p(v_s|h_s) = \mathcal{N}(v_s|C_s h_s, D_s^2). \tag{7.18}$$

The distribution $p(h_1)$ is assumed Gaussian with known parameters. The A_s, B_s, C_s, D_s are also assumed known.

(a) Show that h_s and v_s as defined in the following update and observation equations

$$h_s = A_s h_{s-1} + B_s \xi_s \tag{7.19}$$

$$v_s = C_s h_s + D_s \eta_s \tag{7.20}$$

follow the conditional distributions in (7.17) and (7.18). The random variables ξ_s and η_s are independent from the other variables in the model and follow a standard normal Gaussian distribution, e.g. $\xi_s \sim \mathcal{N}(\xi_s|0,1)$.
Hint: For two constants c_1 and c_2, $y = c_1 + c_2 x$ is Gaussian if x is Gaussian. In other words, an affine transformation of a Gaussian is Gaussian.

The equations mean that h_s is obtained by scaling h_{s-1} and by adding noise with variance B_s^2. The observed value v_s is obtained by scaling the hidden h_s and by corrupting it with Gaussian observation noise of variance D_s^2.

Solution. By assumption, ξ_s is Gaussian. Since we condition on h_{s-1}, $A_s h_{s-1}$ in (7.19) is a constant, and since B_s is a constant too, h_s is Gaussian.

What we have to show next is that (7.19) defines the same conditional mean and variance as the conditional Gaussian in (7.17): The conditional expectation of h_s given h_{s-1} is

$$\begin{aligned}
\mathbb{E}(h_s|h_{s-1}) &= A_s h_{s-1} + \mathbb{E}(B_s \xi_s) && \text{(since we condition on } h_{s-1}\text{)} && \text{(S.7.120)}\\
&= A_s h_{s-1} + B_s \mathbb{E}(\xi_s) && \text{(by linearity of expectation)} && \text{(S.7.121)}\\
&= A_s h_{s-1} && \text{(since } \xi_s \text{ has zero mean)} && \text{(S.7.122)}
\end{aligned}$$

The conditional variance of h_s given h_{s-1} is

$$\begin{aligned}
\mathbb{V}(h_s|h_{s-1}) &= \mathbb{V}(B_s \xi_s) && \text{(since we condition on } h_{s-1}\text{)} && \text{(S.7.123)}\\
&= B_s^2 \mathbb{V}(\xi_s) && \text{(by properties of the variance)} && \text{(S.7.124)}\\
&= B_s^2 && \text{(since } \xi_s \text{ has variance one)} && \text{(S.7.125)}
\end{aligned}$$

We see that the conditional mean and variance of h_s given h_{s-1} match those in (7.17). And since h_s given h_{s-1} is Gaussian as argued above, the result follows.

Exactly the same reasoning also applies to the case of (7.20). Conditional on h_s, v_s is Gaussian because it is an affine transformation of a Gaussian. The conditional mean of v_s given h_s is:

$$\begin{aligned}
\mathbb{E}(v_s|h_s) &= C_s h_s + \mathbb{E}(D_s \eta_s) && \text{(since we condition on } h_s\text{)} && \text{(S.7.126)}\\
&= C_s h_s + D_s \mathbb{E}(\eta_s) && \text{(by linearity of expectation)} && \text{(S.7.127)}\\
&= C_s h_s && \text{(since } \eta_s \text{ has zero mean)} && \text{(S.7.128)}
\end{aligned}$$

The conditional variance of v_s given h_s is

$$\begin{aligned}
\mathbb{V}(v_s|h_s) &= \mathbb{V}(D_s \eta_s) && \text{(since we condition on } h_s\text{)} && \text{(S.7.129)}\\
&= D_s^2 \mathbb{V}(\eta_s) && \text{(by properties of the variance)} && \text{(S.7.130)}\\
&= D_s^2 && \text{(since } \eta_s \text{ has variance one)} && \text{(S.7.131)}
\end{aligned}$$

Hence, conditional on h_s, v_s is Gaussian with mean and variance as in (7.18).

(b) Show that

$$\int \mathcal{N}(x|\mu, \sigma^2)\mathcal{N}(y|Ax, B^2)\mathrm{d}x \propto \mathcal{N}(y|A\mu, A^2\sigma^2 + B^2) \tag{7.21}$$

Hint: While this result can be obtained by integration, an approach that avoids this is as follows: First note that $\mathcal{N}(x|\mu, \sigma^2)\mathcal{N}(y|Ax, B^2)$ is proportional to the joint pdf of x and y. We can thus consider the integral to correspond to the computation of the marginal of y from the joint. Using the equivalence of Equations (7.17)-(7.18) and (7.19)-(7.20), and the fact that the weighted sum of two Gaussian random variables is a Gaussian random variable then allows one to obtain the result.

Solution. We follow the procedure outlined above. The two Gaussian densities correspond to the equations

$$x = \mu + \sigma\xi \tag{S.7.132}$$
$$y = Ax + B\eta \tag{S.7.133}$$

where ξ and η are independent standard normal random variables. The mean of y is

$$\mathbb{E}(y) = A\mathbb{E}(x) + B\mathbb{E}(\eta) \tag{S.7.134}$$
$$= A\mu \tag{S.7.135}$$

where we have use the linearity of expectation and $\mathbb{E}(\eta) = 0$. The variance of y is

$$\mathbb{V}(y) = \mathbb{V}(Ax) + \mathbb{V}(B\eta) \qquad \text{(since } x \text{ and } \eta \text{ are independent)} \tag{S.7.136}$$
$$= A^2\mathbb{V}(x) + B^2\mathbb{V}(\eta) \qquad \text{(by properties of the variance)} \tag{S.7.137}$$
$$= A^2\sigma^2 + B^2 \tag{S.7.138}$$

Since y is the (weighted) sum of two Gaussians, it is Gaussian itself, and hence its distribution is completely defined by its mean and variance, so that

$$y \sim \mathcal{N}(y|A\mu, A^2\sigma^2 + B^2). \tag{S.7.139}$$

Now, the product $\mathcal{N}(x|\mu, \sigma^2)\mathcal{N}(y|Ax, B^2)$ is proportional to the joint pdf of x and y, so that the integral can be considered to correspond to the marginalisation of x, and hence its result is proportional to the density of y, which is $\mathcal{N}(y|A\mu, A^2\sigma^2 + B^2)$.

(c) Show that

$$\mathcal{N}(x|m_1, \sigma_1^2)\mathcal{N}(x|m_2, \sigma_2^2) \propto \mathcal{N}(x|m_3, \sigma_3^2) \tag{7.22}$$

where

$$\sigma_3^2 = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1} = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{7.23}$$
$$m_3 = \sigma_3^2\left(\frac{m_1}{\sigma_1^2} + \frac{m_2}{\sigma_2^2}\right) = m_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(m_2 - m_1) \tag{7.24}$$

Hint: Work in the negative log domain.

Solution. We show the result using a classical technique called "completing the square", see e.g. https://en.wikipedia.org/wiki/Completing_the_square.

We work in the (negative) log-domain and use that

$$-\log\left[\mathcal{N}(x|m,\sigma^2)\right] = \frac{(x-m)^2}{2\sigma^2} + \text{const} \tag{S.7.140}$$

$$= \frac{x^2}{2\sigma^2} - x\frac{m}{\sigma^2} + \frac{m^2}{2\sigma^2} + \text{const} \tag{S.7.141}$$

$$= \frac{x^2}{2\sigma^2} - x\frac{m}{\sigma^2} + \text{const} \tag{S.7.142}$$

where const indicates terms not depending on x. We thus obtain

$$-\log\left[\mathcal{N}(x|m_1,\sigma_1^2)\mathcal{N}(x|m_2,\sigma_2^2)\right] = -\log\left[\mathcal{N}(x|m_1,\sigma_1^2)\right] - \log\left[\mathcal{N}(x|m_2,\sigma_2^2)\right] \tag{S.7.143}$$

$$= \frac{(x-m_1)^2}{2\sigma_1^2} + \frac{(x-m_2)^2}{2\sigma_2^2} + \text{const} \tag{S.7.144}$$

$$= \frac{x^2}{2\sigma_1^2} - x\frac{m_1}{\sigma_1^2} + \frac{x^2}{2\sigma_2^2} - x\frac{m_2}{\sigma_2^2} + \text{const} \tag{S.7.145}$$

$$= \frac{x^2}{2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right) - x\left(\frac{m_1}{\sigma_1^2} + \frac{m_2}{\sigma_2^2}\right) + \text{const} \tag{S.7.146}$$

$$= \frac{x^2}{2\sigma_3^2} - \frac{x}{\sigma_3^2}\sigma_3^2\left(\frac{m_1}{\sigma_1^2} + \frac{m_2}{\sigma_2^2}\right) + \text{const}, \tag{S.7.147}$$

where

$$\frac{1}{\sigma_3^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}. \tag{S.7.148}$$

Comparison with (S.7.142) shows that we can further write

$$\frac{x^2}{2\sigma_3^2} - \frac{x}{\sigma_3^2}\sigma_3^2\left(\frac{m_1}{\sigma_1^2} + \frac{m_2}{\sigma_2^2}\right) = \frac{(x-m_3)^2}{2\sigma_3^2} + \text{const} \tag{S.7.149}$$

where

$$m_3 = \sigma_3^2\left(\frac{m_1}{\sigma_1^2} + \frac{m_2}{\sigma_2^2}\right) \tag{S.7.150}$$

so that

$$-\log\left[\mathcal{N}(x|m_1,\sigma_1^2)\mathcal{N}(x|m_2,\sigma_2^2)\right] = \frac{(x-m_3)^2}{2\sigma_3^2} + \text{const} \tag{S.7.151}$$

and hence

$$\mathcal{N}(x|m_1,\sigma_1^2)\mathcal{N}(x|m_2,\sigma_2^2) \propto \mathcal{N}(x|m_3,\sigma_3^2). \tag{S.7.152}$$

Note that the identity

$$m_3 = \sigma_3^2\left(\frac{m_1}{\sigma_1^2} + \frac{m_2}{\sigma_2^2}\right) = m_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}(m_2 - m_1) \tag{S.7.153}$$

is obtained as follows

$$\sigma_3^2\left(\frac{m_1}{\sigma_1^2}+\frac{m_2}{\sigma_2^2}\right)=\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}\left(\frac{m_1}{\sigma_1^2}+\frac{m_2}{\sigma_2^2}\right) \tag{S.7.154}$$

$$=m_1\frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}+m_2\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2} \tag{S.7.155}$$

$$=m_1\left(1-\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)+m_2\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2} \tag{S.7.156}$$

$$=m_1+\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}(m_2-m_1) \tag{S.7.157}$$

(d) We can use the "alpha-recursion" to recursively compute $p(h_t|v_{1:t}) \propto \alpha(h_t)$ as follows.

$$\alpha(h_1)=p(h_1)\cdot p(v_1|h_1) \qquad \alpha(h_s)=p(v_s|h_s)\sum_{h_{s-1}} p(h_s|h_{s-1})\alpha(h_{s-1}). \tag{7.25}$$

For continuous random variables, the sum above becomes an integral so that

$$\alpha(h_s)=p(v_s|h_s)\int p(h_s|h_{s-1})\alpha(h_{s-1})\mathrm{d}h_{s-1}. \tag{7.26}$$

For reference, let us denote the integral by $I(h_s)$,

$$I(h_s)=\int p(h_s|h_{s-1})\alpha(h_{s-1})\mathrm{d}h_{s-1}. \tag{7.27}$$

Note that $I(h_s)$ is proportional to the predictive distribution $p(h_s|v_{1:s-1})$.

For a Gaussian prior distribution for h_1 and Gaussian emission probability $p(v_1|h_1)$, $\alpha(h_1) = p(h_1)\cdot p(v_1|h_1) \propto p(h_1|v_1)$ is proportional to a Gaussian. We denote its mean by μ_1 and its variance by σ_1^2 so that

$$\alpha(h_1)\propto\mathcal{N}(h_1|\mu_1,\sigma_1^2). \tag{7.28}$$

Assuming $\alpha(h_{s-1})\propto\mathcal{N}(h_{s-1}|\mu_{s-1},\sigma_{s-1}^2)$ (which holds for $s=2$), use Equation (7.21) to show that

$$I(h_s)\propto\mathcal{N}(h_s|A_s\mu_{s-1},P_s) \tag{7.29}$$

where

$$P_s=A_s^2\sigma_{s-1}^2+B_s^2. \tag{7.30}$$

Solution. We can set $\alpha(h_{s-1})\propto\mathcal{N}(h_{s-1}|\mu_{s-1},\sigma_{s-1}^2)$. Since $p(h_s|h_{s-1})$ is Gaussian, see Equation (7.17), Equation (7.27) becomes

$$I(h_s)\propto\int\mathcal{N}(h_s|A_sh_{s-1},B_s^2)\mathcal{N}(h_{s-1}|\mu_{s-1},\sigma_{s-1}^2)\mathrm{d}h_{s-1}. \tag{S.7.158}$$

Equation (7.21) with $x\equiv h_{s-1}$ and $y\equiv h_s$ yields the desired result,

$$I(h_s)\propto\mathcal{N}(h_s|A_s\mu_{s-1},A_s^2\sigma_{s-1}^2+B_s^2). \tag{S.7.159}$$

We can understand the equation as follows: To compute the predictive mean of h_s given $v_{1:s-1}$, we forward propagate the mean of $h_{s-1}|v_{1:s-1}$ using the update equation (7.19). This gives the mean term $A_s\mu_{s-1}$. Since $h_{s-1}|v_{1:s-1}$ has variance σ^2_{s-1}, the variance of $h_s|v_{1:s-1}$ is given by $A_s^2\sigma^2_{s-1}$ plus an additional term, B_s^2, due to the noise in the forward propagation. This gives the variance term $A_s^2\sigma^2_{s-1} + B_s^2$.

(e) Use Equation (7.22) to show that

$$\alpha(h_s) \propto \mathcal{N}\left(h_s|\mu_s, \sigma_s^2\right) \tag{7.31}$$

where

$$\mu_s = A_s\mu_{s-1} + \frac{P_sC_s}{C_s^2P_s + D_s^2}\left(v_s - C_sA_s\mu_{s-1}\right) \tag{7.32}$$

$$\sigma_s^2 = \frac{P_sD_s^2}{P_sC_s^2 + D_s^2} \tag{7.33}$$

Solution. Having computed $I(h_s)$, the final step in the alpha-recursion is

$$\alpha(h_s) = p(v_s|h_s)I(h_s) \tag{S.7.160}$$

With Equation (7.18) we obtain

$$\alpha(h_s) \propto \mathcal{N}(v_s|C_sh_s, D_s^2)\mathcal{N}(h_s|A_s\mu_{s-1}, P_s). \tag{S.7.161}$$

We further note that

$$\mathcal{N}(v_s|C_sh_s, D_s^2) \propto \mathcal{N}\left(h_s|C_s^{-1}v_s, \frac{D_s^2}{C_s^2}\right) \tag{S.7.162}$$

so that we can apply Equation (7.22) (with $m_1 = A\mu_{s-1}$, $\sigma_1^2 = P_s$)

$$\alpha(h_s) \propto \mathcal{N}\left(h_s|C_s^{-1}v_s, \frac{D_s^2}{C_s^2}\right)\mathcal{N}(h_s|A_s\mu_{s-1}, P_s) \tag{S.7.163}$$

$$\propto \mathcal{N}\left(h_s, \mu_s, \sigma_s^2\right) \tag{S.7.164}$$

with

$$\mu_s = A_s\mu_{s-1} + \frac{P_s}{P_s + \frac{D_s^2}{C_s^2}}\left(C_s^{-1}v_s - A_s\mu_{s-1}\right) \tag{S.7.165}$$

$$= A_s\mu_{s-1} + \frac{P_sC_s^2}{C_s^2P_s + D_s^2}\left(C_s^{-1}v_s - A_s\mu_{s-1}\right) \tag{S.7.166}$$

$$= A_s\mu_{s-1} + \frac{P_sC_s}{C_s^2P_s + D_s^2}\left(v_s - C_sA_s\mu_{s-1}\right) \tag{S.7.167}$$

$$\sigma_s^2 = \frac{P_s\frac{D_s^2}{C_s^2}}{P_s + \frac{D_s^2}{C_s^2}} \tag{S.7.168}$$

$$= \frac{P_sD_s^2}{P_sC_s^2 + D_s^2} \tag{S.7.169}$$

(S.7.170)

(f) Show that $\alpha(h_s)$ can be re-written as

$$\alpha(h_s) \propto \mathcal{N}\left(h_s|\mu_s, \sigma_s^2\right) \tag{7.34}$$

where

$$\mu_s = A_s\mu_{s-1} + K_s\left(v_s - C_sA_s\mu_{s-1}\right) \tag{7.35}$$
$$\sigma_s^2 = (1 - K_sC_s)P_s \tag{7.36}$$
$$K_s = \frac{P_sC_s}{C_s^2P_s + D_s^2} \tag{7.37}$$

These are the Kalman filter equations and K_s is called the Kalman filter gain.

Solution. We start from

$$\mu_s = A_s\mu_{s-1} + \frac{P_sC_s}{C_s^2P_s + D_s^2}\left(v_s - C_sA_s\mu_{s-1}\right), \tag{S.7.171}$$

and see that

$$\frac{P_sC_s}{C_s^2P_s + D_s^2} = K_s \tag{S.7.172}$$

so that

$$\mu_s = A_s\mu_{s-1} + K_s\left(v_s - C_sA_s\mu_{s-1}\right). \tag{S.7.173}$$

For the variance σ_s^2, we have

$$\sigma_s^2 = \frac{P_sD_s^2}{P_sC_s^2 + D_s^2} \tag{S.7.174}$$
$$= \frac{D_s^2}{P_sC_s^2 + D_s^2}P_s \tag{S.7.175}$$
$$= \left(1 - \frac{P_sC_s^2}{P_sC_s^2 + D_s^2}\right)P_s \tag{S.7.176}$$
$$= (1 - K_sC_s)P_s, \tag{S.7.177}$$

which is the desired result.

The filtering result generalises to vector valued latents and visibles where the transition and emission distributions in (7.17) and (7.18) become

$$p(\mathbf{h}_s|\mathbf{h}_{s-1}) = \mathcal{N}(\mathbf{h}_s|\mathbf{A}\mathbf{h}_{s-1}, \boldsymbol{\Sigma}^h), \tag{S.7.178}$$
$$p(\mathbf{v}_s|\mathbf{h}_s) = \mathcal{N}(\mathbf{v}_s|\mathbf{C}_s\mathbf{h}_s, \boldsymbol{\Sigma}^v), \tag{S.7.179}$$

where $\mathcal{N}()$ denotes multivariate Gaussian pdfs, e.g.

$$\mathcal{N}(\mathbf{v}_s|\mathbf{C}_s\mathbf{h}_s, \boldsymbol{\Sigma}^v) = \frac{1}{|\det(2\pi\boldsymbol{\Sigma}^v)|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{v}_s - \mathbf{C}_s\mathbf{h}_s)^\top(\boldsymbol{\Sigma}^v)^{-1}(\mathbf{v}_s - \mathbf{C}_s\mathbf{h}_s)\right). \tag{S.7.180}$$

We then have

$$p(\mathbf{h}_t|\mathbf{v}_{1:t}) = \mathcal{N}(\mathbf{h}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{S.7.181}$$

where the posterior mean and variance are recursively computed as

$$\boldsymbol{\mu}_s = \mathbf{A}_s\boldsymbol{\mu}_{s-1} + \mathbf{K}_s(\mathbf{v}_s - \mathbf{C}_s\mathbf{A}_s\boldsymbol{\mu}_{s-1}) \tag{S.7.182}$$

$$\boldsymbol{\Sigma}_s = (\mathbf{I} - \mathbf{K}_s\mathbf{C}_s)\mathbf{P}_s \tag{S.7.183}$$

$$\mathbf{P}_s = \mathbf{A}_s\boldsymbol{\Sigma}_{s-1}\mathbf{A}_s^\top + \boldsymbol{\Sigma}^h \tag{S.7.184}$$

$$\mathbf{K}_s = \mathbf{P}_s\mathbf{C}_s^\top \left(\mathbf{C}_s\mathbf{P}_s\mathbf{C}_s^\top + \boldsymbol{\Sigma}^v\right)^{-1} \tag{S.7.185}$$

and initialised with $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$ equal to the mean and variance of $p(\mathbf{h}_1|\mathbf{v}_1)$. The matrix $\mathbf{K}_s$ is then called the Kalman gain matrix.

The Kalman filter is widely applicable, see e.g. https://en.wikipedia.org/wiki/Kalman_filter, and has played a role in historic events such as the moon landing, see e.g. (Grewal and Andrews, 2010).

An example of the application of the Kalman filter to tracking is shown in Figure 7.1.

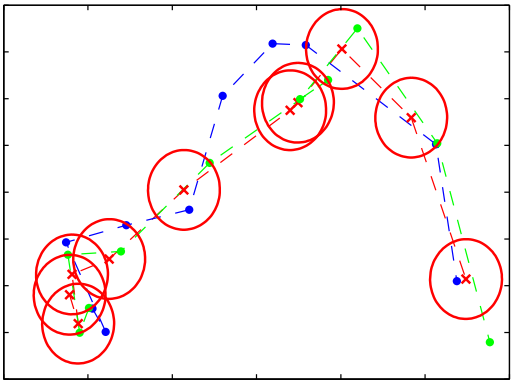


Figure 7.1: Kalman filtering for tracking of a moving object. The blue points indicate the true positions of the object in a two-dimensional space at successive time steps, the green points denote noisy measurements of the positions, and the red crosses indicate the means of the inferred posterior distributions of the positions obtained by running the Kalman filtering equations. The covariances of the inferred positions are indicated by the red ellipses, which correspond to contours having one standard deviation. (Bishop, 2006, Figure 13.22)

(g) Explain Equation (7.35) in non-technical terms. What happens if the variance D_s^2 of the observation noise goes to zero?

Solution. We have already seen that $A_s\mu_{s-1}$ is the predictive mean of h_s given $v_{1:s-1}$. The term $C_sA_s\mu_{s-1}$ is thus the predictive mean of v_s given the observations so far, $v_{1:s-1}$. The difference $v_s - C_sA_s\mu_{s-1}$ is thus the prediction error of the observable. Since $\alpha(h_s)$ is proportional to $p(h_s|v_{1:s})$ and μ_s its mean, we thus see that the posterior mean of $h_s|v_{1:s}$ equals the posterior mean of $h_s|v_{1:s-1}$, $A_s\mu_{s-1}$, updated by the prediction error of the observable weighted by the Kalman gain.

For $D_s^2 \to 0$, $K_s \to C_s^{-1}$ and

$$\mu_s = A_s\mu_{s-1} + K_s\left(v_s - C_sA_s\mu_{s-1}\right) \tag{S.7.186}$$
$$= A_s\mu_{s-1} + C_s^{-1}\left(v_s - C_sA_s\mu_{s-1}\right) \tag{S.7.187}$$
$$= A_s\mu_{s-1} + C_s^{-1}v_s - A_s\mu_{s-1} \tag{S.7.188}$$
$$= C_s^{-1}v_s, \tag{S.7.189}$$

so that the posterior mean of $p(h_s|v_{1:s})$ is obtained by inverting the observation equation. Moreover, the variance σ_s^2 of $h_s|v_{1:s}$ goes to zero so that the value of h_s is known precisely and equals $C_s^{-1}v_s$.

Chapter 8

Model-Based Learning

Exercises

8.1 Maximum likelihood estimation for a Gaussian

The Gaussian pdf parametrised by mean μ and standard deviation σ is given by

$$p(x;\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \qquad \boldsymbol{\theta} = (\mu, \sigma).$$

(a) Given iid data $\mathcal{D} = \{x_1, \ldots, x_n\}$, what is the likelihood function $L(\boldsymbol{\theta})$ for the Gaussian model?

Solution. For iid data, the likelihood function is

$$L(\boldsymbol{\theta}) = \prod_i^n p(x_i;\boldsymbol{\theta}) \tag{S.8.1}$$

$$= \prod_i^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right] \tag{S.8.2}$$

$$= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2\right]. \tag{S.8.3}$$

(b) What is the log-likelihood function $\ell(\boldsymbol{\theta})$?

Solution. Taking the log of the likelihood function gives

$$\ell(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2 \tag{S.8.4}$$

(c) Show that the maximum likelihood estimates for the mean μ and standard deviation σ are the sample mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i \tag{8.1}$$

and the square root of the sample variance

$$S^2 = \frac{1}{n}\sum_{i=1}^n (x_i-\bar{x})^2. \tag{8.2}$$

Solution. Since the logarithm is strictly monotonically increasing, the maximiser of the log-likelihood equals the maximiser of the likelihood. It is easier to take derivatives for the log-likelihood function than for the likelihood function so that the maximum likelihood estimate is typically determined using the log-likelihood.

Given the algebraic expression of $\ell(\boldsymbol{\theta})$, it is simpler to work with the variance $v = \sigma^2$ rather than the standard deviation. Since $\sigma > 0$ the function $v = g(\sigma) = \sigma^2$ is invertible, and the invariance of the MLE to re-parametrisation guarantees that

$$\hat{\sigma} = \sqrt{\hat{v}}.$$

We now thus maximise the function $J(\mu, v)$,

$$J(\mu, v) = -\frac{n}{2}\log(2\pi v) - \frac{1}{2v}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{S.8.5}$$

with respect to μ and v.

Taking partial derivatives gives

$$\frac{\partial J}{\partial \mu} = \frac{1}{v}\sum_{i=1}^{n}(x_i - \mu) \tag{S.8.6}$$

$$= \frac{1}{v}\sum_{i=1}^{n} x_i - \frac{n}{v}\mu \tag{S.8.7}$$

$$\frac{\partial J}{\partial v} = -\frac{n}{2}\frac{1}{v} + \frac{1}{2v^2}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{S.8.8}$$

A necessary condition for optimality is that the partial derivatives are zero. We thus obtain the conditions

$$\frac{1}{v}\sum_{i=1}^{n}(x_i - \mu) = 0 \tag{S.8.9}$$

$$-\frac{n}{2}\frac{1}{v} + \frac{1}{2v^2}\sum_{i=1}^{n}(x_i - \mu)^2 = 0 \tag{S.8.10}$$

From the first condition it follows that

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{S.8.11}$$

The second condition thus becomes

$$-\frac{n}{2}\frac{1}{v} + \frac{1}{2v^2}\sum_{i=1}^{n}(x_i - \hat{\mu})^2 = 0 \qquad \text{(multiply with } v^2 \text{ and rearrange)} \tag{S.8.12}$$

$$\frac{1}{2}\sum_{i=1}^{n}(x_i - \hat{\mu})^2 = \frac{n}{2}v, \tag{S.8.13}$$

and hence

$$\hat{v} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2, \tag{S.8.14}$$

We now check that this solution corresponds to a maximum by computing the Hessian matrix

$$\mathbf{H}(\mu, v) = \begin{pmatrix} \frac{\partial^2 J}{\partial \mu^2} & \frac{\partial^2 J}{\partial \mu \partial v} \\ \frac{\partial^2 J}{\partial \mu \partial v} & \frac{\partial^2 J}{\partial v^2} \end{pmatrix} \tag{S.8.15}$$

If the Hessian negative definite at $(\hat{\mu}, \hat{v})$, the point is a (local) maximum. Since we only have one critical point, $(\hat{\mu}, \hat{v})$, the local maximum is also a global maximum. Taking second derivatives gives

$$\mathbf{H}(\mu, v) = \begin{pmatrix} -\frac{n}{v} & -\frac{1}{v^2}\sum_{i=1}^{n}(x_i - \mu) \\ -\frac{1}{v^2}\sum_{i=1}^{n}(x_i - \mu) & \frac{n}{2}\frac{1}{v^2} - \frac{1}{v^3}\sum_{i=1}^{n}(x_i - \mu)^2 \end{pmatrix}. \tag{S.8.16}$$

Substituting the values for $(\hat{\mu}, \hat{v})$ gives

$$\mathbf{H}(\hat{\mu}, \hat{v}) = \begin{pmatrix} -\frac{n}{\hat{v}} & 0 \\ 0 & -\frac{n}{2}\frac{1}{\hat{v}^2} \end{pmatrix}, \tag{S.8.17}$$

which is negative definite. Note that the the (negative) curvature increases with n, which means that $J(\mu, v)$, and hence the log-likelihood becomes more and more peaked as the number of data points n increases.

8.2 Posterior of the mean of a Gaussian with known variance

Given iid data $\mathcal{D} = \{x_1, \ldots, x_n\}$, compute $p(\mu|\mathcal{D}, \sigma^2)$ for the Bayesian model

$$p(x|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \qquad p(\mu; \mu_0, \sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right] \tag{8.3}$$

where σ^2 is a fixed known quantity.
Hint: You may use that

$$\mathcal{N}(x; m_1, \sigma_1^2)\mathcal{N}(x; m_2, \sigma_2^2) \propto \mathcal{N}(x; m_3, \sigma_3^2) \tag{8.4}$$

where

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{8.5}$$

$$\sigma_3^2 = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1} = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{8.6}$$

$$m_3 = \sigma_3^2\left(\frac{m_1}{\sigma_1^2} + \frac{m_2}{\sigma_2^2}\right) = m_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(m_2 - m_1) \tag{8.7}$$

Solution. We re-use the expression for the likelihood $L(\mu)$ from Exercise 8.1.

$$L(\mu) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right], \tag{S.8.18}$$

which we can write as

$$L(\mu) \propto \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right] \tag{S.8.19}$$

$$\propto \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i^2 - 2\mu x_i + \mu^2)\right] \tag{S.8.20}$$

$$\propto \exp\left[-\frac{1}{2\sigma^2}\left(-2\mu\sum_{i=1}^{n}x_i + n\mu^2\right)\right] \tag{S.8.21}$$

$$\propto \exp\left[-\frac{1}{2\sigma^2}\left(-2n\mu\bar{x} + n\mu^2\right)\right] \tag{S.8.22}$$

$$\propto \exp\left[-\frac{n}{2\sigma^2}(\mu - \bar{x})^2\right] \tag{S.8.23}$$

$$\propto \mathcal{N}(\mu; \bar{x}, \sigma^2/n). \tag{S.8.24}$$

The posterior is

$$p(\mu|\mathcal{D}) \propto L(\theta)p(\mu;\mu_0,\sigma_0^2) \tag{S.8.25}$$
$$\propto \mathcal{N}(\mu;\bar{x},\sigma^2/n)\mathcal{N}(\mu;\mu_0,\sigma_0^2) \tag{S.8.26}$$

so that with (8.4), we have

$$p(\mu|\mathcal{D}) \propto \mathcal{N}(\mu;\mu_n,\sigma_n^2) \tag{S.8.27}$$
$$\sigma_n^2 = \left(\frac{1}{\sigma^2/n}+\frac{1}{\sigma_0^2}\right)^{-1} \tag{S.8.28}$$
$$= \frac{\sigma_0^2\sigma^2/n}{\sigma_0^2+\sigma^2/n} \tag{S.8.29}$$
$$\mu_n = \sigma_n^2\left(\frac{\bar{x}}{\sigma^2/n}+\frac{\mu_0}{\sigma_0^2}\right) \tag{S.8.30}$$
$$= \frac{1}{\sigma_0^2+\sigma^2/n}\left(\sigma_0^2\bar{x}+(\sigma^2/n)\mu_0\right) \tag{S.8.31}$$
$$= \frac{\sigma_0^2}{\sigma_0^2+\sigma^2/n}\bar{x}+\frac{\sigma^2/n}{\sigma_0^2+\sigma^2/n}\mu_0. \tag{S.8.32}$$

As n increases, σ^2/n goes to zero so that $\sigma_n^2 \to 0$ and $\mu_n \to \bar{x}$. This means that with an increasing amount of data, the posterior of the mean tends to be concentrated around the maximum likelihood estimate $\bar{x}$.

From (8.7), we also have that

$$\mu_n = \mu_0 + \frac{\sigma_0^2}{\sigma^2/n+\sigma_0^2}(\bar{x}-\mu_0), \tag{S.8.33}$$

which shows more clearly that the value of μ_n lies on a line with end-points μ_0 (for $n=0$) and $\bar{x}$ (for $n\to\infty$). As the amount of data increases, μ_n moves form the mean under the prior, μ_0, to the average of the observed sample, that is the MLE $\bar{x}$.

8.3 Maximum likelihood estimation of probability tables in fully observed directed graphical models of binary variables

We assume that we are given a parametrised directed graphical model for variables $x_1,\ldots,x_d$,

$$p(\mathbf{x};\boldsymbol{\theta}) = \prod_{i=1}^{d} p(x_i|\mathrm{pa}_i;\boldsymbol{\theta}_i) \qquad x_i \in \{0,1\} \tag{8.8}$$

where the conditionals are represented by parametrised probability tables, For example, if $\mathrm{pa}_3 = \{x_1,x_2\}$, $p(x_3|\mathrm{pa}_3;\boldsymbol{\theta}_3)$ is represented as

$p(x_3=1\|x_1,x_2;\theta_3^1,\ldots,\theta_3^4))$	x_1	x_2
θ_3^1	0	0
θ_3^2	1	0
θ_3^3	0	1
θ_3^4	1	1

with $\boldsymbol{\theta}_3 = (\theta_3^1, \theta_3^2, \theta_3^3, \theta_3^4)$, and where the superscripts j of θ_3^j enumerate the different states that the parents can be in.

(a) Assuming that x_i has m_i parents, verify that the table parametrisation of $p(x_i|\text{pa}_i;\boldsymbol{\theta}_i)$ is equivalent to writing $p(x_i|\text{pa}_i;\boldsymbol{\theta}_i)$ as

$$p(x_i|\text{pa}_i;\boldsymbol{\theta}_i) = \prod_{s=1}^{S_i} (\theta_i^s)^{\mathbb{1}(x_i=1,\text{pa}_i=s)} (1-\theta_i^s)^{\mathbb{1}(x_i=0,\text{pa}_i=s)} \tag{8.9}$$

where $S_i = 2^{m_i}$ is the total number of states/configurations that the parents can be in, and $\mathbb{1}(x_i = 1, \text{pa}_i = s)$ is one if $x_i = 1$ and $\text{pa}_i = s$, and zero otherwise.

Solution. The number of configurations that m binary parents can be in is given by S_i. The questions thus boils down to showing that $p(x_i = 1|\text{pa}_i = k;\boldsymbol{\theta}_i) = \theta_i^k$ for any state $k \in \{1, \ldots, S_i\}$ of the parents of x_i. Since $\mathbb{1}(x_i = 1, \text{pa}_i = s) = 0$ unless $s = k$, we have indeed that

$$p(x_i = 1|\text{pa}_i = k;\boldsymbol{\theta}_i) = \left[\prod_{s\neq k} (\theta_i^s)^0 (1-\theta_i^s)^0\right] (\theta_i^k)^{\mathbb{1}(x_i=1,\text{pa}_i=k)} (1-\theta_i^k)^{\mathbb{1}(x_i=0,\text{pa}_i=k)} \tag{S.8.34}$$

$$= 1 \cdot (\theta_i^k)^{\mathbb{1}(x_i=1,\text{pa}_i=k)} (1-\theta_i^k)^0 \tag{S.8.35}$$

$$= \theta_i^k. \tag{S.8.36}$$

(b) For iid data $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ show that the likelihood can be represented as

$$p(\mathcal{D};\boldsymbol{\theta}) = \prod_{i=1}^{d} \prod_{s=1}^{S_i} (\theta_i^s)^{n^s_{x_i=1}} (1-\theta_i^s)^{n^s_{x_i=0}} \tag{8.10}$$

where $n^s_{x_i=1}$ is the number of times the pattern $(x_i = 1, \text{pa}_i = s)$ occurs in the data $\mathcal{D}$, and equivalently for $n^s_{x_i=0}$.

Solution. Since the data are iid, we have

$$p(\mathcal{D};\boldsymbol{\theta}) = \prod_{j=1}^{n} p(\mathbf{x}^{(j)};\boldsymbol{\theta}) \tag{S.8.37}$$

$$\tag{S.8.38}$$

where each term $p(\mathbf{x}^{(j)};\boldsymbol{\theta})$ factorises as in (8.8),

$$p(\mathbf{x}^{(j)};\boldsymbol{\theta}) = \prod_{i=1}^{d} p(x_i^{(j)}|\text{pa}_i^{(j)};\boldsymbol{\theta}_i) \tag{S.8.39}$$

with $x_i^{(j)}$ denoting the i-th element of $\mathbf{x}^{(j)}$ and $\text{pa}_i^{(j)}$ the corresponding parents. The conditionals $p(x_i^{(j)}|\text{pa}_i^{(j)};\boldsymbol{\theta}_i)$ factorise further according to (8.9),

$$p(x_i^{(j)}|\text{pa}_i^{(j)};\boldsymbol{\theta}_i) = \prod_{s=1}^{S_i} (\theta_i^s)^{\mathbb{1}(x_i^{(j)}=1,\text{pa}_i^{(j)}=s)} (1-\theta_i^s)^{\mathbb{1}(x_i^{(j)}=0,\text{pa}_i^{(j)}=s)}, \tag{S.8.40}$$

so that

$$p(\mathcal{D};\boldsymbol{\theta}) = \prod_{j=1}^{n}\prod_{i=1}^{d} p(x_i^{(j)}|\mathrm{pa}_i^{(j)};\boldsymbol{\theta}_i) \tag{S.8.41}$$

$$= \prod_{j=1}^{n}\prod_{i=1}^{d}\prod_{s=1}^{S_i}(\theta_i^s)^{\mathbb{1}(x_i^{(j)}=1,\mathrm{pa}_i^{(j)}=s)}(1-\theta_i^s)^{\mathbb{1}(x_i^{(j)}=0,\mathrm{pa}_i^{(j)}=s)} \tag{S.8.42}$$

Swapping the order of the products so that the product over the data points comes first, we obtain

$$p(\mathcal{D};\boldsymbol{\theta}) = \prod_{i=1}^{d}\prod_{s=1}^{S_i}\prod_{j=1}^{n}(\theta_i^s)^{\mathbb{1}(x_i^{(j)}=1,\mathrm{pa}_i^{(j)}=s)}(1-\theta_i^s)^{\mathbb{1}(x_i^{(j)}=0,\mathrm{pa}_i^{(j)}=s)} \tag{S.8.43}$$

We next split the product over j into two products, one for all j where $x_i^{(j)} = 1$, and one for all j where $x_i^{(j)} = 0$

$$p(\mathcal{D};\boldsymbol{\theta}) = \prod_{i=1}^{d}\prod_{s=1}^{S_i}\prod_{\substack{j:\\ x_i^{(j)}=1}}\prod_{\substack{j:\\ x_i^{(j)}=0}}(\theta_i^s)^{\mathbb{1}(x_i^{(j)}=1,\mathrm{pa}_i^{(j)}=s)}(1-\theta_i^s)^{\mathbb{1}(x_i^{(j)}=0,\mathrm{pa}_i^{(j)}=s)} \tag{S.8.44}$$

$$= \prod_{i=1}^{d}\prod_{s=1}^{S_i}\prod_{\substack{j:\\ x_i^{(j)}=1}}(\theta_i^s)^{\mathbb{1}(x_i^{(j)}=1,\mathrm{pa}_i^{(j)}=s)}\prod_{\substack{j:\\ x_i^{(j)}=0}}(1-\theta_i^s)^{\mathbb{1}(x_i^{(j)}=0,\mathrm{pa}_i^{(j)}=s)} \tag{S.8.45}$$

$$= \prod_{i=1}^{d}\prod_{s=1}^{S_i}(\theta_i^s)^{\sum_{j=1}^{n}\mathbb{1}(x_i^{(j)}=1,\mathrm{pa}_i^{(j)}=s)}(1-\theta_i^s)^{\sum_{j=1}^{n}\mathbb{1}(x_i^{(j)}=0,\mathrm{pa}_i^{(j)}=s)} \tag{S.8.46}$$

$$= \prod_{i=1}^{d}\prod_{s=1}^{S_i}(\theta_i^s)^{n_{x_i=1}^s}(1-\theta_i^s)^{n_{x_i=0}^s} \tag{S.8.47}$$

where

$$n_{x_i=1}^s = \sum_{j=1}^{n}\mathbb{1}(x_i^{(j)}=1,\mathrm{pa}_i^{(j)}=s) \qquad n_{x_i=0}^s = \sum_{j=1}^{n}\mathbb{1}(x_i^{(j)}=0,\mathrm{pa}_i^{(j)}=s) \tag{S.8.48}$$

is the number of times $x_i = 1$ and $x_i = 0$, respectively, with its parents being in state s.

(c) Show that the log-likelihood decomposes into sums of terms that can be independently optimised, and that each term corresponds to the log-likelihood for a Bernoulli model.

Solution. The log-likelihood $\ell(\boldsymbol{\theta})$ equals

$$\ell(\boldsymbol{\theta}) = \log p(\mathcal{D}; \boldsymbol{\theta}) \tag{S.8.49}$$

$$= \log \prod_{i=1}^{d} \prod_{s=1}^{S_i} (\theta_i^s)^{n^s_{x_i=1}} (1-\theta_i^s)^{n^s_{x_i=0}} \tag{S.8.50}$$

$$= \sum_{i=1}^{d} \sum_{s=1}^{S_i} \log \left[(\theta_i^s)^{n^s_{x_i=1}} (1-\theta_i^s)^{n^s_{x_i=0}} \right] \tag{S.8.51}$$

$$= \sum_{i=1}^{d} \sum_{s=1}^{S_i} n^s_{x_i=1} \log(\theta_i^s) + n^s_{x_i=0} \log(1-\theta_i^s) \tag{S.8.52}$$

Since the parameters θ_i^s are not coupled in any way, maximising $\ell(\boldsymbol{\theta})$ can be achieved by maximising each term $\ell_{is}(\theta_i^s)$ individually,

$$\ell_{is}(\theta_i^s) = n^s_{x_i=1} \log(\theta_i^s) + n^s_{x_i=0} \log(1-\theta_i^s). \tag{S.8.53}$$

Moreover, $\ell_{is}(\theta_i^s)$ corresponds to the log-likelihood for a Bernoulli model with success probability θ_i^s and data with $n^s_{x_i=1}$ number of ones and $n^s_{x_i=0}$ number of zeros.

(d) Determine the maximum likelihood estimate $\hat{\theta}$ for the Bernoulli model

$$p(x; \theta) = \theta^x (1-\theta)^{1-x}, \qquad \theta \in [0,\ 1], \qquad x \in \{0, 1\} \tag{8.11}$$

and iid data $x_1, \ldots, x_n$.

Solution. The log-likelihood function is

$$\ell(\theta) = \sum_{i=1}^{n} \log p(x_i; \theta) \tag{S.8.54}$$

$$= \sum_{i=1}^{n} x_i \log(\theta) + 1 - x_i \log(1-\theta). \tag{S.8.55}$$

Since $\log(\theta)$ and $\log(1-\theta)$ do not depend on i, we can pull them outside the sum and the log-likelihood function can be written as

$$\ell(\theta) = n_{x=1} \log(\theta) + n_{x=0} \log(1-\theta) \tag{S.8.56}$$

where $n_{x=1} = \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} \mathbb{1}(x_i = 1)$ and $n_{x=0} = n - n_{x=1}$ are the number of ones and zeros in the data. Since $\theta \in [0,\ 1]$, we have to solve the constrained optimisation problem

$$\hat{\theta} = \operatorname*{argmax}_{\theta \in [0,\ 1]} \ell(\theta) \tag{S.8.57}$$

There are multiple ways to solve the problem. One option is to determine the *unconstrained* optimiser and then check whether it satisfies the constraint. The first derivative equals

$$\ell'(\theta) = \frac{n_{x=1}}{\theta} - \frac{n_{x=0}}{1-\theta} \tag{S.8.58}$$

and the second derivative is

$$\ell''(\theta) = -\frac{n_{x=1}}{\theta^2} - \frac{n_{x=0}}{(1-\theta)^2} \tag{S.8.59}$$

The second derivative is always negative for $\theta \in (0,1)$, which means that $\ell(\theta)$ is strictly concave on $(0,1)$ and that an optimiser that is not on the boundary corresponds to a maximum. Setting the first derivative to zero gives the condition

$$\frac{n_{x=1}}{\theta} = \frac{n_{x=0}}{1-\theta} \tag{S.8.60}$$

Solving for θ gives

$$(1-\theta)n_{x=1} = n_{x=0}\theta \tag{S.8.61}$$

$$\tag{S.8.62}$$

so that

$$n_{x=1} = \theta(n_{x=0} + n_{x=1}) \tag{S.8.63}$$

$$= \theta n \tag{S.8.64}$$

Hence, we find

$$\hat{\theta} = \frac{n_{x=1}}{n}. \tag{S.8.65}$$

For $n_{x=1} < n$, we have $\hat{\theta} \in (0,1)$ so that the constraint is actually not active.

In the derivation, we had to exclude boundary cases where θ is 0 or 1. We note that e.g. $\hat{\theta} = 1$ is obtained when $n_{x=1} = n$, i.e. when we only observe 1's in the data set. In that case, $n_{x=0} = 0$ and the log-likelihood function equals $n\log(\theta)$, which is strictly increasing and hence attains the maximum at $\hat{\theta} = 1$. A similar argument shows that if $n_{x=1} = 0$, the maximum is at $\hat{\theta} = 0$. Hence, the maximum likelihood estimate

$$\hat{\theta} = \frac{n_{x=1}}{n} \tag{S.8.66}$$

is valid for all $n_{x=1} \in \{0, \ldots, n\}$.

An alternative approach to deal with the constraint is to reparametrise the objective function and work with the log-odds η,

$$\eta = g(\theta) = \log\left[\frac{\theta}{1-\theta}\right]. \tag{S.8.67}$$

The log-odds take values in $\mathbb{R}$ so that η is unconstrained. The transformation from θ to η is invertible and

$$\theta = g^{-1}(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)} = \frac{1}{1+\exp(-\eta)}. \tag{S.8.68}$$

The optimisation problem then becomes

$$\hat{\eta} = \underset{\eta}{\operatorname{argmax}}\, n_{x=1}\eta - n\log(1+\exp(\eta))$$

Computing the second derivative shows that the objective is concave for all η and the maximiser $\hat{\eta}$ can be determined by setting the first derivative to zero. The maximum likelihood estimate of θ is then given by

$$\hat{\theta} = \frac{\exp(\hat{\eta})}{1+\exp(\hat{\eta})} \tag{S.8.69}$$

The reason for this is as follows: Let $J(\eta) = \ell(g^{-1}(\eta))$ be the log-likelihood seen as a function of η. Since g and g^{-1} are invertible, we have that

$$\max_{\theta\in[0,1]} \ell(\theta) = \max_{\eta} J(\eta) \tag{S.8.70}$$

$$\operatorname*{argmax}_{\theta\in[0,1]} \ell(\theta) = g^{-1}\left(\operatorname*{argmax}_{\eta} J(\eta)\right). \tag{S.8.71}$$

(e) Returning to the fully observed directed graphical model, conclude that the maximum likelihood estimates are given by

$$\hat{\theta}_i^s = \frac{n^s_{x_i=1}}{n^s_{x_i=1} + n^s_{x_i=0}} = \frac{\sum_{j=1}^n \mathbb{1}(x_i^{(j)} = 1, \mathrm{pa}_i^{(j)} = s)}{\sum_{j=1}^n \mathbb{1}(\mathrm{pa}_i^{(j)} = s)} \tag{8.12}$$

Solution. Given the result from question (c), we can optimise each term $\ell_{is}(\theta_i^s)$ separately. Each term formally corresponds to a log-likelihood for a Bernoulli model, so that we can use the results from question (d) to obtain

$$\hat{\theta}_i^s = \frac{n^s_{x_i=1}}{n^s_{x_i=1} + n^s_{x_i=0}}. \tag{S.8.72}$$

Since $n^s_{x_i=1} = \sum_{j=1}^n \mathbb{1}(x_i^{(j)} = 1, \mathrm{pa}_i^{(j)} = s)$ and

$$n^s_{x_i=1} + n^s_{x_i=0} = \sum_{j=1}^n \mathbb{1}(x_i^{(j)} = 1, \mathrm{pa}_i^{(j)} = s) + \sum_{j=1}^n \mathbb{1}(x_i^{(j)} = 0, \mathrm{pa}_i^{(j)} = s) \tag{S.8.73}$$

$$= \sum_{j=1}^n \mathbb{1}(\mathrm{pa}_i^{(j)} = s), \tag{S.8.74}$$

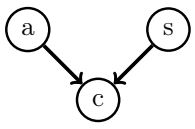
we further have

$$\hat{\theta}_i^s = \frac{\sum_{j=1}^n \mathbb{1}(x_i^{(j)} = 1, \mathrm{pa}_i^{(j)} = s)}{\sum_{j=1}^n \mathbb{1}(\mathrm{pa}_i^{(j)} = s)}. \tag{S.8.75}$$

Hence, to determine $\hat{\theta}_i^s$, we first count the number of times the parents of x_i are in state s, which gives the denominator, and then among them, count the number of times $x_i = 1$, which gives the numerator.

8.4 Cancer-asbestos-smoking example: MLE

Consider the model specified by the DAG



The distribution of a and s are Bernoulli distributions with parameter (success probability) θ_a and θ_s, respectively, i.e.

$$p(a;\theta_a) = \theta_a^a(1-\theta_a)^{1-a} \qquad p(s;\theta_s) = \theta_s^s(1-\theta_s)^{1-s}, \tag{8.13}$$

and the distribution of c given the parents is parametrised as specified in the following table

$p(c=1\|a,s;\theta_c^1,\ldots,\theta_c^4))$	a	s
θ_c^1	0	0
θ_c^2	1	0
θ_c^3	0	1
θ_c^4	1	1

The free parameters of the model are $(\theta_a, \theta_s, \theta_c^1, \ldots, \theta_c^4)$.

Assume we observe the following iid data (each row is a data point).

a	s	c
0	1	1
0	0	0
1	0	1
0	0	0
0	1	0

(a) Determine the maximum-likelihood estimates of θ_a and θ_s

Solution. The maximum likelihood estimate (MLE) $\hat{\theta}_a$ is given by the fraction of times that a is 1 in the data set. Hence $\hat{\theta}_a = 1/5$. Similarly, the MLE $\hat{\theta}_s$ is 2/5.

(b) Determine the maximum-likelihood estimates of $\theta_c^1, \ldots, \theta_c^4$.

Solution. With (S.8.75), we have

$\hat{p}(c=1\|a,s)$	a	s
$\hat{\theta}_c^1 = 0$	0	0
$\hat{\theta}_c^2 = 1/1$	1	0
$\hat{\theta}_c^3 = 1/2$	0	1
$\hat{\theta}_c^4$ not defined	1	1

This because, for example, we have two observations where $(a, s) = (0, 0)$, and among them, $c = 1$ never occurs, so that the MLE for $p(c = 1|a, s)$ is zero.

This example illustrates some issues with maximum likelihood estimates: We may get extreme probabilities, zero or one, or if the parent configuration does not occur in the observed data, the estimate is undefined.

8.5 Bayesian inference for the Bernoulli model

Consider the Bayesian model

$$p(x|\theta) = \theta^x(1-\theta)^{1-x} \qquad p(\theta;\boldsymbol{\alpha}_0) = \mathcal{B}(\theta;\alpha_0,\beta_0)$$

where $x \in \{0,1\},\ \theta \in [0,1], \boldsymbol{\alpha}_0 = (\alpha_0,\beta_0)$, and

$$\mathcal{B}(\theta;\alpha,\beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1} \qquad \theta \in [0,1] \tag{8.14}$$

(a) Given iid data $\mathcal{D} = \{x_1,\ldots,x_n\}$ show that the posterior of θ given $\mathcal{D}$ is

$$p(\theta|\mathcal{D}) = \mathcal{B}(\theta;\alpha_n,\beta_n)$$
$$\alpha_n = \alpha_0 + n_{x=1} \qquad \beta_n = \beta_0 + n_{x=0}$$

where $n_{x=1}$ denotes the number of ones and $n_{x=0}$ the number of zeros in the data.

Solution. This follows from

$$p(\theta|\mathcal{D}) \propto L(\theta)p(\theta;\boldsymbol{\alpha}_0) \tag{S.8.76}$$

and from the expression for the likelihood function of the Bernoulli model, which is

$$L(\theta) = \prod_{i=1}^{n} p(x_i|\theta) \tag{S.8.77}$$
$$= \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i} \tag{S.8.78}$$
$$= \theta^{\sum_{i=1}^n x_i}(1-\theta)^{\sum_{i=1}^n (1-x_i)} \tag{S.8.79}$$
$$= \theta^{n_{x=1}}(1-\theta)^{n_{x=0}}, \tag{S.8.80}$$

where $n_{x=1} = \sum_{i=1}^n x_i$ denotes the number of 1's in the data, and $n_{x=0} = \sum_{i=1}^n (1-x_i) = n - n_{x=1}$ the number of 0's.

Inserting the expressions for the likelihood and prior into (S.8.76) gives

$$p(\theta|\mathcal{D}) \propto \theta^{n_{x=1}}(1-\theta)^{n_{x=0}}\theta^{\alpha_0-1}(1-\theta)^{\beta_0-1} \tag{S.8.81}$$
$$\propto \theta^{\alpha_0+n_{x=1}-1}(1-\theta)^{\beta_0+n_{x=0}-1} \tag{S.8.82}$$
$$\propto \mathcal{B}(\theta,\alpha_0+n_{x=1},\beta_0+n_{x=0}), \tag{S.8.83}$$

which is the desired result. Since α_0 and β_0 are updated by the counts of ones and zeros in the data, these hyperparameters are also referred to as "pseudo-counts". Alternatively, one can think that they are the counts that are observed in another iid data set which has been previously analysed and used to determine the prior.

(b) Compute the mean of a Beta random variable f,

$$p(f;\alpha,\beta) = \mathcal{B}(f;\alpha,\beta) \qquad f \in [0,1], \tag{8.15}$$

using that

$$\int_0^1 f^{\alpha-1}(1-f)^{\beta-1}\mathrm{d}f = B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{8.16}$$

where $B(\alpha, \beta)$ denotes the Beta function and where the Gamma function $\Gamma(t)$ is defined as

$$\Gamma(t) = \int_o^\infty f^{t-1} \exp(-f) \mathrm{d}f \tag{8.17}$$

and satisfies $\Gamma(t+1) = t\Gamma(t)$.
Hint: It will be useful to represent the partition function in terms of the Beta function.

Solution. We first write the partition function of $p(f; \alpha, \beta)$ in terms of the Beta function

$$Z(\alpha, \beta) = \int_0^1 f^{\alpha-1}(1-f)^{\beta-1} \tag{S.8.84}$$

$$= B(\alpha, \beta). \tag{S.8.85}$$

We then have that the mean $\mathbb{E}[f]$ is given by

$$\mathbb{E}[f] = \int_0^1 f p(f; \alpha, \beta) \mathrm{d}f \tag{S.8.86}$$

$$= \frac{1}{B(\alpha, \beta)} \int_0^1 f f^{\alpha-1}(1-f)^{\beta-1} \mathrm{d}f \tag{S.8.87}$$

$$= \frac{1}{B(\alpha, \beta)} \int_0^1 f^{\alpha+1-1}(1-f)^{\beta-1} \mathrm{d}f \tag{S.8.88}$$

$$= \frac{B(\alpha+1, \beta)}{B(\alpha, \beta)} \tag{S.8.89}$$

$$= \frac{\Gamma(\alpha+1)\Gamma(\beta)}{\Gamma(\alpha+1+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{S.8.90}$$

$$= \frac{\alpha\Gamma(\alpha)\Gamma(\beta)}{(\alpha+\beta)\Gamma(\alpha+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{S.8.91}$$

$$= \frac{\alpha}{\alpha+\beta} \tag{S.8.92}$$

where we have used the definition of the Beta function in terms of the Gamma function and the property $\Gamma(t+1) = t\Gamma(t)$.

(c) Show that the predictive posterior probability $p(x = 1|\mathcal{D})$ for a new independently observed data point x equals the posterior mean of $p(\theta|\mathcal{D})$, which in turn is given by

$$\mathbb{E}(\theta|\mathcal{D}) = \frac{\alpha_0 + n_{x=1}}{\alpha_0 + \beta_0 + n}. \tag{8.18}$$

Solution. We obtain

$$p(x = 1|\mathcal{D}) = \int_0^1 p(x = 1, \theta|\mathcal{D}) \mathrm{d}\theta \qquad \text{(sum rule)} \tag{S.8.93}$$

$$= \int_0^1 p(x = 1|\theta, \mathcal{D}) p(\theta|\mathcal{D}) \mathrm{d}\theta \qquad \text{(product rule)} \tag{S.8.94}$$

$$= \int_0^1 p(x = 1|\theta) p(\theta|\mathcal{D}) \mathrm{d}\theta \qquad (x \perp\!\!\!\perp \mathcal{D}|\theta) \tag{S.8.95}$$

$$= \int_0^1 \theta p(\theta|\mathcal{D}) \mathrm{d}\theta \tag{S.8.96}$$

$$= \mathbb{E}[\theta|\mathcal{D}] \tag{S.8.97}$$

From the previous question we know the mean of a Beta random variable. Since $\theta \sim \mathcal{B}(\theta; \alpha_n, \beta_n)$, we obtain

$$p(x = 1|\mathcal{D}) = \mathbb{E}[\theta|\mathcal{D}] \tag{S.8.98}$$
$$= \frac{\alpha_n}{\alpha_n + \beta_n} \tag{S.8.99}$$
$$= \frac{\alpha_0 + n_{x=1}}{\alpha_0 + n_{x=1} + \beta_0 + n_{x=0}} \tag{S.8.100}$$
$$= \frac{\alpha_0 + n_{x=1}}{\alpha_0 + \beta_0 + n} \tag{S.8.101}$$

where the last equation follows from the fact that $n = n_{x=0} + n_{x=1}$. Note that for $n \to \infty$, the posterior mean tends to the MLE $n_{x=1}/n$.

8.6 Bayesian inference of probability tables in fully observed directed graphical models of binary variables

This is the Bayesian analogue of Exercise 8.3 and the notation follows that exercise. We consider the Bayesian model

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^{d} p(x_i|\text{pa}_i, \boldsymbol{\theta}_i) \qquad x_i \in \{0, 1\} \tag{8.19}$$
$$p(\boldsymbol{\theta}; \boldsymbol{\alpha}_0, \boldsymbol{\beta}_0) = \prod_{i=1}^{d} \prod_{s=1}^{S_i} \mathcal{B}(\theta_i^s; \alpha_{i,0}^s, \beta_{i,0}^s) \tag{8.20}$$

where $p(x_i|\text{pa}_i, \boldsymbol{\theta}_i)$ is defined via (8.9), $\boldsymbol{\alpha}_0$ is a vector of hyperparameters containing all $\alpha_{i,0}^s$, $\boldsymbol{\beta}_0$ the vector containing all $\beta_{i,0}^s$, and as before $\mathcal{B}$ denotes the Beta distribution. Under the prior, all parameters are independent.

(a) For iid data $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ show that

$$p(\boldsymbol{\theta}|\mathcal{D}) = \prod_{i=1}^{d} \prod_{s=1}^{S_i} \mathcal{B}(\theta_i^s, \alpha_{i,n}^s, \beta_{i,n}^s) \tag{8.21}$$

where

$$\alpha_{i,n}^s = \alpha_{i,0}^s + n_{x_i=1}^s \qquad \beta_{i,n}^s = \beta_{i,0}^s + n_{x_i=0}^s \tag{8.22}$$

and that the parameters are also independent under the posterior.

Solution. We start with

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}; \boldsymbol{\alpha}_0, \boldsymbol{\beta}_0). \tag{S.8.102}$$

Inserting the expression for $p(\mathcal{D}|\boldsymbol{\theta})$ given in (8.10) and the assumed form of the prior gives

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \prod_{i=1}^{d}\prod_{s=1}^{S_i}(\theta_i^s)^{n^s_{x_i=1}}(1-\theta_i^s)^{n^s_{x_i=0}}\prod_{i=1}^{d}\prod_{s=1}^{S_i}\mathcal{B}(\theta_i^s;\alpha^s_{i,0},\beta^s_{i,0}) \tag{S.8.103}$$

$$\propto \prod_{i=1}^{d}\prod_{s=1}^{S_i}(\theta_i^s)^{n^s_{x_i=1}}(1-\theta_i^s)^{n^s_{x_i=0}}\mathcal{B}(\theta_i^s;\alpha^s_{i,0},\beta^s_{i,0}) \tag{S.8.104}$$

$$\propto \prod_{i=1}^{d}\prod_{s=1}^{S_i}(\theta_i^s)^{n^s_{x_i=1}}(1-\theta_i^s)^{n^s_{x_i=0}}(\theta_i^s)^{\alpha^s_{i,0}-1}(1-\theta_i^s)^{\beta^s_{i,0}-1} \tag{S.8.105}$$

$$\propto \prod_{i=1}^{d}\prod_{s=1}^{S_i}(\theta_i^s)^{\alpha^s_{i,0}+n^s_{x_i=1}-1}(1-\theta_i^s)^{\beta^s_{i,0}+n^s_{x_i=0}-1} \tag{S.8.106}$$

$$\propto \prod_{i=1}^{d}\prod_{s=1}^{S_i}\mathcal{B}(\theta_i^s;\alpha^s_{i,0}+n^s_{x_i=1},\beta^s_{i,0}+n^s_{x_i=0}) \tag{S.8.107}$$

It can be immediately verified that $\mathcal{B}(\theta_i^s;\alpha^s_{i,0}+n^s_{x_i=1},\beta^s_{i,0}+n^s_{x_i=0})$ is proportional to the marginal $p(\theta_i^s|\mathcal{D})$ so that the parameters are independent under the posterior too.

(b) For a variable x_i with parents pa_i, compute the posterior predictive probability $p(x_i = 1|\mathrm{pa}_i, \mathcal{D})$

Solution. The solution is analogue to the solution for question (c), using the sum rule, independencies, and properties of beta random variables:

$$p(x_i = 1|\mathrm{pa}_i = s, \mathcal{D}) = \int p(x_i = 1, \theta_i^s|\mathrm{pa}_i = s, \mathcal{D})\mathrm{d}\theta_i^s \tag{S.8.108}$$

$$= \int p(x_i = 1|\theta_i^s, \mathrm{pa}_i = s, \mathcal{D})p(\theta_i^s|\mathrm{pa}_i = s, \mathcal{D}) \tag{S.8.109}$$

$$= \int p(x_i = 1|\theta_i^s, \mathrm{pa}_i = s)p(\theta_i^s|\mathcal{D}) \tag{S.8.110}$$

$$= \int \theta_i^s p(\theta_i^s|\mathcal{D}) \tag{S.8.111}$$

$$= \mathbb{E}[\theta_i^s|\mathcal{D})] \tag{S.8.112}$$

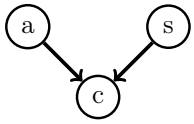
$$\overset{(S.8.92)}{=} \frac{\alpha^s_{i,n}}{\alpha^s_{i,n}+\beta^s_{i,n}} \tag{S.8.113}$$

$$= \frac{\alpha^s_{i,0}+n^s_{x_i=1}}{\alpha^s_{i,0}+\beta^s_{i,0}+n^s} \tag{S.8.114}$$

where $n^s = n^s_{x_i=0} + n^s_{x_i=1}$ denotes the number of times the parent configuration s occurs in the observed data $\mathcal{D}$.

8.7 Cancer-asbestos-smoking example: Bayesian inference

Consider the model specified by the DAG



The distribution of a and s are Bernoulli distributions with parameter (success probability) θ_a and θ_s, respectively, i.e.

$$p(a|\theta_a) = \theta_a^a(1-\theta_a)^{1-a} \qquad p(s|\theta_s) = \theta_s^s(1-\theta_s)^{1-s}, \tag{8.23}$$

and the distribution of c given the parents is parametrised as specified in the following table

$p(c=1\|a,s,\theta_c^1,\ldots,\theta_c^4))$	a	s
θ_c^1	0	0
θ_c^2	1	0
θ_c^3	0	1
θ_c^4	1	1

We assume that the prior over the parameters of the model, $(\theta_a, \theta_s, \theta_c^1, \ldots, \theta_c^4)$, factorises and is given by beta distributions with hyperparameters $\alpha_0 = 1$ and $\beta_0 = 1$ (same for all parameters).

Assume we observe the following iid data (each row is a data point).

a	s	c
0	1	1
0	0	0
1	0	1
0	0	0
0	1	0

(a) Determine the posterior predictive probabilities $p(a=1|\mathcal{D})$ and $p(s=1|\mathcal{D})$.

Solution. With Exercise 8.5 question (c), we have

$$p(a=1|\mathcal{D}) = \mathbb{E}(\theta^a|\mathcal{D}) = \frac{1+1}{1+1+5} = \frac{2}{7} \tag{S.8.115}$$

$$p(s=1|\mathcal{D}) = \mathbb{E}(\theta^s|\mathcal{D}) = \frac{1+2}{1+1+5} = \frac{3}{7} \tag{S.8.116}$$

(b) Determine the posterior predictive probabilities $p(c=1|\text{pa},\mathcal{D})$ for all possible parent configurations.

Solution. The parents of c are (a, s). With Exercise 8.6 question (b), we have

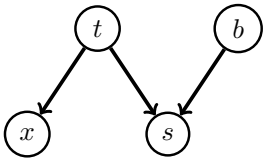
$p(c = 1\|a, s, \mathcal{D})$	a	s
$(1 + 0)/(1 + 1 + 2) = 1/4$	0	0
$(1 + 1)/(1 + 1 + 1) = 2/3$	1	0
$(1 + 1)/(1 + 1 + 2) = 1/2$	0	1
$(1 + 0)/(1 + 1) = 1/2$	1	1

Compared to the MLE solution in Exercise (b) question (b), we see that the estimates are less extreme. This is because they are a combination of the prior knowledge and the observed data. Moreover, when we do not have any data, the posterior equals the prior, unlike for the mle where the estimate is not defined.

8.8 Learning parameters of a directed graphical model

We consider the directed graphical model shown below on the left for the four binary variables t, b, s, x, each being either zero or one. Assume that we have observed the data shown in the table on the right.

Model:



$t = 1$ has tuberculosis
$b = 1$ has bronchitis
$s = 1$ has shortness of breath
$x = 1$ has positive x-ray

Observed data:

x	s	t	b
0	1	0	1
0	0	0	0
0	1	0	1
0	1	0	1
0	0	0	0
0	0	0	0
0	1	0	1
0	1	0	1
0	0	0	1
1	1	1	0

We assume the (conditional) pmf of $s|t, b$ is specified by the following parametrised probability table:

$p(s = 1\|t, b; \theta_s^1, \ldots, \theta_s^4))$	t	b
θ_s^1	0	0
θ_s^2	1	0
θ_s^3	0	1
θ_s^4	1	1

(a) What are the maximum likelihood estimates for $p(s = 1|b = 0, t = 0)$ and $p(s = 1|b = 0, t = 1)$, i.e. the parameters θ_s^1 and θ_s^3?

Solution. The maximum likelihood estimates (MLEs) are equal to the fraction of occurrences of the relevant events.

$$\hat{\theta}_s^1 = \frac{\sum_{i=1}^n \mathbb{1}(s_i = 1, b_i = 0, t_i = 0)}{\sum_{i=1}^n \mathbb{1}(b_i = 0, t_i = 0)} = \frac{0}{3} = 0 \tag{S.8.117}$$

$$\hat{\theta}_s^3 = \frac{\sum_{i=1}^n \mathbb{1}(s_i = 1, b_i = 0, t_i = 1)}{\sum_{i=1}^n \mathbb{1}(b_i = 0, t_i = 1)} = \frac{1}{1} = 1 \tag{S.8.118}$$

(b) Assume each parameter in the table for $p(s|t, b)$ has a uniform prior on $(0, 1)$. Compute the posterior mean of the parameters of $p(s = 1|b = 0, t = 0)$ and $p(s = 1|b = 0, t = 1)$ and explain the difference to the maximum likelihood estimates.

Solution. A uniform prior corresponds to a Beta distribution with hyperparameters $\alpha_0 = \beta_0 = 1$. With Exercise 8.6 question (b), we have

$$\mathbb{E}(\theta_s^1|\mathcal{D}) = \frac{\alpha_0 + 0}{\alpha_0 + \beta_0 + 3} = \frac{1}{5} \tag{S.8.119}$$

$$\mathbb{E}(\theta_s^3|\mathcal{D}) = \frac{\alpha_0 + 1}{\alpha_0 + \beta_0 + 1} = \frac{2}{3} \tag{S.8.120}$$

Compared to the MLE, the posterior mean is less extreme. It can be considered a "smoothed out" or regularised estimate, where $\alpha_0 > 0$ and $\beta_0 > 0$ provides regularisation (see https://en.wikipedia.org/wiki/Additive_smoothing). We can see a pull of the parameters towards the prior predictive mean, which equals $1/2$.

8.9 Factor analysis

A friend proposes to improve the factor analysis model by working with correlated latent variables. The proposed model is

$$p(\mathbf{h}; \mathbf{C}) = \mathcal{N}(\mathbf{h}; \mathbf{0}, \mathbf{C}) \qquad p(\mathbf{v}|\mathbf{h}; \mathbf{F}, \boldsymbol{\Psi}, \mathbf{c}) = \mathcal{N}(\mathbf{v}; \mathbf{F}\mathbf{h} + \mathbf{c}, \boldsymbol{\Psi}) \tag{8.24}$$

where $\mathbf{C}$ is some $H \times H$ covariance matrix, $\mathbf{F}$ is the $D \times H$ matrix with the factor loadings, $\boldsymbol{\Psi} = \text{diag}(\Psi_1, \ldots, \Psi_D)$, $\mathbf{c} \in \mathbb{R}^D$ and the dimension of the latents H is less than the dimension of the visibles D. $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the pdf of a Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The standard factor analysis model is obtained when $\mathbf{C}$ is the identity matrix.

(a) What is marginal distribution of the visibles $p(\mathbf{v}; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ stands for the parameters $\mathbf{C}, \mathbf{F}, \mathbf{c}, \boldsymbol{\Psi}$?

Solution. The model specifications are equivalent to the following data generating process:

$$\mathbf{h} \sim \mathcal{N}(\mathbf{h}; \mathbf{0}, \mathbf{C}) \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \boldsymbol{\Psi}) \qquad \mathbf{v} = \mathbf{F}\mathbf{h} + \mathbf{c} + \boldsymbol{\epsilon} \tag{S.8.121}$$

Recall the basic result on the distribution of linear transformations of Gaussians: if $\mathbf{x}$ has density $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x, \mathbf{C}_x)$, $\mathbf{z}$ density $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_z, \mathbf{C}_z)$, and $\mathbf{x} \perp\!\!\!\perp \mathbf{z}$ then $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$ has density

$$\mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_x + \boldsymbol{\mu}_z, \mathbf{A}\mathbf{C}_x\mathbf{A}^\top + \mathbf{C}_z).$$

It thus follows that $\mathbf{v}$ is Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$,

$$\boldsymbol{\mu} = \mathbf{F}\underbrace{\mathbb{E}[\mathbf{h}]}_{\mathbf{0}} + \mathbf{c} + \underbrace{\mathbb{E}[\boldsymbol{\epsilon}]}_{\mathbf{0}} \tag{S.8.122}$$

$$= \mathbf{c} \tag{S.8.123}$$

$$\boldsymbol{\Sigma} = \mathbf{F}\mathbb{V}[\mathbf{h}]\mathbf{F}^\top + \mathbb{V}[\boldsymbol{\epsilon}] \tag{S.8.124}$$

$$= \mathbf{F}\mathbf{C}\mathbf{F}^\top + \boldsymbol{\Psi}. \tag{S.8.125}$$

(b) Assume that the singular value decomposition of $\mathbf{C}$ is given by

$$\mathbf{C} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\top \tag{8.25}$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_D)$ is a diagonal matrix containing the eigenvalues, and $\mathbf{E}$ is a orthonormal matrix containing the corresponding eigenvectors. The matrix square root of $\mathbf{C}$ is the matrix $\mathbf{M}$ such that

$$\mathbf{M}\mathbf{M} = \mathbf{C}, \tag{8.26}$$

and we denote it by $\mathbf{C}^{1/2}$. Show that the matrix square root of $\mathbf{C}$ equals

$$\mathbf{C}^{1/2} = \mathbf{E}\,\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D})\mathbf{E}^\top. \tag{8.27}$$

Solution. We verify that $\mathbf{C}^{1/2}\mathbf{C}^{1/2} = \mathbf{C}$:

$$\mathbf{C}^{1/2}\mathbf{C}^{1/2} = \mathbf{E}\,\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D})\mathbf{E}^\top\mathbf{E}\,\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D})\mathbf{E}^\top \tag{S.8.126}$$

$$= \mathbf{E}\,\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D})\;\mathbf{I}\;\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D})\mathbf{E}^\top \tag{S.8.127}$$

$$= \mathbf{E}\,\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D})\,\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_D})\mathbf{E}^\top \tag{S.8.128}$$

$$= \mathbf{E}\,\text{diag}(\lambda_1, \ldots, \lambda_D)\mathbf{E}^\top \tag{S.8.129}$$

$$= \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\top \tag{S.8.130}$$

$$= \mathbf{C} \tag{S.8.131}$$

(c) Show that the proposed factor analysis model is equivalent to the original factor analysis model

$$p(\mathbf{h}; \mathbf{I}) = \mathcal{N}(\mathbf{h}; \mathbf{0}, \mathbf{I}) \qquad p(\mathbf{v}|\mathbf{h}; \tilde{\mathbf{F}}, \boldsymbol{\Psi}, \mathbf{c}) = \mathcal{N}(\mathbf{v}; \tilde{\mathbf{F}}\mathbf{h} + \mathbf{c}, \boldsymbol{\Psi}) \tag{8.28}$$

with $\tilde{\mathbf{F}} = \mathbf{F}\mathbf{C}^{1/2}$, so that the extra parameters given by the covariance matrix $\mathbf{C}$ are actually redundant and nothing is gained with the richer parametrisation.

Solution. We verify that the model has the same distribution for the visibles. As before $\mathbb{E}[\mathbf{v}] = \mathbf{c}$, and the covariance matrix is

$$\mathbb{V}[\mathbf{v}] = \tilde{\mathbf{F}}\mathbf{I}\tilde{\mathbf{F}}^\top + \boldsymbol{\Psi} \tag{S.8.132}$$

$$= \mathbf{F}\mathbf{C}^{1/2}\mathbf{C}^{1/2}\mathbf{F}^\top + \boldsymbol{\Psi} \tag{S.8.133}$$

$$= \mathbf{F}\mathbf{C}\mathbf{F}^\top + \boldsymbol{\Psi} \tag{S.8.134}$$

where we have used that $\mathbf{C}^{1/2}$ is a symmetric matrix. This means that the correlation between the $\mathbf{h}$ can be absorbed into the factor matrix $\mathbf{F}$ and the set of pdfs defined by the proposed model equals the set of pdfs of the original factor analysis model.

Another way to see the result is to consider the data generating process and noting that we can sample $\mathbf{h}$ from $\mathcal{N}(\mathbf{h}; \mathbf{0}, \mathbf{C})$ by first sampling $\mathbf{h}'$ from $\mathcal{N}(\mathbf{h}'; \mathbf{0}, \mathbf{I})$ and then transforming the sample by $\mathbf{C}^{1/2}$,

$$\mathbf{h} \sim \mathcal{N}(\mathbf{h}; \mathbf{0}, \mathbf{C}) \qquad \Longleftrightarrow \qquad \mathbf{h} = \mathbf{C}^{1/2}\mathbf{h}' \qquad \mathbf{h}' \sim \mathcal{N}(\mathbf{h}'; \mathbf{0}, \mathbf{I}). \tag{S.8.135}$$

This follows again from the basic properties of linear transformations of Gaussians, i.e.

$$\mathbb{V}(\mathbf{C}^{1/2}\mathbf{h}') = \mathbf{C}^{1/2}\mathbb{V}(\mathbf{h}')(\mathbf{C}^{1/2})^\top = \mathbf{C}^{1/2}\mathbf{I}\mathbf{C}^{1/2} = \mathbf{C}$$

and $\mathbb{E}(\mathbf{C}^{1/2}\mathbf{h}') = \mathbf{C}^{1/2}\mathbb{E}(\mathbf{h}') = \mathbf{0}$.

To generate samples from the proposed factor analysis model, we would thus proceed as follows:

$$\mathbf{h}' \sim \mathcal{N}(\mathbf{h}'; \mathbf{0}, \mathbf{I}) \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \boldsymbol{\Psi}) \qquad \mathbf{v} = \mathbf{F}(\mathbf{C}^{1/2}\mathbf{h}') + \mathbf{c} + \boldsymbol{\epsilon} \tag{S.8.136}$$

But the term

$$\mathbf{v} = \mathbf{F}(\mathbf{C}^{1/2}\mathbf{h}') + \mathbf{c} + \boldsymbol{\epsilon}$$

can be written as

$$\mathbf{v} = (\mathbf{F}\mathbf{C}^{1/2})\mathbf{h}' + \mathbf{c} + \boldsymbol{\epsilon} = \tilde{\mathbf{F}}\mathbf{h}' + \mathbf{c} + \boldsymbol{\epsilon}$$

and since $\mathbf{h}'$ follows $\mathcal{N}(\mathbf{h}'; \mathbf{0}, \mathbf{I})$, we are back at the original factor analysis model.

8.10 Independent component analysis

(a) Whitening corresponds to linearly transforming a random variable $\mathbf{x}$ (or the corresponding data) so that the resulting random variable $\mathbf{z}$ has an identity covariance matrix, i.e.

$$\mathbf{z} = \mathbf{V}\mathbf{x} \quad \text{with} \quad \mathbb{V}[\mathbf{x}] = \mathbf{C} \quad \text{and} \quad \mathbb{V}[\mathbf{z}] = \mathbf{I}.$$

The matrix $\mathbf{V}$ is called the whitening matrix. We do not make a distributional assumption on $\mathbf{x}$, in particular $\mathbf{x}$ may or may not be Gaussian.

Given the eigenvalue decomposition $\mathbf{C} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\top$, show that

$$\mathbf{V} = \operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\mathbf{E}^\top \tag{8.29}$$

is a whitening matrix.

Solution. From $\mathbb{V}[\mathbf{z}] = \mathbb{V}[\mathbf{V}\mathbf{x}] = \mathbf{V}\mathbb{V}[\mathbf{x}]\mathbf{V}^\top$, it follows that

$$\begin{aligned}
\mathbb{V}[\mathbf{z}] &= \mathbf{V}\mathbb{V}[\mathbf{x}]\mathbf{V}^\top && \text{(S.8.137)}\\
&= \mathbf{V}\mathbf{C}\mathbf{V}^\top && \text{(S.8.138)}\\
&= \mathbf{V}\mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\top\mathbf{V}^\top && \text{(S.8.139)}\\
&= \operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\mathbf{E}^\top\mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\top\mathbf{V}^\top && \text{(S.8.140)}\\
&= \operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\boldsymbol{\Lambda}\mathbf{E}^\top\mathbf{V}^\top && \text{(S.8.141)}
\end{aligned}$$

where we have used that $\mathbf{E}^\top\mathbf{E} = \mathbf{I}$. Since

$$\mathbf{V}^\top = \left[\operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\mathbf{E}^\top\right]^\top = \mathbf{E}\operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})$$

we further have

$$\begin{aligned}
\mathbb{V}[\mathbf{z}] &= \operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\mathbf{\Lambda}\mathbf{E}^\top\mathbf{E}\operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2}) && \text{(S.8.142)}\\
&= \operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\mathbf{\Lambda}\operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2}) && \text{(S.8.143)}\\
&= \operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\operatorname{diag}(\lambda_1, \ldots, \lambda_d)\operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2}) && \text{(S.8.144)}\\
&= \mathbf{I}, && \text{(S.8.145)}
\end{aligned}$$

so that $\mathbf{V}$ is indeed a valid whitening matrix. Note that whitening matrices are not unique. For example,

$$\tilde{\mathbf{V}} = \mathbf{E}\operatorname{diag}(\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2})\mathbf{E}^\top$$

is also a valid whitening matrix. More generally, if $\mathbf{V}$ is a whitening matrix, then $\mathbf{RV}$ is also a whitening matrix when $\mathbf{R}$ is an orthonormal matrix. This is because

$$\mathbb{V}[\mathbf{RVx}] = \mathbf{R}\mathbb{V}[\mathbf{Vx}]\mathbf{R}^\top = \mathbf{RIR}^\top = \mathbf{I}$$

where we have used that $\mathbf{V}$ is a whitening matrix so that $\mathbf{Vx}$ has identity covariance matrix.

(b) Consider the ICA model

$$\mathbf{v} = \mathbf{Ah}, \qquad \mathbf{h} \sim p_{\mathbf{h}}(\mathbf{h}), \qquad p_{\mathbf{h}}(\mathbf{h}) = \prod_{i=1}^{D} p_h(h_i), \tag{8.30}$$

where the matrix $\mathbf{A}$ is invertible and the h_i are independent random variables of mean zero and variance one. Let $\mathbf{V}$ be a whitening matrix for $\mathbf{v}$. Show that $\mathbf{z} = \mathbf{Vv}$ follows the ICA model

$$\mathbf{z} = \tilde{\mathbf{A}}\mathbf{h}, \qquad \mathbf{h} \sim p_{\mathbf{h}}(\mathbf{h}), \qquad p_{\mathbf{h}}(\mathbf{h}) = \prod_{i=1}^{D} p_h(h_i), \tag{8.31}$$

where $\tilde{\mathbf{A}}$ is an orthonormal matrix.

Solution. If $\mathbf{v}$ follows the ICA model, we have

$$\begin{aligned}
\mathbf{z} &= \mathbf{Vv} && \text{(S.8.146)}\\
&= \mathbf{VAh} && \text{(S.8.147)}\\
&= \tilde{\mathbf{A}}\mathbf{h} && \text{(S.8.148)}
\end{aligned}$$

with $\tilde{\mathbf{A}} = \mathbf{VA}$. By the whitening operation, the covariance matrix of $\mathbf{z}$ is identity, so that

$$\mathbf{I} = \mathbb{V}(\mathbf{z}) = \tilde{\mathbf{A}}\mathbb{V}(\mathbf{h})\tilde{\mathbf{A}}^\top. \tag{S.8.149}$$

By the ICA model, $\mathbb{V}(\mathbf{h}) = \mathbf{I}$, so that $\tilde{\mathbf{A}}$ must satisfy

$$\mathbf{I} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top, \tag{S.8.150}$$

which means that $\tilde{\mathbf{A}}$ is orthonormal.

In the original ICA model, the number of parameters is given by the number of elements of the matrix $\mathbf{A}$, which is D^2 if $\mathbf{v}$ is D-dimensional. An orthogonal matrix contains $D(D-1)/2$ degrees of freedom (see e.g. https://en.wikipedia.org/wiki/Orthogonal_matrix), so that we can think that whitening "solves half of the ICA problem". Since whitening is a relatively simple standard operation, many algorithms (e.g. "fastICA", Hyvärinen, 1999) first reduce the complexity of the estimation problem by whitening the data. Moreover, due to the properties of the orthogonal matrix, the log-likelihood for the ICA model also simplifies for whitened data: The log-likelihood for ICA model without whitening is

$$\ell(\mathbf{B}) = \sum_{i=1}^{n}\sum_{j=1}^{D} \log p_h(\mathbf{b}_j\mathbf{v}_i) + n\log|\det\mathbf{B}| \tag{S.8.151}$$

where $\mathbf{B} = \mathbf{A}^{-1}$. If we first whiten the data, the log-likelihood becomes

$$\ell(\tilde{\mathbf{B}}) = \sum_{i=1}^{n}\sum_{j=1}^{D} \log p_h(\tilde{\mathbf{b}}_j\mathbf{z}_i) + n\log|\det\tilde{\mathbf{B}}| \tag{S.8.152}$$

where $\tilde{\mathbf{B}} = \tilde{\mathbf{A}}^{-1} = \tilde{\mathbf{A}}^\top$ since $\mathbf{A}$ is an orthogonal matrix. This means $\tilde{\mathbf{B}}^{-1} = \tilde{\mathbf{A}} = \tilde{\mathbf{B}}^\top$ and $\tilde{\mathbf{B}}$ is an orthogonal matrix. Hence $\det\tilde{\mathbf{B}} = 1$, and the log det term is zero. Hence, the log-likelihood on whitened data simplifies to

$$\ell(\tilde{\mathbf{B}}) = \sum_{i=1}^{n}\sum_{j=1}^{D} \log p_h(\tilde{\mathbf{b}}_j\mathbf{z}_i). \tag{S.8.153}$$

While the log-likelihood takes a simpler form, the optimisation problem is now a constrained optimisation problem: $\tilde{\mathbf{B}}$ is constrained to be orthonormal. For further information, see e.g. (Hyvärinen et al., 2001, Chapter 9).

8.11 Score matching for the exponential family

The objective function $J(\boldsymbol{\theta})$ that is minimised in score matching is

$$J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\partial_j\psi_j(\mathbf{x}_i;\boldsymbol{\theta}) + \frac{1}{2}\psi_j(\mathbf{x}_i;\boldsymbol{\theta})^2\right], \tag{8.32}$$

where ψ_j is the partial derivative of the log model-pdf $\log p(\mathbf{x};\boldsymbol{\theta})$ with respect to the j-th coordinate (slope) and $\partial_j\psi_j$ its second partial derivative (curvature). The observed data are denoted by $\mathbf{x}_1,\ldots,\mathbf{x}_n$ and $\mathbf{x}\in\mathbb{R}^m$.

The goal of this exercise is to show that for statistical models of the form

$$\log p(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{K}\theta_k F_k(\mathbf{x}) - \log Z(\boldsymbol{\theta}), \qquad \mathbf{x}\in\mathbb{R}^m, \tag{8.33}$$

the score matching objective function becomes a quadratic form, which can be optimised efficiently (see e.g. Barber, 2012, Appendix A.5.3).

The set of models above are called the (continuous) exponential family, or also log-linear models because the models are linear in the parameters θ_k. Since the exponential family generally includes probability mass functions as well, the qualifier "continuous" may be used to highlight that we are here considering continuous random variables only. The functions $F_k(\mathbf{x})$ are assumed to be known (they are called the sufficient statistics).

(a) Denote by $\mathbf{K}(\mathbf{x})$ the matrix with elements $K_{kj}(\mathbf{x})$,

$$K_{kj}(\mathbf{x}) = \frac{\partial F_k(\mathbf{x})}{\partial x_j}, \qquad k = 1 \dots K, \quad j = 1 \dots m, \tag{8.34}$$

and by $\mathbf{H}(\mathbf{x})$ the matrix with elements $H_{kj}(\mathbf{x})$,

$$H_{kj}(\mathbf{x}) = \frac{\partial^2 F_k(\mathbf{x})}{\partial x_j^2}, \qquad k = 1 \dots K, \quad j = 1 \dots m. \tag{8.35}$$

Furthermore, let $\mathbf{h}_j(\mathbf{x}) = (H_{1j}(\mathbf{x}), \dots, H_{Kj}(\mathbf{x}))^\top$ be the j–th column vector of $\mathbf{H}(\mathbf{x})$. Show that for the continuous exponential family, the score matching objective in Equation (8.32) becomes

$$J(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{r} + \frac{1}{2}\boldsymbol{\theta}^\top \mathbf{M}\boldsymbol{\theta}, \tag{8.36}$$

where

$$\mathbf{r} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{h}_j(\mathbf{x}_i), \qquad \mathbf{M} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{K}(\mathbf{x}_i)\mathbf{K}(\mathbf{x}_i)^\top. \tag{8.37}$$

Solution. For

$$\log p(\mathbf{x}; \boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k F_k(\mathbf{x}) - \log Z(\boldsymbol{\theta}) \tag{S.8.154}$$

the first derivative with respect to x_j, the j-th element of $\mathbf{x}$, is

$$\psi_j(\mathbf{x}; \boldsymbol{\theta}) = \frac{\partial \log p(\mathbf{x}; \boldsymbol{\theta})}{\partial x_j} \tag{S.8.155}$$

$$= \sum_{k=1}^{K} \theta_k \frac{\partial F_k(\mathbf{x})}{\partial x_j} \tag{S.8.156}$$

$$= \sum_{k=1}^{K} \theta_k K_{kj}(\mathbf{x}). \tag{S.8.157}$$

The second derivative is

$$\partial_j \psi_j(\mathbf{x}; \boldsymbol{\theta}) = \frac{\partial^2 \log p(\mathbf{x}; \boldsymbol{\theta})}{\partial x_j^2} \tag{S.8.158}$$

$$= \sum_{k=1}^{K} \theta_k \frac{\partial^2 F_k(\mathbf{x})}{\partial x_j^2} \tag{S.8.159}$$

$$= \sum_{k=1}^{K} \theta_k H_{kj}(\mathbf{x}), \tag{S.8.160}$$

which we can write more compactly as

$$\partial_j \psi_j(\mathbf{x};\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{h}_j(\mathbf{x}). \tag{S.8.161}$$

The score matching objective in Equation (8.32) features the sum $\sum_j \psi_j(\mathbf{x};\boldsymbol{\theta})^2$. The term $\psi_j(\mathbf{x};\boldsymbol{\theta})^2$ equals

$$\psi_j(\mathbf{x};\boldsymbol{\theta})^2 = \left[\sum_{k=1}^{K} \theta_k K_{kj}(\mathbf{x})\right]^2 \tag{S.8.162}$$

$$= \sum_{k=1}^{K}\sum_{k'=1}^{K} K_{kj}(\mathbf{x}) K_{k'j}(\mathbf{x}) \theta_k \theta_{k'}, \tag{S.8.163}$$

so that

$$\sum_{j=1}^{m} \psi_j(\mathbf{x};\boldsymbol{\theta})^2 = \sum_{j=1}^{m}\sum_{k=1}^{K}\sum_{k'=1}^{K} K_{kj}(\mathbf{x}) K_{k'j}(\mathbf{x}) \theta_k \theta_{k'} \tag{S.8.164}$$

$$= \sum_{k=1}^{K}\sum_{k'=1}^{K} \theta_k \theta_{k'} \left[\sum_{j=1}^{m} K_{kj}(\mathbf{x}) K_{k'j}(\mathbf{x})\right], \tag{S.8.165}$$

which can be more compactly expressed using matrix notation. Noting that

$$\sum_{j=1}^{m} K_{kj}(\mathbf{x}_i) K_{k'j}(\mathbf{x}_i)$$

equals the (k, k') element of the matrix-matrix product $\mathbf{K}(\mathbf{x}_i)\mathbf{K}(\mathbf{x}_i)^\top$,

$$\sum_{j=1}^{m} K_{kj}(\mathbf{x}_i) K_{k'j}(\mathbf{x}_i) = \left[\mathbf{K}(\mathbf{x}_i)\mathbf{K}(\mathbf{x}_i)^\top\right]_{k,k'}, \tag{S.8.166}$$

we can write

$$\sum_{j=1}^{m} \psi_j(\mathbf{x};\boldsymbol{\theta})^2 = \sum_{k=1}^{K}\sum_{k'=1}^{K} \theta_k \theta_{k'} \left[\mathbf{K}(\mathbf{x}_i)\mathbf{K}(\mathbf{x}_i)^\top\right]_{k,k'} \tag{S.8.167}$$

$$= \boldsymbol{\theta}^\top \mathbf{K}(\mathbf{x}_i)\mathbf{K}(\mathbf{x}_i)^\top \boldsymbol{\theta} \tag{S.8.168}$$

where we have used that for some matrix $\mathbf{A}$

$$\boldsymbol{\theta}^\top \mathbf{A} \boldsymbol{\theta} = \sum_{k,k'} \theta_k \theta_{k'} [\mathbf{A}]_{k,k'} \tag{S.8.169}$$

where $[\mathbf{A}]_{k,k'}$ is the (k, k') element of the matrix $\mathbf{A}$.

Inserting the expressions into Equation (8.32) gives

$$J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\partial_j\psi_j(\mathbf{x}_i;\boldsymbol{\theta}) + \frac{1}{2}\psi_j(\mathbf{x}_i;\boldsymbol{\theta})^2\right] \tag{S.8.170}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\partial_j\psi_j(\mathbf{x}_i;\boldsymbol{\theta}) + \frac{1}{2}\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\psi_j(\mathbf{x}_i;\boldsymbol{\theta})^2 \tag{S.8.171}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{\theta}^\top\mathbf{h}_j(\mathbf{x}_i) + \frac{1}{2}\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\theta}^\top\mathbf{K}(\mathbf{x}_i)\mathbf{K}(\mathbf{x}_i)^\top\boldsymbol{\theta} \tag{S.8.172}$$

$$= \boldsymbol{\theta}^\top\left[\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{h}_j(\mathbf{x}_i)\right] + \frac{1}{2}\boldsymbol{\theta}^\top\left[\frac{1}{n}\sum_{i=1}^{n}\mathbf{K}(\mathbf{x}_i)\mathbf{K}(\mathbf{x}_i)^\top\right]\boldsymbol{\theta} \tag{S.8.173}$$

$$= \boldsymbol{\theta}^\top\mathbf{r} + \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{M}\boldsymbol{\theta}, \tag{S.8.174}$$

which is the desired result.

(b) The pdf of a zero mean Gaussian parametrised by the variance σ^2 is

$$p(x;\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2}{2\sigma^2}\right), \qquad x \in \mathbb{R}. \tag{8.38}$$

The (multivariate) Gaussian is a member of the exponential family. By comparison with Equation (8.33), we can re-parametrise the statistical model $\{p(x;\sigma^2)\}_{\sigma^2}$ and work with

$$p(x;\theta) = \frac{1}{Z(\theta)}\exp\left(\theta x^2\right), \qquad \theta < 0, \qquad x \in \mathbb{R}, \tag{8.39}$$

instead. The two parametrisations are related by $\theta = -1/(2\sigma^2)$. Using the previous result on the (continuous) exponential family, determine the score matching estimate $\hat{\theta}$, and show that the corresponding $\hat{\sigma}^2$ is the same as the maximum likelihood estimate. This result is noteworthy because unlike in maximum likelihood estimation, score matching does not need the partition function $Z(\theta)$ for the estimation.

Solution. By comparison with Equation (8.33), the sufficient statistics $F(x)$ is x^2. We first determine the score matching objective function. For that, we need to determine the quantities $\mathbf{r}$ and $\mathbf{M}$ in Equation (8.37). Here, both $\mathbf{r}$ and $\mathbf{M}$ are scalars, and so are the matrices $\mathbf{K}$ and $\mathbf{H}$ that define $\mathbf{r}$ and $\mathbf{M}$. By their definitions, we obtain

$$K(x) = \frac{\partial F(x)}{\partial x} = 2x \tag{S.8.175}$$

$$H(x) = \frac{\partial^2 F(x)}{\partial x^2} = 2 \tag{S.8.176}$$

$$r = 2 \tag{S.8.177}$$

$$M = \frac{1}{n}\sum_{i=1}^{n}K(x_i)^2 \tag{S.8.178}$$

$$= 4m_2 \tag{S.8.179}$$

where m_2 denotes the second empirical moment,

$$m_2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2. \tag{S.8.180}$$

With Equation (8.32), the score matching objective thus is

$$J(\theta) = 2\theta + \frac{1}{2}4m_2\theta^2 \tag{S.8.181}$$

$$= 2\theta + 2m_2\theta^2 \tag{S.8.182}$$

A necessary condition for the minimiser to satisfy is

$$\frac{\partial J(\theta)}{\partial \theta} = 2 + 4\theta m_2 \tag{S.8.183}$$

$$= 0 \tag{S.8.184}$$

The only parameter value that satisfies the condition is

$$\hat{\theta} = -\frac{1}{2m_2}. \tag{S.8.185}$$

The second derivative of $J(\theta)$ is

$$\frac{\partial^2 J(\theta)}{\theta^2} = m_2, \tag{S.8.186}$$

which is positive (as long as all data points are non-zero). Hence $\hat{\theta}$ is a minimiser.

From the relation $\theta = -1/(2\sigma^2)$, we obtain that the score matching estimate of the variance σ^2 is

$$\hat{\sigma}^2 = -\frac{1}{2\hat{\theta}} = m_2. \tag{S.8.187}$$

We can obtain the score matching estimate $\hat{\sigma}^2$ from $\hat{\theta}$ in this manner for the same reason that we were able to work with transformed parameters in maximum likelihood estimation.

For zero mean Gaussians, the second moment m_2 is the maximum likelihood estimate of the variance, which shows that the score matching and maximum likelihood estimate are here the same. While the two methods generally yield different estimates, the result also holds for multivariate Gaussians where the score matching estimates also equal the maximum likelihood estimates, see the original article on score matching by Hyvärinen (2005).

8.12 Maximum likelihood estimation and unnormalised models

Consider the Ising model for two binary random variables (x_1, x_2),

$$p(x_1, x_2; \theta) \propto \exp\left(\theta x_1 x_2 + x_1 + x_2\right), \qquad x_i \in \{-1, 1\},$$

(a) Compute the partition function $Z(\theta)$.

Solution. The definition of the partition function is

$$Z(\theta) = \sum_{\{-1,1\}^2} \exp\left(\theta x_1 x_2 + x_1 + x_2\right). \tag{S.8.188}$$

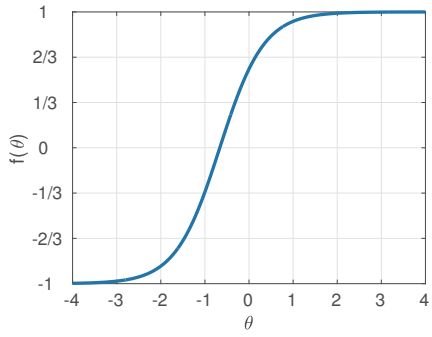
where have have to sum over $(x_1, x_2) \in \{-1, 1\}^2 = \{(-1, 1), (1, 1), (1, -1)\,(-1-1)\}$. This gives

$$Z(\theta) = \exp(-\theta - 1 + 1) + \exp(\theta + 2) + \exp(-\theta + 1 - 1) + \exp(\theta - 2) \tag{S.8.189}$$

$$= 2\exp(-\theta) + \exp(\theta + 2) + \exp(\theta - 2) \tag{S.8.190}$$

(b) The figure below shows the graph of $f(\theta) = \frac{\partial \log Z(\theta)}{\partial \theta}$.

Assume you observe three data points (x_1, x_2) equal to $(-1, -1)$, $(-1, 1)$, and $(1, -1)$. Using the figure, what is the maximum likelihood estimate of θ? Justify your answer.



Solution. Denoting the i-th observed data point by (x_1^i, x_2^i), the log-likelihood is

$$\ell(\theta) = \sum_{i=1}^{n} \log p(x_1^i, x_2^i; \theta) \tag{S.8.191}$$

Inserting the definition of the $p(x_1, x_2; \theta)$ yields

$$\ell(\theta) = \sum_{i=1}^{n} \left[\theta x_1^i x_2^i + x_1^i + x_2^i\right] - n \log Z(\theta) \tag{S.8.192}$$

$$= \theta \sum_{i=1}^{n} \left[x_1^i x_2^i\right] + \sum_{i=1}^{n} \left[x_1^i + x_2^i\right] - n \log Z(\theta) \tag{S.8.193}$$

Its derivative with respect to the θ is

$$\frac{\partial \ell(\theta)}{\partial \theta} = \sum_{i=1}^{n} \left[x_1^i x_2^i\right] - n \frac{\partial \log Z(\theta)}{\partial \theta} \tag{S.8.194}$$

$$= \sum_{i=1}^{n} \left[x_1^i x_2^i\right] - n f(\theta) \tag{S.8.195}$$

Setting it to zero yields

$$\frac{1}{n} \sum_{i=1}^{n} \left[x_1^i x_2^i\right] = f(\theta) \tag{S.8.196}$$

An alternative approach is to start with the more general relationship that relates the gradient of the partition function to the gradient of the log unnormalised model. For example, if

$$p(\mathbf{x}, \boldsymbol{\theta}) = \frac{\phi(\mathbf{x}; \boldsymbol{\theta})}{Z(\boldsymbol{\theta})}$$

we have

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p(\mathbf{x}_i; \boldsymbol{\theta}) \tag{S.8.197}$$

$$= \sum_{i=1}^{n} \log \phi(\mathbf{x}_i; \boldsymbol{\theta}) - n \log Z(\boldsymbol{\theta}) \tag{S.8.198}$$

Setting the derivative to zero gives,

$$\frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log \phi(\mathbf{x}_i; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})$$

In either case, numerical evaluation of $1/n \sum_{i=1}^{n} x_1^i x_2^i$ gives

$$\frac{1}{n} \sum_{i=1}^{n} \left[x_1^i x_2^i \right] = \frac{1}{3} (1 - 1 - 1) \tag{S.8.199}$$

$$= -\frac{1}{3} \tag{S.8.200}$$

From the graph, we see that $f(\theta)$ takes on the value $-1/3$ for $\theta = -1$, which is the desired MLE.

8.13 Parameter estimation for unnormalised models

Let $p(\mathbf{x}; \mathbf{A}) \propto \exp(-\mathbf{x}^\top \mathbf{A} \mathbf{x})$ be a parametric statistical model for $\mathbf{x} = (x_1, \ldots, x_{100})$, where the parameters are the elements of the matrix $\mathbf{A}$. Assume that $\mathbf{A}$ is symmetric and positive semi-definite, i.e. $\mathbf{A}$ satisfies $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$ for all values of $\mathbf{x}$.

(a) For n iid data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, a friend proposes to estimate $\mathbf{A}$ by maximising $J(\mathbf{A})$,

$$J(\mathbf{A}) = \prod_{k=1}^{n} \exp\left(-\mathbf{x}_k^\top \mathbf{A} \mathbf{x}_k \right). \tag{8.40}$$

Explain why this procedure cannot give reasonable parameter estimates.

Solution. We have that $\mathbf{x}_k^\top \mathbf{A} \mathbf{x}_k \geq 0$ so that $\exp\left(-\mathbf{x}_k^\top \mathbf{A} \mathbf{x}_k\right) \leq 1$. Hence $\exp\left(-\mathbf{x}_k^\top \mathbf{A} \mathbf{x}_k\right)$ is maximal if the elements of $\mathbf{A}$ are zero. This means that $J(\mathbf{A})$ is maximal if $\mathbf{A} = 0$ whatever the observed data, which does not correspond to a meaningful estimation procedure (estimator).

(b) Explain why maximum likelihood estimation is easy when the x_i are real numbers, i.e. $x_i \in \mathbb{R}$, while typically very difficult when the x_i are binary, i.e. $x_i \in \{0, 1\}$.

Solution. For maximum likelihood estimation, we needed to normalise the model by computing the partition function $Z(\boldsymbol{\theta})$, which is defined as the sum/integral of $\exp(-\mathbf{x}^\top \mathbf{A}\mathbf{x})$ over the domain of $\mathbf{x}$.

When the x_i are numbers, we can here obtain an analytical expression for $Z(\boldsymbol{\theta})$. However, if the x_i are binary, no such analytical expression is available and computing $Z(\boldsymbol{\theta})$ is then very costly.

(c) Can we use score matching instead of maximum likelihood estimation to learn $\mathbf{A}$ if the x_i are binary?

Solution. No, score matching cannot be used for binary data.

Chapter 9

Sampling and Monte Carlo Integration

Exercises

9.1 Importance sampling to estimate tail probabilities (based on Robert and Casella, 2010, Exercise 3.5)

We would like to use importance sampling to compute the probability that a standard Gaussian random variable x takes on a value larger than 5, i.e

$$\mathbb{P}(x > 5) = \int_5^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \tag{9.1}$$

We know that the probability equals

$$\begin{aligned}
\mathbb{P}(x > 5) &= 1 - \int_{-\infty}^5 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx && (9.2)\\
&= 1 - \Phi(5) && (9.3)\\
&\approx 2.87 \cdot 10^{-7} && (9.4)
\end{aligned}$$

where $\Phi(.)$ is the cumulative distribution function of a standard normal random variable.

(a) With the indicator function $\mathbb{1}_{x>5}(x)$, which equals one if x is larger than 5 and zero otherwise, we can write $\mathbb{P}(x > 5)$ in form of the expectation

$$\mathbb{P}(x > 5) = \mathbb{E}[\mathbb{1}_{x>5}(x)], \tag{9.5}$$

where the expectation is taken with respect to the density $\mathcal{N}(x; 0, 1)$ of a standard normal random variable,

$$\mathcal{N}(x; 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \tag{9.6}$$

This suggests that we can approximate $\mathbb{P}(x > 5)$ by a Monte Carlo average

$$\mathbb{P}(x > 5) \approx \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{x>5}(x_i), \qquad x_i \sim \mathcal{N}(x; 0, 1). \tag{9.7}$$

Explain why this approach does not work well.

Solution. In this approach, we essentially count how many times the x_i are larger than 5. However, we know that the chance that $x_i > 5$ is only $2.87 \cdot 10^{-7}$. That is, we only get about one value above 5 every 20 million simulations! The approach is thus very sample inefficient.

(b) Another approach is to use importance sampling with an importance distribution $q(x)$ that is zero for $x < 5$. We can then write $\mathbb{P}(x > 5)$ as

$$\begin{aligned}
\mathbb{P}(x > 5) &= \int_5^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx && (9.8)\\
&= \int_5^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \frac{q(x)}{q(x)} dx && (9.9)\\
&= \mathbb{E}_{q(x)} \left[\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \frac{1}{q(x)}\right] && (9.10)
\end{aligned}$$

and estimate $\mathbb{P}(x > 5)$ as a sample average.

We here use an exponential distribution shifted by 5 to the right. It has pdf

$$q(x) = \begin{cases} \exp(-(x-5)) & \text{if } x \geq 5 \\ 0 & \text{otherwise} \end{cases} \tag{9.11}$$

For background on the exponential distribution, see e.g. https://en.wikipedia.org/wiki/Exponential_distribution.

Provide a formula that approximates $\mathbb{P}(x > 5)$ as a sample average over n samples $x_i \sim q(x)$.

Solution. The provided equation

$$\mathbb{P}(x > 5) = \mathbb{E}_{q(x)} \left[\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \frac{1}{q(x)} \right] \tag{S.9.1}$$

can be approximated as a sample average as follows:

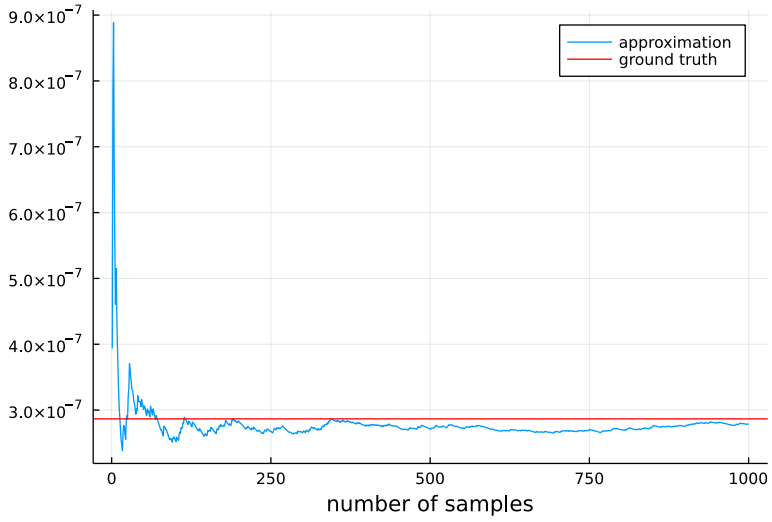
$$\mathbb{P}(x > 5) \approx \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_i^2}{2}\right) \frac{1}{q(x_i)} \tag{S.9.2}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_i^2}{2} + x - 5\right) \tag{S.9.3}$$

with $x_i \sim q(x)$.

(c) Numerically compute the importance estimate for various sample sizes $n \in [0, 1000]$. Plot the estimate against the sample size and compare with the ground truth value.

Solution. The following figure shows the importance sampling estimate as a function of the sample size (numbers do depend on the random seed used). We can see that we can obtain a good estimate with a few hundred samples already.



Python code is as follows.

```
import numpy as np
from numpy.random import default_rng
import matplotlib.pyplot as plt
from scipy.stats import norm

n = 1000
alpha = 5

# compute the tail probability
p = 1-norm.cdf(alpha)

# sample from the importance distribution
rng = default_rng()
vals = rng.exponential(scale=1, size=n) + alpha

# compute average
def w(x):
return 1/np.sqrt(2*np.pi)*np.exp(-x**2/2+x-alpha)

Ihat = np.cumsum(w(vals))/ np.arange(1, n+1)

# plot
plt.plot(Ihat)
plt.axhline(y=p, color="r")
plt.xlabel("number of samples")
```

And code in Julia is:

```
using Distributions
using Plots
using Statistics

# compute the tail probability
phi(x) = cdf(Normal(0,1),x)
alpha = 5
p = (1-phi(alpha))

# sample from the importance distribution
n = 1000
exprv = Exponential(1)
x = rand(exprv, n).+alpha;

# compute the approximation
w(x) = 1/sqrt(2*pi)*exp(-x^2/2+x-alpha)
#w(x) = pdf(Normal(0,1),x)/pdf(exprv, x-alpha);

Ihat = zeros(length(x));
for k in 1:length(x)
```

```
    lhat[k] = mean(w.(x[1:k]));
end

# plot
plt=plot(lhat, label="approximation");
hline!([p], color=:red, label="ground truth")
xlabel!("number of samples")
```

9.2 Monte Carlo integration and importance sampling

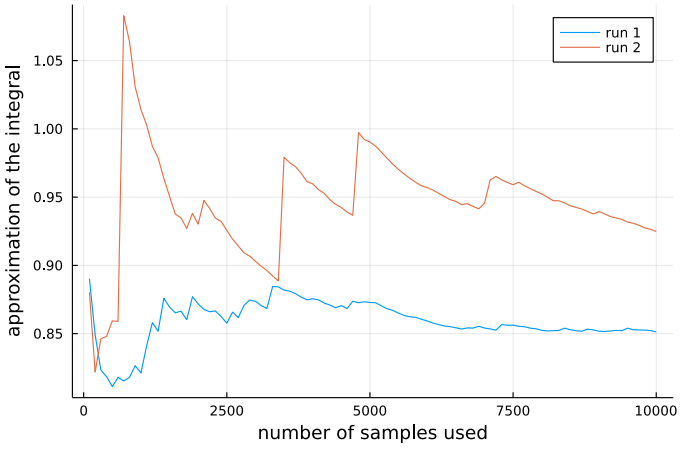
A standard Cauchy distribution has the density function (pdf)

$$p(x) = \frac{1}{\pi}\frac{1}{1+x^2} \tag{9.12}$$

with $x \in \mathbb{R}$. A friend would like to verify that $\int p(x)dx = 1$ but doesn't quite know how to solve the integral analytically. They thus use importance sampling and approximate the integral as

$$\int p(x)dx \approx \frac{1}{n}\sum_{i=1}^{n}\frac{p(x_i)}{q(x_i)} \qquad x_i \sim q \tag{9.13}$$

where q is the density of the auxiliary/importance distribution. Your friend chooses a standard normal density for q and produces the following figure:



The figure shows two independent runs. In each run, your friend computes the approximation with different sample sizes by subsequently including more and more x_i in the approximation, so that, for example, the approximation with $n = 2000$ shares the first 1000 samples with the approximation that uses $n = 1000$.

Your friend is puzzled that the two runs give rather different results (which are not equal to one), and also that within each run, the estimate very much depends on the sample size. Explain these findings.

Solution. While the estimate $\hat{I}_n$

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n}\frac{p(x_i)}{q(x_i)} \tag{S.9.4}$$

is unbiased by construction, we have to check whether its second moment is finite. Otherwise, we have an invalid estimator that behaves erratically in practice. The ratio $w(x)$ between $p(x)$ and $q(x)$ equals

$$w(x) = \frac{p(x)}{q(x)} \tag{S.9.5}$$

$$= \frac{\frac{1}{\pi}\frac{1}{1+x^2}}{\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)} \tag{S.9.6}$$

which can be simplified to

$$w(x) = \frac{\sqrt{2\pi}\exp(x^2/2)}{\pi(1+x^2)}. \tag{S.9.7}$$

The second moment of $w(x)$ under $q(x)$ thus is

$$\mathbb{E}_{q(x)}\left[w(x)^2\right] = \int_{-\infty}^{\infty}\frac{2\pi}{\pi^2}\frac{\exp(x^2)}{(1+x^2)^2}q(x)dx \tag{S.9.8}$$

$$= \int_{-\infty}^{\infty}\frac{2\pi}{\pi^2}\frac{\exp(x^2)}{(1+x^2)^2}\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)dx \tag{S.9.9}$$

$$\propto \int_{-\infty}^{\infty}\frac{\exp(x^2/2)}{(1+x^2)^2}dx \tag{S.9.10}$$

The exponential function grows more quickly than any polynomial so that the integral becomes arbitrarily large. Hence, the second moment (and the variance) of $\hat{I}_n$ is unbounded, which explains the erratic behaviour of the curves in the plot.

A less formal but quicker way to see that, for this problem, a standard normal is a poor choice of an importance distribution is to note that its density decays more quickly than the Cauchy pdf in (9.12), which means that the standard normal pdf is "small" when the Cauchy pdf is still "large" (see Figure 9.1). This leads to large variance of the estimate. The overall conclusion is that the integral $\int p(x)dx$ should not be approximated with importance sampling with a Gaussian importance distribution.

9.3 Inverse transform sampling

The cumulative distribution function (cdf) $F_x(\alpha)$ of a (continuous or discrete) random variable x indicates the probability that x takes on values smaller or equal to α,

$$F_x(\alpha) = \mathbb{P}(x \leq \alpha). \tag{9.14}$$

For continuous random variables, the cdf is defined via the integral

$$F_x(\alpha) = \int_{-\infty}^{\alpha} p_x(u)\mathrm{d}u, \tag{9.15}$$

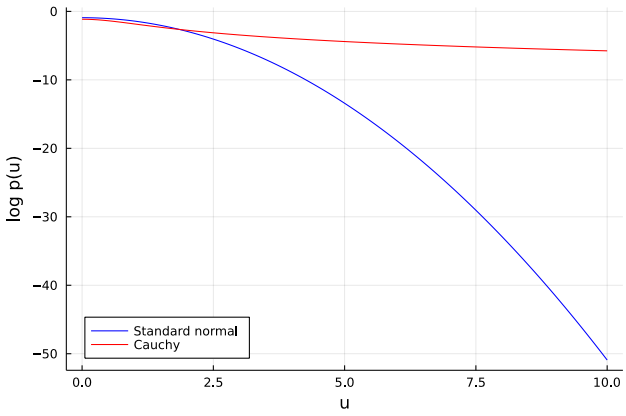


Figure 9.1: Exercise 9.2. Comparison of the log pdf of a standard normal (blue) and the Cauchy random variable (red) for positive inputs. The Cauchy pdf has much heavier tails than a Gaussian so that the Gaussian pdf is already "small" when the Cauchy pdf is still "large".

where p_x denotes the pdf of the random variable x (u is here a dummy variable). Note that F_x maps the domain of x to the interval $[0, 1]$. For simplicity, we here assume that F_x is invertible.

For a continuous random variable x with cdf F_x show that the random variable $y = F_x(x)$ is uniformly distributed on $[0, 1]$.

Importantly, this implies that for a random variable y which is uniformly distributed on $[0, 1]$, the transformed random variable $F_x^{-1}(y)$ has cdf F_x. This gives rise to a method called "inverse transform sampling" to generate n iid samples of a random variable x with cdf F_x. Given a target cdf F_x, the method consists of:

- calculating the inverse F_x^{-1}
- sampling n iid random variables uniformly distributed on $[0, 1]$: $y^{(i)} \sim \mathcal{U}(0, 1)$, $i = 1, \ldots, n$.
- transforming each sample by F_x^{-1}: $x^{(i)} = F_x^{-1}(y^{(i)})$, $i = 1, \ldots, n$.

By construction of the method, the $x^{(i)}$ are n iid samples of x.

Solution. We start with the cumulative distribution function (cdf) F_y for y,

$$F_y(\beta) = \mathbb{P}(y \leq \beta). \tag{S.9.11}$$

Since $F_x(x)$ maps x to $[0, 1]$, $F_y(\beta)$ is zero for $\beta < 0$ and one for $\beta > 1$. We next consider $\beta \in [0, 1]$.

Let α be the value of x that F_x maps to β, i.e. $F_x(\alpha) = \beta$, which means $\alpha = F_x^{-1}(\beta)$. Since F_x is a non-decreasing function, we have

$$F_y(\beta) = \mathbb{P}(y \leq \beta) = \mathbb{P}(F_x(x) \leq \beta) = \mathbb{P}(x \leq F_x^{-1}(\beta)) = \mathbb{P}(x \leq \alpha) = F_x(\alpha). \tag{S.9.12}$$

Since $\alpha = F_x^{-1}(\beta)$ we obtain

$$F_y(\beta) = F_x(F_x^{-1}(\beta)) = \beta \tag{S.9.13}$$

The cdf F_y is thus given by

$$F_y(\beta) = \begin{cases} 0 & \text{if } \beta < 0 \\ \beta & \text{if } \beta \in [0,1] \\ 1 & \text{if } \beta > 1 \end{cases} \tag{S.9.14}$$

which is the cdf of a uniform random variable on $[0,1]$. Hence $y = F_x(x)$ is uniformly distributed on $[0,1]$.

9.4 Sampling from the exponential distribution

The exponential distribution has the density

$$p(x;\lambda) = \begin{cases} \lambda \exp(-\lambda x) & x \geq 0 \\ 0 & x < 0, \end{cases} \tag{9.16}$$

where λ is a parameter of the distribution. Use inverse transform sampling to generate n iid samples from $p(x;\lambda)$.

Solution. We first compute the cumulative distribution function.

$$\begin{aligned} F_x(\alpha) &= \mathbb{P}(x \leq \alpha) && \text{(S.9.15)} \\ &= \int_0^\alpha \lambda \exp(-\lambda x) && \text{(S.9.16)} \\ &= -\exp(-\lambda)\big|_0^\alpha && \text{(S.9.17)} \\ &= 1 - \exp(-\lambda\alpha) && \text{(S.9.18)} \end{aligned}$$

It's inverse is obtained by solving

$$y = 1 - \exp(-\lambda x) \tag{S.9.19}$$

for x, which gives:

$$\begin{aligned} \exp(-\lambda x) &= 1 - y && \text{(S.9.20)} \\ -\lambda x &= \log(1-y) && \text{(S.9.21)} \\ x &= \frac{-\log(1-y)}{\lambda} && \text{(S.9.22)} \end{aligned}$$

To generate samples $x^{(i)} \sim p(x;\lambda)$, we thus first sample $y^{(i)} \sim U(0,1)$, and then set

$$x^{(i)} = \frac{-\log(1-y^{(i)})}{\lambda}. \tag{S.9.23}$$

Inverse transform sampling can be used to generate samples from many standard distributions. For example, it allows one to generate Gaussian random variables from uniformly distributed random variables. The method is called the Box-Muller transform,

see e.g. https://en.wikipedia.org/wiki/Box-Muller_transform. How to generate the required samples from the uniform distribution is a research field on its own, see e.g. https://en.wikipedia.org/wiki/Random_number_generation and (Owen, 2013, Chapter 3).

9.5 Sampling from a Laplace distribution

A Laplace random variable x of mean zero and variance one has the density $p(x)$

$$p(x) = \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}|x|\right) \qquad x \in \mathbb{R}. \tag{9.17}$$

Use inverse transform sampling to generate n iid samples from x.

Solution. The main task is to compute the cumulative distribution function (cdf) F_x of x and its inverse. The cdf is by definition

$$F_x(\alpha) = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}|u|\right) \mathrm{d}u. \tag{S.9.24}$$

We first consider the case where $\alpha \leq 0$. Since $-|u| = u$ for $u \leq 0$, we have

$$F_x(\alpha) = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2}} \exp\left(\sqrt{2}u\right) \mathrm{d}u \tag{S.9.25}$$

$$= \frac{1}{2} \exp\left(\sqrt{2}u\right) \Big|_{-\infty}^{\alpha} \tag{S.9.26}$$

$$= \frac{1}{2} \exp\left(\sqrt{2}\alpha\right). \tag{S.9.27}$$

For $\alpha > 0$, we have

$$F_x(\alpha) = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}|u|\right) \mathrm{d}u \tag{S.9.28}$$

$$= 1 - \int_{\alpha}^{\infty} \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}|u|\right) \mathrm{d}u \tag{S.9.29}$$

where we have used the fact that the pdf has to integrate to one. For values of $u > 0$, $-|u| = -u$, so that

$$F_x(\alpha) = 1 - \int_{\alpha}^{\infty} \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}u\right) \mathrm{d}u \tag{S.9.30}$$

$$= 1 + \frac{1}{2} \exp\left(-\sqrt{2}u\right) \Big|_{\alpha}^{\infty} \tag{S.9.31}$$

$$= 1 - \frac{1}{2} \exp\left(-\sqrt{2}\alpha\right). \tag{S.9.32}$$

In total, for $\alpha \in \mathbb{R}$, we thus have

$$F_x(\alpha) = \begin{cases} \frac{1}{2} \exp\left(\sqrt{2}\alpha\right) & \text{if } \alpha \leq 0 \\ 1 - \frac{1}{2} \exp\left(-\sqrt{2}\alpha\right) & \text{if } \alpha > 0 \end{cases} \tag{S.9.33}$$

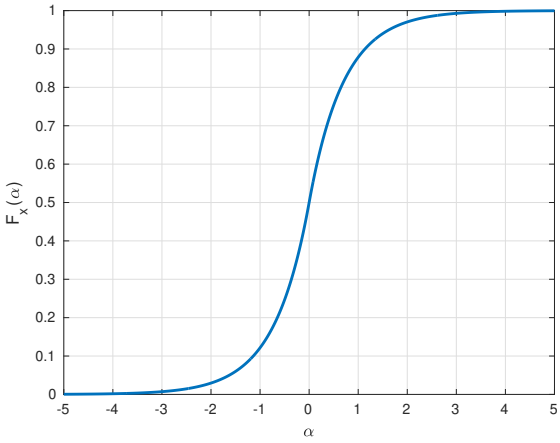


Figure 9.2: The cumulative distribution function $F_x(\alpha)$ for a Laplace distributed random variable.

Figure 9.2 visualises $F_x(\alpha)$.

As the figure suggests, there is a unique inverse to $y = F_x(\alpha)$. For $y \leq 1/2$, we have

$$y = \frac{1}{2}\exp\left(\sqrt{2}\alpha\right) \tag{S.9.34}$$

$$\log(2y) = \sqrt{2}\alpha \tag{S.9.35}$$

$$\alpha = \frac{1}{\sqrt{2}}\log(2y) \tag{S.9.36}$$

For $y > 1/2$, we have

$$y = 1 - \frac{1}{2}\exp\left(-\sqrt{2}\alpha\right) \tag{S.9.37}$$

$$-y = -1 + \frac{1}{2}\exp\left(-\sqrt{2}\alpha\right) \tag{S.9.38}$$

$$1 - y = \frac{1}{2}\exp\left(-\sqrt{2}\alpha\right) \tag{S.9.39}$$

$$\log(2 - 2y) = -\sqrt{2}\alpha \tag{S.9.40}$$

$$\alpha = -\frac{1}{\sqrt{2}}\log(2 - 2y) \tag{S.9.41}$$

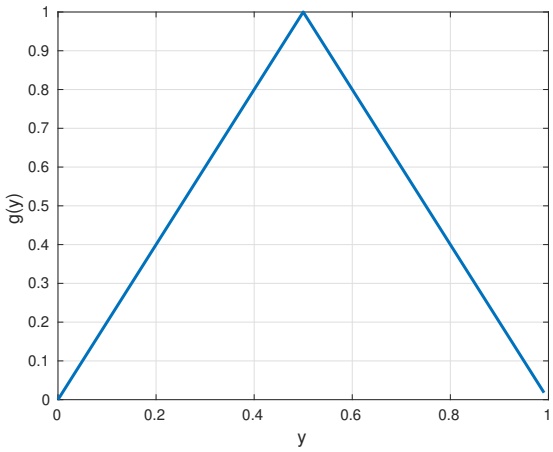
The function $y \mapsto g(y)$ that occurs in the logarithm in both cases is

$$g(y) = \begin{cases} 2y & \text{if } y \leq \frac{1}{2} \\ 2 - 2y & \text{if } y > \frac{1}{2} \end{cases}. \tag{S.9.42}$$

It is shown below and can be written more compactly as $g(y) = 1 - 2|y - 1/2|$.

We thus can write the inverse $F_x^{-1}(y)$ of the cdf $y = F_x(\alpha)$ as

$$F_x^{-1}(y) = -\text{sign}\left(y - \frac{1}{2}\right)\frac{1}{\sqrt{2}}\log\left[1 - 2\left|y - \frac{1}{2}\right|\right]. \tag{S.9.43}$$



To generate n iid samples from x, we first generate n iid samples $y^{(i)}$ that are uniformly distributed on $[0, 1]$, and then compute for each $F_x^{-1}(y^{(i)})$. The properties of inverse transform sampling guarantee that the $x^{(i)}$,

$$x^{(i)} = F_x^{-1}(y^{(i)}), \tag{S.9.44}$$

are independent and Laplace distributed.

9.6 Rejection sampling (based on Robert and Casella, 2010, Exercise 2.8)

Most compute environments provide functions to sample from a standard normal distribution. Popular algorithms include the Box-Muller transform, see e.g. `https://en.wikipedia.org/wiki/Box-Muller_transform`. We here use rejection sampling to sample from a standard normal distribution with density $p(x)$ using a Laplace distribution as our proposal/auxiliary distribution.

The density $q(x)$ of a zero-mean Laplace distribution with variance $2b^2$ is

$$q(x; b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{9.18}$$

We can sample from it by sampling a Laplace variable with variance 1 as in Exercise 9.5 and then scaling the sample by $\sqrt{2}b$.

Rejection sampling then repeats the following steps:

- Generate $x \sim q(x; b)$
- Accept x with probability $f(x) = \frac{1}{M}\frac{p(x)}{q(x)}$, i.e. generate $u \sim U(0, 1)$ and accept x if $u \leq f(x)$.

(a) Compute the ratio $M(b) = \max_x \frac{p(x)}{q(x;b)}$.

Solution. By the definitions of the pdf $p(x)$ of a standard normal and the pdf $q(x;b)$ of the Laplace distribution, we have

$$\frac{p(x)}{q(x;b)} = \frac{\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)}{\frac{1}{2b}\exp(-|x|/b)} \tag{S.9.45}$$

$$= \frac{2b}{\sqrt{2\pi}}\exp(-x^2/2 + |x|/b) \tag{S.9.46}$$

The ratio is symmetric in x. Moreover, since the exponential function is strictly increasing, we can find the maximiser of $-x^2/2 + x/b$ for $x \geq 0$ to determine the maximiser of $M(b)$. With $g(x) = -x^2/2 + x/b$, we have

$$g'(x) = -x + 1/b \tag{S.9.47}$$

$$g''(x) = -1 \tag{S.9.48}$$

The critical point (for which the first derivative is zero) is $x = 1/b$ and since the second derivative is negative for all x, the point is a maximum. The maximal ratio $M(b)$ thus is

$$M(b) = \frac{2b}{\sqrt{2\pi}}\exp\left(-x^2/2 + |x|/b\right)\Big|_{x=1/b} \tag{S.9.49}$$

$$= \frac{2b}{\sqrt{2\pi}}\exp\left(-1/(2b^2) + 1/b^2\right) \tag{S.9.50}$$

$$= \frac{2b}{\sqrt{2\pi}}\exp\left(1/(2b^2)\right) \tag{S.9.51}$$

(b) How should you choose b to maximise the probability of acceptance?

Solution. The probability of acceptance is $1/M$. Hence to maximise it, we have to choose b such that $M(b)$ is minimal. We compute the derivatives

$$M'(b) = \frac{2}{\sqrt{2\pi}}\exp(1/(2b^2)) - \frac{2b}{\sqrt{2\pi}}\exp(1/(2b^2))b^{-3} \tag{S.9.52}$$

$$= \frac{2}{\sqrt{2\pi}}\exp(1/(2b^2)) - \frac{2}{\sqrt{2\pi}}\exp(1/(2b^2))b^{-2} \tag{S.9.53}$$

$$= \frac{2}{\sqrt{2\pi}}\exp(1/(2b^2))(1 - b^{-2}) \tag{S.9.54}$$

$$M''(b) = -b^{-3}\frac{2}{\sqrt{2\pi}}\exp(1/(2b^2))(1 - b^{-2}) + 2b^{-3}\frac{2}{\sqrt{2\pi}}\exp(1/(2b^2)) \tag{S.9.55}$$

$$\tag{S.9.56}$$

Setting the first derivative to zero gives

$$\frac{2}{\sqrt{2\pi}}\exp(1/(2b^2)) = \frac{2}{\sqrt{2\pi}}\exp(1/(2b^2))b^{-2} \tag{S.9.57}$$

$$1 = b^{-2} \tag{S.9.58}$$

Hence the optimal $b = 1$. The second derivative at $b = 1$ is

$$M''(1) = 2\frac{2}{\sqrt{2\pi}}\exp(1/2) \tag{S.9.59}$$

which is positive so that the $b = 1$ is a minimum. The smallest value of M thus is

$$M(1) = \frac{2b}{\sqrt{2\pi}} \exp(1/(2b^2))\Big|_{b=1} \tag{S.9.60}$$

$$= \frac{2}{\sqrt{2\pi}} \exp(1/2) \tag{S.9.61}$$

$$= \sqrt{\frac{2e}{\pi}} \tag{S.9.62}$$

where $e = \exp(1)$. The maximal acceptance probability thus is

$$\frac{1}{\min_b M(b)} = \sqrt{\frac{\pi}{2e}} \tag{S.9.63}$$

$$\approx 0.76 \tag{S.9.64}$$

This means for each sample x generated from $q(x; 1)$, there is chance of 0.76 that it gets accepted. In other words, for each accepted sample, we need to generate $1/0.76 = 1.32$ samples from $q(x; 1)$.

The variance of the Laplace distribution for $b = 1$ equals 2. Hence the variance of the auxiliary distribution is larger (twice as large) as the variance of the distribution we would like to sample from.

(c) Assume you sample from $p(x_1, \ldots, x_d) = \prod_{i=1}^d p(x_i)$ using $q(x_1, \ldots, x_d) = \prod_{i=1}^d q(x_i; b)$ as auxiliary distribution without exploiting any independencies. How does the acceptance probability scale as a function of d? You may denote the acceptance probability in case of $d = 1$ by A.

Solution. We have to determine the maximal ratio

$$M_d = \max_{x_1,\ldots,x_d} \frac{p(x_1, \ldots, x_d)}{q(x_1, \ldots, x_d)} \tag{S.9.65}$$

Plugging-in the factorisation gives

$$M_d = \max_{x_1,\ldots,x_d} \prod_{i=1}^d \frac{p(x_i)}{q(x_i)} \tag{S.9.66}$$

$$= \prod_{i=1}^d \underbrace{\max_{x_i} \frac{p(x_i)}{q(x_i)}}_{M_1 = 1/A} \tag{S.9.67}$$

$$= \prod_{i=1}^d \frac{1}{A} \tag{S.9.68}$$

$$= \frac{1}{A^d} \tag{S.9.69}$$

Hence, the acceptance probability is

$$\frac{1}{M_d} = A^d \tag{S.9.70}$$

Note that $A \leq 1$ since it is a probability. This means that, unless $A = 1$, we have an acceptance probability that decays exponentially in the number of dimensions if the target and auxiliary distributions factorise and we do not exploit the independencies.

9.7 Sampling from a restricted Boltzmann machine

The restricted Boltzmann machine (RBM) is a model for binary variables $\mathbf{v} = (v_1, \ldots, v_n)^\top$ and $\mathbf{h} = (h_1, \ldots, h_m)^\top$ which asserts that the joint distribution of $(\mathbf{v}, \mathbf{h})$ can be described by the probability mass function

$$p(\mathbf{v}, \mathbf{h}) \propto \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h}\right), \tag{9.19}$$

where $\mathbf{W}$ is a $n \times m$ matrix, and $\mathbf{a}$ and $\mathbf{b}$ vectors of size n and m, respectively. Both the v_i and h_i take values in $\{0, 1\}$. The v_i are called the "visibles" variables since they are assumed to be observed while the h_i are the hidden variables since it is assumed that we cannot measure them.

Explain how to use Gibbs sampling to generate samples from the marginal $p(\mathbf{v})$,

$$p(\mathbf{v}) = \frac{\sum_{\mathbf{h}} \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h}\right)}{\sum_{\mathbf{h},\mathbf{v}} \exp\left(\mathbf{v}^\top \mathbf{W}\mathbf{h} + \mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h}\right)}, \tag{9.20}$$

for any given values of $\mathbf{W}$, $\mathbf{a}$, and $\mathbf{b}$.

Hint: You may use that

$$p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{m} p(h_i|\mathbf{v}), \qquad p(h_i = 1|\mathbf{v}) = \frac{1}{1 + \exp\left(-\sum_j v_j W_{ji} - b_i\right)}, \tag{9.21}$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{n} p(v_i|\mathbf{h}), \qquad p(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp\left(-\sum_j W_{ij} h_j - a_i\right)}. \tag{9.22}$$

Solution. In order to generate samples $\mathbf{v}^{(k)}$ from $p(\mathbf{v})$ we generate samples $(\mathbf{v}^{(k)}, \mathbf{h}^{(k)})$ from $p(\mathbf{v}, \mathbf{h})$ and then ignore the $\mathbf{h}^{(k)}$.

Gibbs sampling is a MCMC method to produce a sequence of samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \ldots$ that follow a pdf/pmf $p(\mathbf{x})$ (if the chain is run long enough). Assuming that $\mathbf{x}$ is d-dimensional, we generate the next sample $\mathbf{x}^{(k+1)}$ in the sequence from the previous sample $\mathbf{x}^{(k)}$ by:

1. picking (randomly) an index $i \in \{1, \ldots, d\}$
2. sampling $x_i^{(k+1)}$ from $p(x_i \mid \mathbf{x}_{\setminus i}^{(k)})$ where $\mathbf{x}_{\setminus i}^{(k)}$ is vector $\mathbf{x}$ with x_i removed, i.e. $\mathbf{x}_{\setminus i}^{(k)} = (x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_{i+1}^{(k)}, \ldots, x_d^{(k)})$
3. setting $\mathbf{x}^{(k+1)} = (x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k+1)}, x_{i+1}^{(k)}, \ldots, x_d^{(k)})$.

For the RBM, the tuple $(\mathbf{h}, \mathbf{v})$ corresponds to $\mathbf{x}$ so that a x_i in the above steps can either be a hidden variable or a visible. Hence

$$p(x_i \mid \mathbf{x}_{\setminus i}) = \begin{cases} p(h_i \mid \mathbf{h}_{\setminus i}, \mathbf{v}) & \text{if } x_i \text{ is a hidden variable } h_i \\ p(v_i \mid \mathbf{v}_{\setminus i}, \mathbf{h}) & \text{if } x_i \text{ is a visible variable } v_i \end{cases} \tag{S.9.71}$$

($\mathbf{h}_{\setminus i}$ denotes the vector $\mathbf{h}$ with element h_i removed, and equivalently for $\mathbf{v}_{\setminus i}$)

To compute the conditionals on the right hand side, we use the hint:

$$p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{m} p(h_i|\mathbf{v}), \qquad p(h_i = 1|\mathbf{v}) = \frac{1}{1 + \exp\left(-\sum_j v_j W_{ji} - b_i\right)}, \tag{S.9.72}$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{n} p(v_i|\mathbf{h}), \qquad p(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp\left(-\sum_j W_{ij} h_j - a_i\right)}. \tag{S.9.73}$$

Given the independencies between the hiddens given the visibles and vice versa, we have

$$p(h_i \mid \mathbf{h}_{\setminus i}, \mathbf{v}) = p(h_i \mid \mathbf{v}) \qquad p(v_i \mid \mathbf{v}_{\setminus i}, \mathbf{h}) = p(v_i \mid \mathbf{h}) \tag{S.9.74}$$

so that the expressions for $p(h_i = 1|\mathbf{v})$ and $p(v_i = 1|\mathbf{h})$ allow us to implement the Gibbs sampler.

Given the independencies, it makes further sense to sample the $\mathbf{h}$ and $\mathbf{v}$ variables in blocks: first we sample all the h_i given $\mathbf{v}$, and then all the v_i given the $\mathbf{h}$ (or vice versa). This is also known as block Gibbs sampling.

In summary, given a sample $(\mathbf{h}^{(k)}, \mathbf{v}^{(k)})$, we thus generate the next sample $(\mathbf{h}^{(k+1)}, \mathbf{v}^{(k+1)})$ in the sequence as follows:

- For all h_i, $i = 1, \ldots, m$:
  - compute $p_i^h = p(h_i = 1|\mathbf{v}^{(k)})$
  - sample u_i from a uniform distribution on $[0, 1]$ and set $h_i^{(k+1)}$ to 1 if $u_i \leq p_i^h$.
- For all v_i, $i = 1, \ldots, n$:
  - compute $p_i^v = p(v_i = 1|\mathbf{h}^{(k+1)})$
  - sample u_i from a uniform distribution on $[0, 1]$ and set $v_i^{(k+1)}$ to 1 if $u_i \leq p_i^v$.

As final step, after sampling S pairs $(\mathbf{h}^{(k)}, \mathbf{v}^{(k)})$, $k = 1, \ldots, S$, the set of visibles $\mathbf{v}^{(k)}$ form samples from the marginal $p(\mathbf{v})$.

9.8 Basic Markov chain Monte Carlo inference

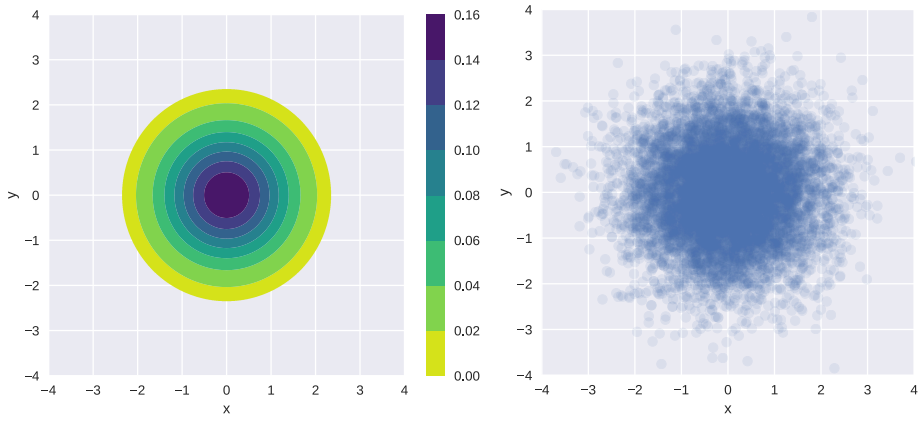
This exercise is on sampling and approximate inference by Markov chain Monte Carlo (MCMC). MCMC can be used to obtain samples from a probability distribution, e.g. a posterior distribution. The samples approximately represent the distribution, as illustrated in Figure 9.3, and can be used to approximate expectations.

We denote the density of a zero mean Gaussian with variance σ^2 by $\mathcal{N}(x; \mu, \sigma^2)$, i.e.

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{9.23}$$

Consider a vector of d random variables $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ and some observed data $\mathcal{D}$. In many cases, we are interested in computing expectations under the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$, e.g.

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[g(\boldsymbol{\theta})\right] = \int g(\boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) \mathrm{d}\boldsymbol{\theta} \tag{9.24}$$



(a) True density (b) Density represented by 10,000 samples.

Figure 9.3: Density and samples from $p(x, y) = \mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$.

for some function $g(\boldsymbol{\theta})$. If d is small, e.g. $d \leq 3$, deterministic numerical methods can be used to approximate the integral to high accuracy, see e.g. https://en.wikipedia.org/wiki/Numerical_integration. But for higher dimensions, these methods are generally not applicable any more. The expectation, however, can be approximated as a sample average if we have samples $\boldsymbol{\theta}^{(i)}$ from $p(\boldsymbol{\theta} \mid \mathcal{D})$:

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[g(\boldsymbol{\theta})\right] \approx \frac{1}{S}\sum_{i=1}^{S} g(\boldsymbol{\theta}^{(i)}) \tag{9.25}$$

Note that in MCMC methods, the samples $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(S)}$ used in the above approximation are typically not statistically independent.

Metropolis-Hastings is an MCMC algorithm that generates samples from a distribution $p(\boldsymbol{\theta})$, where $p(\boldsymbol{\theta})$ can be any distribution on the parameters (and not only posteriors). The algorithm is iterative and at iteration t, it uses:

- a proposal distribution $q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$, parametrised by the current state of the Markov chain, i.e. $\boldsymbol{\theta}^{(t)}$;
- a function $p^*(\boldsymbol{\theta})$, which is proportional to $p(\boldsymbol{\theta})$. In other words, $p^*(\boldsymbol{\theta})$ is unnormalised[1] and the normalised density $p(\boldsymbol{\theta})$ is

$$p(\boldsymbol{\theta}) = \frac{p^*(\boldsymbol{\theta})}{\int p^*(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}. \tag{9.26}$$

For all tasks in this exercise, we work with a Gaussian proposal distribution $q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$, whose mean is the previous sample in the Markov chain, and whose variance is ϵ^2. That is, at iteration t of our Metropolis-Hastings algorithm,

$$q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}) = \prod_{k=1}^{d} \mathcal{N}(\theta_k; \theta_k^{(t-1)}, \epsilon^2). \tag{9.27}$$

When used with this proposal distribution, the algorithm is called Random Walk Metropolis-Hastings algorithm.

[1] We here follow the notation of Barber (2012); $\tilde{p}$ or ϕ are often to denote unnormalised models too.

(a) Read Section 27.4 of Barber (2012) to familiarise yourself with the Metropolis-Hastings algorithm.

(b) Write a function mh implementing the Metropolis Hasting algorithm, as given in Algorithm 27.3 in Barber (2012), using the Gaussian proposal distribution in (9.27) above. The function should take as arguments

- p_star: a function on $\boldsymbol{\theta}$ that is proportional to the density of interest $p(\boldsymbol{\theta})$;
- param_init: the initial sample — a value for $\boldsymbol{\theta}$ from where the Markov chain starts;
- num_samples: the number S of samples to generate;
- vari: the variance ϵ^2 for the Gaussian proposal distribution q;

and return $[\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(S)}]$ — a list of S samples from $p(\boldsymbol{\theta}) \propto p^*(\boldsymbol{\theta})$. For example:

```
def mh(p_star, param_init, num_samples=5000, vari=1.0):
    # your code here
    return samples
```

Solution. Below is a Python implementation.

```
def mh(p_star, param_init, num_samples=5000, vari=1.0):
  x = []

  x_current = param_init
  for n in range(num_samples):

    # proposal
    x_proposed = multivariate_normal.rvs(mean=x_current, cov=vari)

    # MH step
    a = multivariate_normal.pdf(x_current, mean=x_proposed, cov=vari) *
        p_star(x_proposed)
    a = a / (multivariate_normal.pdf(x_proposed, mean=x_current, cov=
        vari) * p_star(x_current))

    # accept or not
    if a >= 1:
      x_next = np.copy(x_proposed)
    elif uniform.rvs(0, 1) < a:
      x_next = np.copy(x_proposed)
    else:
      x_next = np.copy(x_current)

    # keep record
    x.append(x_next)
    x_current = x_next
```

```
    return x
```

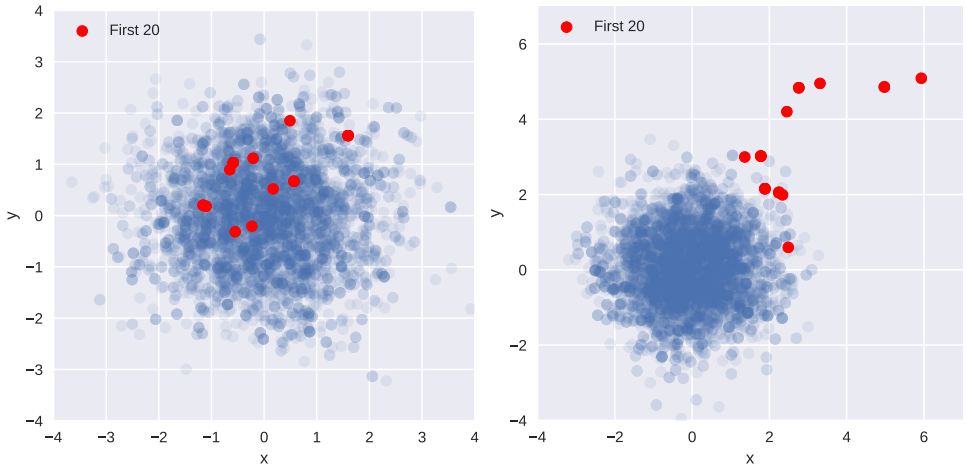
As we are using a symmetrical proposal distribution, $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*) = q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta})$, and one could simplify the algorithm by having $a = \frac{p^*(\boldsymbol{\theta}^*)}{p^*(\boldsymbol{\theta})}$, where $\boldsymbol{\theta}$ is the current sample and $\boldsymbol{\theta}^*$ is the proposed sample.

In practice, it is desirable to implement the function in the log domain, to avoid numerical problems. That is, instead of p^*, `mh` will accept as an argument $\log p^*$, and a will be calculated as:

$$a = (\log q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*) + \log p^*(\boldsymbol{\theta}^*)) - (\log q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}) + \log p^*(\boldsymbol{\theta}))$$

(c) Test your algorithm by sampling $5,000$ samples from $p(x, y) = \mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$. Initialise at $(x = 0, y = 0)$ and use $\epsilon^2 = 1$. Generate a scatter plot of the obtained samples. The plot should be similar to Figure 9.3b. Highlight the first 20 samples only. Do these 20 samples alone adequately approximate the true density?

Sample another $5,000$ points from $p(x, y) = \mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$ using `mh` with $\epsilon^2 = 1$, but this time initialise at $(x = 7, y = 7)$. Generate a scatter plot of the drawn samples and highlight the first 20 samples. If everything went as expected, your plot probably shows a "trail" of samples, starting at $(x = 7, y = 7)$ and slowly approaching the region of space where most of the probability mass is.



(a) Starting the chain at $(0, 0)$.

(b) Starting the chain at $(7, 7)$

Figure 9.4: $5,000$ samples from $\mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$ (blue), with the first 20 samples highlighted (red). Drawn using Metropolis-Hastings with different starting points.

Solution. Figure 9.4 shows the two scatter plots of draws from $\mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$:

- Figure 9.4a highlights the first 20 samples obtained by the chain when starting at $(x = 0, y = 0)$. They appear to be representative samples from the distribution, however, they are not enough to approximate the distribution on their own. This would mean that a sample average computed with 20 samples only would

have high variance, i.e. its value would depend strongly on the values of the 20 samples used to compute the average.

- Figure 9.4b highlights the first 20 samples obtained by the chain when starting at $(x = 7, y = 7)$. One can clearly see the "burn-in" tail which slowly approaches the region where most of the probability mass is.

(d) In practice, we don't know where the distribution we wish to sample from has high density, so we typically initialise the Markov Chain somewhat arbitrarily, or at the maximum a-posterior (MAP) sample if available. The samples obtained in the beginning of the chain are typically discarded, as they are not considered to be representative of the target distribution. This initial period between initialisation and starting to collect samples is called "warm-up", or also "burn-in".

Extended your function `mh` to include an additional warm-up argument W, which specifies the number of MCMC steps taken before starting to collect samples. Your function should still return a list of S samples as in (b).

Solution. We can extend the `mh` function with a warm-up argument by, for example, iterating for `num_samples + warmup` steps, and start recording samples only after the warm-up period:

```
def mh(p_star, param_init, num_samples=5000, vari=1.0, warmup=0):
    x = []
    x_current = param_init
    for n in range(num_samples+warmup):
        ... # body same as before

        if n >= warmup: x.append(x_next)
        x_current = x_next

    return x
```

9.9 Bayesian Poisson regression

Consider a Bayesian Poisson regression model, where outputs y_n are generated from a Poisson distribution of rate $\exp(\alpha x_n + \beta)$, where the x_n are the inputs (covariates), and α and β the parameters of the regression model for which we assume a broad Gaussian prior:

$$\alpha \sim \mathcal{N}(\alpha; 0, 100) \tag{9.28}$$

$$\beta \sim \mathcal{N}(\beta; 0, 100) \tag{9.29}$$

$$y_n \sim \text{Poisson}(y_n; \exp(\alpha x_n + \beta)) \quad \text{for } n = 1, \ldots, N \tag{9.30}$$

$\text{Poisson}(y; \lambda)$ denotes the probability mass function of a Poisson random variable with rate λ,

$$\text{Poisson}(y; \lambda) = \frac{\lambda^y}{y!} \exp(-\lambda), \qquad y \in \{0, 1, 2, \ldots\}, \quad \lambda > 0 \tag{9.31}$$

Consider $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ where $N = 5$ and

$$(x_1, \ldots, x_5) = (-0.50519053, -0.17185719, 0.16147614, 0.49480947, 0.81509851) \quad (9.32)$$
$$(y_1, \ldots, y_5) = (1, 0, 2, 1, 2) \quad (9.33)$$

We are interested in computing the posterior density of the parameters (α, β) given the data $\mathcal{D}$ above.

(a) Derive an expression for the unnormalised posterior density of α and β given $\mathcal{D}$, i.e. a function p^* of the parameters α and β that is proportional to the posterior density $p(\alpha, \beta \mid \mathcal{D})$, and which can thus be used as target density in the Metropolis Hastings algorithm.

Solution. By the product rule, the joint distribution described by the model, with $\mathcal{D}$ plugged in, is proportional to the posterior and hence can be taken as p^*:

$$p^*(\alpha, \beta) = p(\alpha, \beta, \{(x_n, y_n)\}_{n=1}^N) \quad (S.9.75)$$
$$= \mathcal{N}(\alpha; 0, 100)\mathcal{N}(\beta; 0, 100) \prod_{n=1}^N \text{Poisson}(y_n \mid \exp(\alpha x_n + \beta)) \quad (S.9.76)$$

(b) Implement the derived unnormalised posterior density p^*. If your coding environment provides an implementation of the above Poisson pmf, you may use it directly rather than implementing the pmf yourself.

Use the Metropolis Hastings algorithm from Question 9.8(c) to draw $5,000$ samples from the posterior density $p(\alpha, \beta \mid \mathcal{D})$. Set the hyperparameters of the Metropolis-Hastings algorithm to:

- param_init $= (\alpha_{\text{init}}, \beta_{\text{init}}) = (0, 0)$,
- vari $= 1$, and
- number of warm-up steps $W = 1000$.

Plot the drawn samples with x-axis α and y-axis β and report the posterior mean of α and β, as well as their correlation coefficient under the posterior.

Solution. A Python implementation is:

```
import numpy as np
from scipy.stats import multivariate_normal, norm, poisson, uniform

xx1 = np.array([-0.5051905265552105, -0.17185719322187715,
    0.16147614011145617, 0.49480947344478954, 0.8150985069051909])
yy1 = np.array([1, 0, 2, 1, 2])
N1 = len(xx1)

def poisson_regression(params):
    a = params[0]
    b = params[1]
    # mean zero, standard deviation 10 == variance 100
```

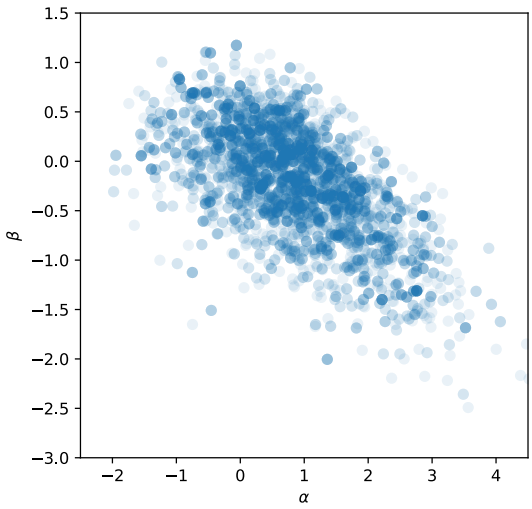


Figure 9.5: Posterior samples for Poisson regression problem; $\boldsymbol{\theta}_{\text{init}} = (0, 0)$.

```
    p = norm.pdf(a, loc=0, scale=10) * norm.pdf(b, loc=0, scale=10)
    for n in range(N1):
        p = p * poisson.pmf(yy1[n], np.exp(a * xx1[n] + b))

    return p

# sample
S = 5000
samples = np.array(mh(poisson_regression, np.array([0, 0]), num_samples
    =S, vari=1.0, warmup=1000))
```

A scatter plot showing $5,000$ samples from the posterior is shown on Figure 9.5. The posterior mean of α is 0.84, the posterior mean of β is -0.2, and posterior correlation coefficient is -0.63. Note that the numerical values are sample-specific.

9.10 Mixing and convergence of Metropolis-Hasting MCMC

Under weak conditions, an MCMC algorithm is an asymptotically exact inference algorithm, meaning that if it is run forever, it will generate samples that correspond to the desired probability distribution. In this case, the chain is said to converge.

In practice, we want to run the algorithm long enough to be able to approximate the posterior adequately. How long is long enough for the chain to converge varies drastically depending on the algorithm, the hyperparameters (e.g. the variance vari), and the target posterior distribution. It is impossible to determine exactly whether the chain has run long enough, but there exist various diagnostics that can help us determine if we can "trust" the sample-based approximation to the posterior.

A very quick and common way of assessing convergence of the Markov chain is to visually inspect the *trace plots* for each parameter. A trace plot shows how the drawn samples evolve

through time, i.e. they are a time-series of the samples generated by the Markov chain. Figure 9.6 shows examples of trace plots obtained by running the Metropolis Hastings algorithm for different values of the hyperparameters vari and param_init. Ideally, the time series covers the whole domain of the target distribution and it is hard to "see" any structure in it so that predicting values of future samples from the current one is difficult. If so, the samples are likely independent from each other and the chain is said to be well "mixed".

(a) Consider the trace plots in Figure 9.6: Is the variance vari used in Figure 9.6b larger or smaller than the value of vari used in Figure 9.6a? Is vari used in Figure 9.6c larger or smaller than the value used in Figure 9.6a?

In both cases, explain the behaviour of the trace plots in terms of the workings of the Metropolis Hastings algorithm and the effect of the variance vari.

Solution. MCMC methods are sensitive to different hyperparameters, and we usually need to carefully diagnose the inference results to ensure that our algorithm adequately approximates the target posterior distribution.

(i) Figure 9.6b uses a *small* variance (vari was set to 0.001) . The trace plots show that the samples for β are very highly correlated and evolve very slowly through time. This is because the introduced randomness is quite small compared to the scale of the posterior, thus the proposed sample at each MCMC iteration will be very close to the current sample and hence likely accepted.

More mathematical explanation: for a symmetric proposal distribution, the acceptance ratio a becomes

$$a = \frac{p^*(\boldsymbol{\theta}^*)}{p^*(\boldsymbol{\theta})}, \tag{S.9.77}$$

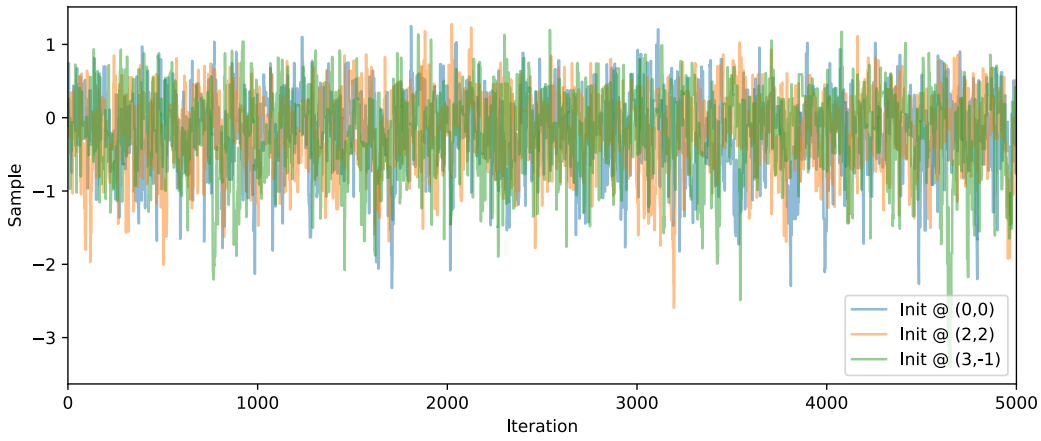
where $\boldsymbol{\theta}$ is the current sample and $\boldsymbol{\theta}^*$ is the proposed sample. For variances that are small compared to the (squared) scale of the posterior, a is close to one and the proposed sample $\boldsymbol{\theta}^*$ gets likely accepted. This then gives rise to the slowly changing time series shown in Figure 9.6b.

(ii) In Figure 9.6c, the variance is larger than the reference (vari was set to 50) . The trace plots suggest that many iterations of the algorithm result in the proposed sample being rejected, and thus we end up copying the same sample over and over again. This is because if the random perturbations are large compared to the scale of the posterior, $p^*(\boldsymbol{\theta}^*)$ may be very different from $p^*(\boldsymbol{\theta})$ and a may be very small.

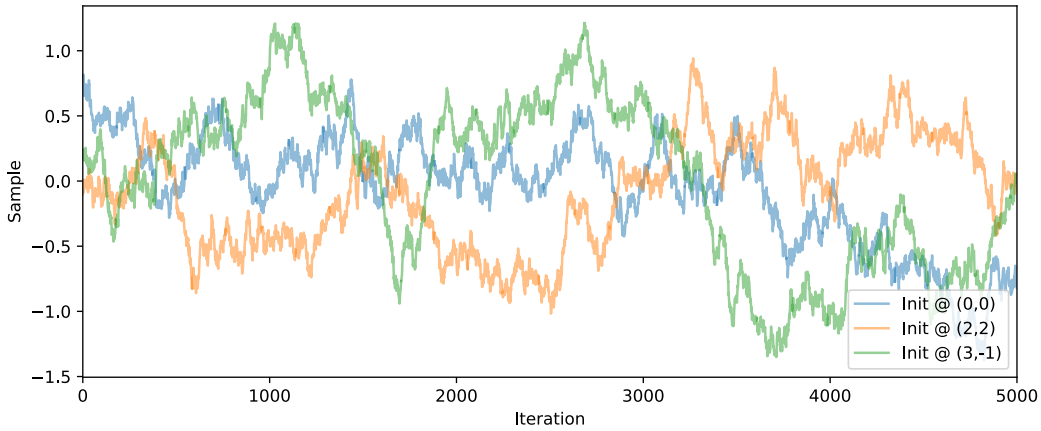
(b) In Metropolis-Hastings, and MCMC in general, any sample depends on the previously generated sample, and hence the algorithm generates samples that are generally statistically dependent. The *effective sample size* of a sequence of dependent samples is the number of independent samples that are, in some sense, equivalent to our number of dependent samples. A definition of the effective sample size (ESS) is

$$\text{ESS} = \frac{S}{1 + 2\sum_{k=1}^{\infty} \rho(k)} \tag{9.34}$$

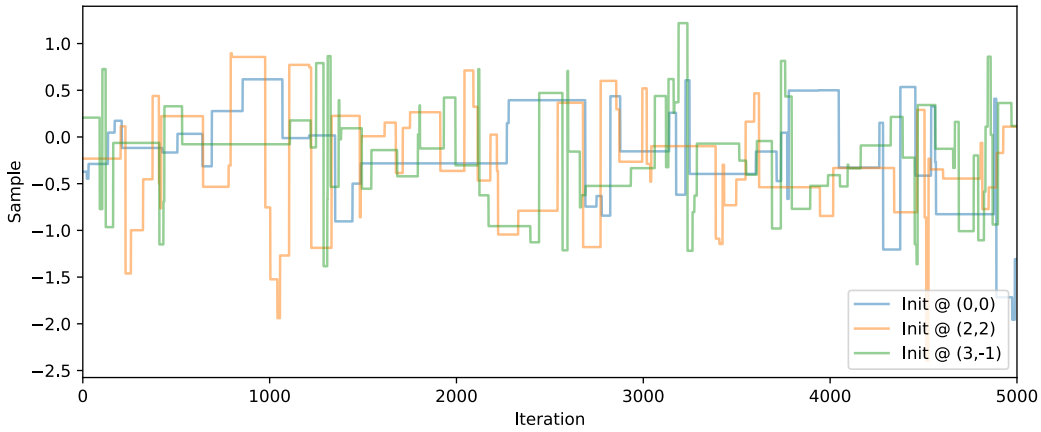
where S is the number of dependent samples drawn and $\rho(k)$ the correlation coefficient between two samples in the Markov chain that are k time points apart. We can



(a) variance vari: 1



(b) Alternative value of vari



(c) Alternative value of vari

Figure 9.6: For Question 9.10(a): Trace plots of the parameter β from Question 9.9 drawn using Metropolis-Hastings with different variances of the proposal distribution.

see that if the samples are strongly correlated, $\sum_{k=1}^{\infty} \rho(k)$ is large and the effective sample size is small. On the other hand, if $\rho(k) = 0$ for all k, the effective sample size is S.

ESS, as defined above, is the number of independent samples which are needed to obtain a sample average that has the same variance as the sample average computed from correlated samples.

To illustrate how correlation between samples is related to a reduction of sample size, consider two pairs of samples (θ_1, θ_2) and (ω_1, ω_2). All variables have variance σ^2 and the same mean μ, but ω_1 and ω_1 are uncorrelated while the covariance matrix for θ_1, θ_2 is $\mathbf{C}$,

$$\mathbf{C} = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \tag{9.35}$$

with $\rho > 0$. The variance of the average $\bar{\omega} = 0.5(\omega_1 + \omega_2)$ is

$$\mathbb{V}(\bar{\omega}) = \frac{\sigma^2}{2}, \tag{9.36}$$

where the 2 in the denominator is the sample size.

Derive an equation for the variance of $\bar{\theta} = 0.5(\theta_1 + \theta_2)$ and compute the reduction α of the sample size when working with the correlated (θ_1, θ_2). In other words, derive an equation of α in

$$\mathbb{V}\left(\bar{\theta}\right) = \frac{\sigma^2}{2/\alpha}. \tag{9.37}$$

What is the effective sample size $2/\alpha$ as $\rho \to 1$?

Solution. Note that $\mathbb{E}(\bar{\theta}) = \mu$. From the definition of variance, we then have

$$\begin{aligned}
\mathbb{V}(\bar{\theta}) &= \mathbb{E}\left((\bar{\theta} - \mu)^2\right) && \text{(S.9.78)} \\
&= \mathbb{E}\left(\left(\frac{1}{2}(\theta_1 + \theta_2) - \mu\right)^2\right) && \text{(S.9.79)} \\
&= \mathbb{E}\left(\left(\frac{1}{2}(\theta_1 - \mu + \theta_2 - \mu\right)^2\right) && \text{(S.9.80)} \\
&= \frac{1}{4}\mathbb{E}\left((\theta_1 - \mu)^2 + (\theta_2 - \mu)^2 + 2(\theta_1 - \mu)(\theta_2 - \mu)\right) && \text{(S.9.81)} \\
&= \frac{1}{4}(\sigma^2 + \sigma^2 + 2\sigma^2\rho) && \text{(S.9.82)} \\
&= \frac{1}{4}(2\sigma^2 + 2\sigma^2\rho) && \text{(S.9.83)} \\
&= \frac{\sigma^2}{2}(1 + \rho) && \text{(S.9.84)} \\
&= \frac{\sigma^2}{2/(1 + \rho)} && \text{(S.9.85)}
\end{aligned}$$

Hence: $\alpha = (1 + \rho)$, and for $\rho \to 1$, $2/\alpha \to 1$.

Because of the strong correlation, we effectively only have one sample and not two if $\rho \to 1$.

Chapter 10

Variational Inference

Exercises

10.1 Mean field variational inference I

Let $\mathcal{L}_{\mathbf{x}}(q)$ be the evidence lower bound for the marginal $p(\mathbf{x})$ of a joint pdf/pmf $p(\mathbf{x}, \mathbf{y})$,

$$\mathcal{L}_{\mathbf{x}}(q) = \mathbb{E}_{q(\mathbf{y}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}, \mathbf{y})}{q(\mathbf{y}|\mathbf{x})}\right]. \tag{10.1}$$

Mean field variational inference assumes that the variational distribution $q(\mathbf{y}|\mathbf{x})$ fully factorises, i.e.

$$q(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{d} q_i(y_i|\mathbf{x}), \tag{10.2}$$

when $\mathbf{y}$ is d-dimensional. An approach to learning the q_i for each dimension is to update one at a time while keeping the others fixed. We here derive the corresponding update equations.

(a) Show that the evidence lower bound $\mathcal{L}_{\mathbf{x}}(q)$ can be written as

$$\mathcal{L}_{\mathbf{x}}(q) = \mathbb{E}_{q_1(y_1|\mathbf{x})}\mathbb{E}_{q(\mathbf{y}_{\setminus 1}|\mathbf{x})}\left[\log p(\mathbf{x}, \mathbf{y})\right] - \sum_{i=1}^{d} \mathbb{E}_{q_i(y_i|\mathbf{x})}\left[\log q_i(y_i|\mathbf{x})\right] \tag{10.3}$$

where $q(\mathbf{y}_{\setminus 1}|\mathbf{x}) = \prod_{i=2}^{d} q_i(y_i|\mathbf{x})$ is the variational distribution without $q_1(y_1|\mathbf{x})$.

Solution. This follows directly from the definition of the ELBO and the assumed factorisation of $q(\mathbf{y}|\mathbf{x})$. We have

$$\begin{aligned}
\mathcal{L}_{\mathbf{x}}(q) &= \mathbb{E}_{q(\mathbf{y}|\mathbf{x})} \log p(\mathbf{x}, \mathbf{y}) - \mathbb{E}_{q(\mathbf{y}|\mathbf{x})} \log q(\mathbf{y}|\mathbf{x}) && \text{(S.10.1)}\\
&= \mathbb{E}_{\prod_{i=1}^{d} q_i(y_i|\mathbf{x})} \log p(\mathbf{x}, \mathbf{y}) - \mathbb{E}_{\prod_{i=1}^{d} q_i(y_i|\mathbf{x})} \sum_{i=1}^{d} \log q_i(y_i|\mathbf{x}) && \text{(S.10.2)}\\
&= \mathbb{E}_{\prod_{i=1}^{d} q_i(y_i|\mathbf{x})} \log p(\mathbf{x}, \mathbf{y}) - \sum_{i=1}^{d} \mathbb{E}_{q_i(y_i|\mathbf{x})} \log q_i(y_i|\mathbf{x}) && \text{(S.10.3)}\\
&= \mathbb{E}_{q_1(y_1|\mathbf{x})}\mathbb{E}_{\prod_{i=2}^{d} q_i(y_i|\mathbf{x})} \log p(\mathbf{x}, \mathbf{y}) - \sum_{i=1}^{d} \mathbb{E}_{q_i(y_i|\mathbf{x})} \log q_i(y_i|\mathbf{x}) && \text{(S.10.4)}\\
&= \mathbb{E}_{q_1(y_1|\mathbf{x})}\mathbb{E}_{q(\mathbf{y}_{\setminus 1}|\mathbf{x})}\left[\log p(\mathbf{x}, \mathbf{y})\right] - \sum_{i=1}^{d} \mathbb{E}_{q_i(y_i|\mathbf{x})}\left[\log q_i(y_i|\mathbf{x})\right] && \text{(S.10.5)}
\end{aligned}$$

We have here used the linearity of expectation. In case of continuous random variables, for instance, we have

$$\mathbb{E}_{\prod_{i=1}^{d} q_i(y_i|\mathbf{x})} \sum_{i=1}^{d} \log q_i(y_i|\mathbf{x}) = \int q_1(y_1|\mathbf{x}) \cdot \ldots \cdot q_d(y_d|\mathbf{x}) \sum_{i=1}^{d} \log q_i(y_i|\mathbf{x}) dy_1 \ldots dy_d \tag{S.10.6}$$

$$= \sum_{i=1}^{d} \int q_1(y_1|\mathbf{x}) \cdot \ldots \cdot q_d(y_d|\mathbf{x}) \log q_i(y_i|\mathbf{x}) dy_1 \ldots dy_d \tag{S.10.7}$$

$$= \sum_{i=1}^{d} \int q_i(y_i|\mathbf{x}) \log q_i(y_i|\mathbf{x}) dy_i \underbrace{\int \prod_{j \neq i} q_j(y_j|\mathbf{x}) dy_j}_{=1} \tag{S.10.8}$$

$$= \sum_{i=1}^{d} E_{q_i(y_i|\mathbf{x})} \log q_i(y_i|\mathbf{x}) \tag{S.10.9}$$

For discrete random variables, the integral is replaced with a sum and leads to the same result.

(b) Assume that we would like to update $q_1(y_1|\mathbf{x})$ and that the variational marginals of the other dimensions are kept fixed. Show that

$$\underset{q_1(y_1|\mathbf{x})}{\operatorname{argmax}} \mathcal{L}_{\mathbf{x}}(q) = \underset{q_1(y_1|\mathbf{x})}{\operatorname{argmin}} \mathrm{KL}(q_1(y_1|\mathbf{x})||\bar{p}(y_1|\mathbf{x})) \tag{10.4}$$

with

$$\log \bar{p}(y_1|\mathbf{x}) = \mathbb{E}_{q(\mathbf{y}_{\setminus 1}|\mathbf{x})} \left[\log p(\mathbf{x}, \mathbf{y})\right] + \text{const}, \tag{10.5}$$

where const refers to terms not depending on y_1. That is,

$$\bar{p}(y_1|\mathbf{x}) = \frac{1}{Z} \exp\left[\mathbb{E}_{q(\mathbf{y}_{\setminus 1}|\mathbf{x})} \left[\log p(\mathbf{x}, \mathbf{y})\right]\right], \tag{10.6}$$

where Z is the normalising constant. Note that variables $y_2, \ldots, y_d$ are marginalised out due to the expectation with respect to $q(\mathbf{y}_{\setminus 1}|\mathbf{x})$.

Solution. Starting from

$$\mathcal{L}_{\mathbf{x}}(q) = \mathbb{E}_{q_1(y_1|\mathbf{x})} \mathbb{E}_{q(\mathbf{y}_{\setminus 1}|\mathbf{x})} \left[\log p(\mathbf{x}, \mathbf{y})\right] - \sum_{i=1}^{d} \mathbb{E}_{q_i(y_i|\mathbf{x})} \left[\log q_i(y_i|\mathbf{x})\right] \tag{S.10.10}$$

we drop terms that do not depend on q_1. We then obtain

$$J(q_1) = \mathbb{E}_{q_1(y_1|\mathbf{x})} \mathbb{E}_{q(\mathbf{y}_{\setminus 1}|\mathbf{x})} \left[\log p(\mathbf{x}, \mathbf{y})\right] - \mathbb{E}_{q_1(y_1|\mathbf{x})} \left[\log q_1(y_1|\mathbf{x})\right] \tag{S.10.11}$$

$$= \mathbb{E}_{q_1(y_1|\mathbf{x})} \log \bar{p}(y_1|\mathbf{x}) - \mathbb{E}_{q_1(y_1|\mathbf{x})} \left[\log q_1(y_1|\mathbf{x})\right] + \text{const} \tag{S.10.12}$$

$$= \mathbb{E}_{q_1(y_1|\mathbf{x})} \left[\log \frac{\bar{p}(y_1|\mathbf{x})}{q_1(y_1|\mathbf{x})}\right] \tag{S.10.13}$$

$$= -\mathrm{KL}(q_1(y_1|\mathbf{x})||\bar{p}(y_1|\mathbf{x})) \tag{S.10.14}$$

Hence

$$\underset{q_1(y_1|\mathbf{x})}{\operatorname{argmax}} \mathcal{L}_{\mathbf{x}}(q) = \underset{q_1(y_1|\mathbf{x})}{\operatorname{argmin}} \mathrm{KL}(q_1(y_1|\mathbf{x})||\bar{p}(y_1|\mathbf{x})) \tag{S.10.15}$$

(c) Conclude that given $q_i(y_i|\mathbf{x})$, $i = 2, \ldots, d$, the optimal $q_1(y_1|\mathbf{x})$ equals $\bar{p}(y_1|\mathbf{x})$.

This then leads to an iterative updating scheme where we cycle through the different dimensions, each time updating the corresponding marginal variational distribution according to:

$$q_i(y_i|\mathbf{x}) = \bar{p}(y_i|\mathbf{x}), \qquad \bar{p}(y_i|\mathbf{x}) = \frac{1}{Z} \exp\left[\mathbb{E}_{q(\mathbf{y}_{\setminus i}|\mathbf{x})}\left[\log p(\mathbf{x}, \mathbf{y})\right]\right] \tag{10.7}$$

where $q(\mathbf{y}_{\setminus i}|\mathbf{x}) = \prod_{j \neq i} q(y_j|\mathbf{x})$ is the product of all marginals without marginal $q_i(y_i|\mathbf{x})$.

Solution. This follows immediately from the fact that the KL divergence is minimised when $q_1(y_1|\mathbf{x}) = \bar{p}(y_1|\mathbf{x})$. Side-note: The iterative update rule can be considered to be coordinate ascent optimisation in function space, where each "coordinate" corresponds to a $q_i(y_i|\mathbf{x})$.

10.2 Mean field variational inference II

Assume random variables y_1, y_2, x are generated according to the following process

$$y_1 \sim \mathcal{N}(y_1; 0, 1) \qquad y_2 \sim \mathcal{N}(y_2; 0, 1) \tag{10.8}$$

$$n \sim \mathcal{N}(n; 0, 1) \qquad x = y_1 + y_2 + n \tag{10.9}$$

where y_1, y_2, n are statistically independent.

(a) y_1, y_2, x are jointly Gaussian. Determine their mean and their covariance matrix.

Solution. The expected value of y_1 and y_2 is zero. By linearity of expectation, the expected value of x is

$$\mathbb{E}(x) = \mathbb{E}(y_1) + \mathbb{E}(y_2) + \mathbb{E}(n) = 0 \tag{S.10.16}$$

The variance of y_1 and y_2 is 1. Since y_1, y_2, n are statistically independent,

$$\mathbb{V}(x) = \mathbb{V}(y_1) + \mathbb{V}(y_2) + \mathbb{V}(n) = 1 + 1 + 1 = 3. \tag{S.10.17}$$

The covariance between y_1 and x is

$$\begin{aligned}
\mathrm{cov}(y_1, x) &= \mathbb{E}((y_1 - \mathbb{E}(y_1))(x - \mathbb{E}(x))) = \mathbb{E}(y_1 x) && \text{(S.10.18)}\\
&= \mathbb{E}(y_1(y_1 + y_2 + n)) = \mathbb{E}(y_1^2) + \mathbb{E}(y_1 y_2) + \mathbb{E}(y_1 n) && \text{(S.10.19)}\\
&= 1 + \mathbb{E}(y_1)\mathbb{E}(y_2) + \mathbb{E}(y_1)\mathbb{E}(n) && \text{(S.10.20)}\\
&= 1 + 0 + 0 && \text{(S.10.21)}
\end{aligned}$$

where we have used that y_1 and x have zero mean and the independence assumptions.

The covariance between y_2 and x is computed in the same way and equals 1 too.

We thus obtain the covariance matrix $\mathbf{\Sigma}$,

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 3 \end{pmatrix} \tag{S.10.22}$$

(b) The conditional $p(y_1, y_2|x)$ is Gaussian with mean $\mathbf{m}$ and covariance $\mathbf{C}$,

$$\mathbf{m} = \frac{x}{3}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \mathbf{C} = \frac{1}{3}\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \tag{10.10}$$

Since x is the sum of three random variables that have the same distribution, it makes intuitive sense that the mean assigns $1/3$ of the observed value of x to y_1 and y_2. Moreover, y_1 and y_2 are negatively corrected since an increase in y_1 must be compensated with a decrease in y_2.

Let us now approximate the posterior $p(y_1, y_2|x)$ with mean field variational inference. Determine the optimal variational distribution using the method and results from Exercise 10.1. You may use that

$$p(y_1, y_2, x) = \mathcal{N}\left((y_1, y_2, x); \mathbf{0}, \mathbf{\Sigma}\right) \quad \mathbf{\Sigma} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix} \quad \mathbf{\Sigma}^{-1} = \begin{pmatrix} 2 & 1 & -1 \\ 1 & 2 & -1 \\ -1 & -1 & 1 \end{pmatrix} \tag{10.11}$$

Solution. The mean field assumption means that the variational distribution is assumed to factorise as

$$q(y_1, y_2|x) = q_1(y_1|x)q_2(y_2|x) \tag{S.10.23}$$

From Exercise 10.1, the optimal $q_1(y_1|x)$ and $q_2(y_2|x)$ satisfy

$$q_1(y_1|x) = \bar{p}(y_1|x), \qquad \bar{p}(y_1|x) = \frac{1}{Z}\exp\left[\mathbb{E}_{q_2(y_2|x)}\left[\log p(y_1, y_2, x)\right]\right] \tag{S.10.24}$$

$$q_2(y_2|x) = \bar{p}(y_2|x), \qquad \bar{p}(y_2|x) = \frac{1}{Z}\exp\left[\mathbb{E}_{q_1(y_1|x)}\left[\log p(y_1, y_2, x)\right]\right] \tag{S.10.25}$$

Note that these are coupled equations: q_2 features in the equation for q_1 via $\bar{p}(y_1|x)$, and q_1 features in the equation for q_2 via $\bar{p}(y_2|x)$. But we have two equations for two unknowns, which for the Gaussian joint model $p(x, y_1, y_2)$ can be solved in closed form.

Given the provided equation for $p(y_1, y_2, x)$, we have that

$$\log p(y_1, y_2, x) = -\frac{1}{2}\begin{pmatrix} y_1 \\ y_2 \\ x \end{pmatrix}^\top \begin{pmatrix} 2 & 1 & -1 \\ 1 & 2 & -1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ x \end{pmatrix} + \text{const} \tag{S.10.26}$$

$$= -\frac{1}{2}\left(2y_1^2 + 2y_2^2 + x^2 + 2y_1y_2 - 2y_1x - 2y_2x\right) + \text{const} \tag{S.10.27}$$

Let us start with the equation for $\bar{p}(y_1|x)$. It is easier to work in the logarithmic domain, where we obtain:

$$\log \bar{p}(y_1|x) = \mathbb{E}_{q_2(y_2|x)}\left[\log p(y_1, y_2, x)\right] + \text{const} \tag{S.10.28}$$

$$= -\frac{1}{2}\mathbb{E}_{q_2(y_2|x)}\left[2y_1^2 + 2y_2^2 + x^2 + 2y_1y_2 - 2y_1x - 2y_2x\right] + \text{const} \tag{S.10.29}$$

$$= -\frac{1}{2}\left(2y_1^2 + 2y_1\mathbb{E}_{q_2(y_2|x)}[y_2] - 2y_1x\right) + \text{const} \tag{S.10.30}$$

$$= -\frac{1}{2}\left(2y_1^2 + 2y_1m_2 - 2y_1x\right) + \text{const} \tag{S.10.31}$$

$$= -\frac{1}{2}\left(2y_1^2 - 2y_1(x - m_2)\right) + \text{const} \tag{S.10.32}$$

where we have absorbed all terms not involving y_1 into the constant. Moreover, we set $\mathbb{E}_{q_2(y_2|x)}[y_2] = m_2$.

Note that an arbitrary Gaussian density $\mathcal{N}(y; m, \sigma^2)$ with mean m and variance σ^2 can be written in the log-domain as

$$\log \mathcal{N}(y; m, \sigma^2) = -\frac{1}{2}\frac{(y-m)^2}{\sigma^2} + \text{const} \tag{S.10.33}$$

$$= -\frac{1}{2}\left(\frac{y^2}{\sigma^2} - 2y\frac{m}{\sigma^2}\right) + \text{const} \tag{S.10.34}$$

Comparison with (S.10.32) shows that $\bar{p}(y_1|x)$, and hence $q_1(y_1|x)$, is Gaussian with variance and mean equal to

$$\sigma_1^2 = \frac{1}{2} \qquad\qquad m_1 = \frac{1}{2}(x - m_2) \tag{S.10.35}$$

Note that we have not made a Gaussianity assumption on $q_1(y_1|x)$. The optimal $q_1(y_1|x)$ turns out to be Gaussian because the model $p(y_1, y_2, x)$ is Gaussian.

The equation for $\bar{p}(y_2|x)$ gives similarly

$$\log \bar{p}(y_2|x) = \mathbb{E}_{q_1(y_1|x)}\left[\log p(y_1, y_2, x)\right] + \text{const} \tag{S.10.36}$$

$$= -\frac{1}{2}\mathbb{E}_{q_1(y_1|x)}\left[2y_1^2 + 2y_2^2 + x^2 + 2y_1y_2 - 2y_1x - 2y_2x\right] + \text{const} \tag{S.10.37}$$

$$= -\frac{1}{2}\left(2y_2^2 + 2\mathbb{E}_{q_1(y_1|x)}[y_1]y_2 - 2y_2x\right) + \text{const} \tag{S.10.38}$$

$$= -\frac{1}{2}\left(2y_2^2 + 2m_1y_2 - 2y_2x\right) + \text{const} \tag{S.10.39}$$

$$= -\frac{1}{2}\left(2y_2^2 - 2y_2(x - m_1)\right) + \text{const} \tag{S.10.40}$$

where we have absorbed all terms not involving y_2 into the constant. Moreover, we set $\mathbb{E}_{q_1(y_1|x)}[y_1] = m_1$. With (S.10.34), this is defines a Gaussian distribution with variance and mean equal to

$$\sigma_2^2 = \frac{1}{2} \qquad\qquad m_2 = \frac{1}{2}(x - m_1) \tag{S.10.41}$$

Hence the optimal marginal variational distributions $q_1(y_1|x)$ and $q_2(y_2|x)$ are both Gaussian with variance equal to 1/2. Their means satisfy

$$m_1 = \frac{1}{2}(x - m_2) \qquad\qquad m_2 = \frac{1}{2}(x - m_1) \tag{S.10.42}$$

These are two equations for two unknowns. We can solve them as follows

$$\begin{aligned} 2m_1 &= x - m_2 && \text{(S.10.43)}\\ &= x - \frac{1}{2}(x - m_1) && \text{(S.10.44)}\\ 4m_1 &= 2x - x + m_1 && \text{(S.10.45)}\\ 3m_1 &= x && \text{(S.10.46)}\\ m_1 &= \frac{1}{3}x && \text{(S.10.47)} \end{aligned}$$

Hence

$$m_2 = \frac{1}{2}x - \frac{1}{6}x = \frac{2}{6}x = \frac{1}{3}x \tag{S.10.48}$$

In summary, we find

$$q_1(y_1|x) = \mathcal{N}\left(y_1; \frac{x}{3}, \frac{1}{2}\right) \qquad\qquad q_2(y_2|x) = \mathcal{N}\left(y_2; \frac{x}{3}, \frac{1}{2}\right) \tag{S.10.49}$$

and the optimal variational distribution $q(y_1, y_2|x) = q_1(y_1|x)q_2(y_2|x)$ is Gaussian. We have made the mean field (independence) assumption but not the Gaussianity assumption. Gaussianity of the variational distribution is a consequence of the Gaussianity of the model $p(y_1, y_2, x)$.

Comparison with the true posterior shows that the mean field variational distribution $q(y_1, y_2|x)$ has the same mean but ignores the correlation and underestimates the marginal variances. The true posterior and the mean field approximation are shown in Figure 10.1.

10.3 Variational posterior approximation I

We have seen that maximising the evidence lower bound (ELBO) with respect to the variational distribution q minimises the Kullback-Leibler divergence to the true posterior p. We here assume that q and p are probability density functions so that the Kullback-Leibler divergence between them is defined as

$$\mathrm{KL}(q||p) = \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \mathrm{d}\mathbf{x} = \mathbb{E}_q \left[\log \frac{q(\mathbf{x})}{p(\mathbf{x})}\right]. \tag{10.12}$$

(a) You can here assume that $\mathbf{x}$ is one-dimensional so that p and q are univariate densities. Consider the case where p is a bimodal density but the variational densities q are unimodal. Sketch a figure that shows p and a variational distribution q that has been learned by minimising $\mathrm{KL}(q||p)$. Explain qualitatively why the sketched q minimises $\mathrm{KL}(q||p)$.

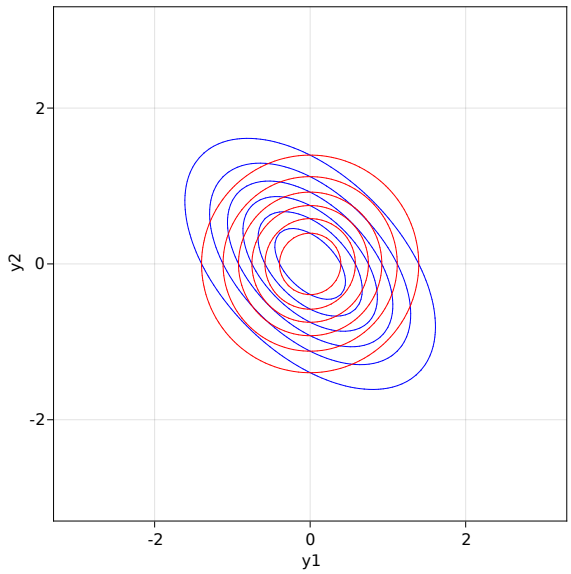
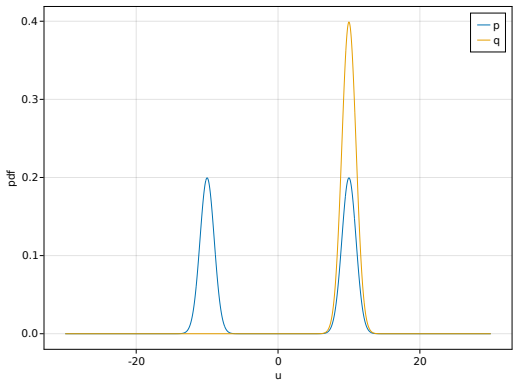


Figure 10.1: In blue: correlated true posterior. In red: mean field approximation.



Solution. A possible sketch is shown in the figure below.

Explanation: We can divide the domain of p and q into the areas where p is small (zero) and those where p has significant mass. Since the objective features q in the numerator while p is in the denominator, an optimal q needs to be zero where p is zero. Otherwise, it would incur a large penalty (division by zero). Since we take the expectation with respect to q, however, regions where $p > 0$ do not need to be covered by q; cutting them out does not incur a penalty. Hence, optimal unimodal q only cover one peak of the bimodal p.

(b) Assume that the true posterior $p(\mathbf{x}) = p(x_1, x_2)$ factorises into two Gaussians of mean zero and variances σ_1^2 and σ_2^2,

$$p(x_1, x_2) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left[-\frac{x_1^2}{2\sigma_1^2}\right] \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left[-\frac{x_2^2}{2\sigma_2^2}\right]. \tag{10.13}$$

Assume further that the variational density $q(x_1, x_2; \lambda^2)$ is parametrised as

$$q(x_1, x_2; \lambda^2) = \frac{1}{2\pi\lambda^2} \exp\left[-\frac{x_1^2 + x_2^2}{2\lambda^2}\right] \tag{10.14}$$

where λ^2 is the variational parameter that is learned by minimising $\mathrm{KL}(q||p)$. If σ_2^2 is much larger than σ_1^2, do you expect λ^2 to be closer to σ_2^2 or to σ_1^2? Provide an explanation.

Solution. The learned variational parameter will be closer to σ_1^2 (the smaller of the two σ_i^2).

Explanation: First note that the σ_i^2 are the variances along the two different axes, and that λ^2 is the single variance for both x_1 and x_2. The objective penalises q if it is non-zero where p is zero (see above). The variational parameter λ^2 thus will get adjusted during learning so that the variance of q is close to the smallest of the two σ_i^2.

10.4 Variational posterior approximation II

We have seen that maximising the evidence lower bound (ELBO) with respect to the variational distribution minimises the Kullback-Leibler divergence to the true posterior. We here investigate the nature of the approximation if the family of variational distributions does not include the true posterior.

(a) Assume that the true posterior for $\mathbf{x} = (x_1, x_2)$ is given by

$$p(\mathbf{x}) = \mathcal{N}(x_1; \sigma_1^2)\mathcal{N}(x_2; \sigma_2^2) \tag{10.15}$$

and that our variational distribution $q(\mathbf{x}; \lambda^2)$ is

$$q(\mathbf{x}; \lambda^2) = \mathcal{N}(x_1; \lambda^2)\mathcal{N}(x_2; \lambda^2), \tag{10.16}$$

where $\lambda > 0$ is the variational parameter. Provide an equation for

$$J(\lambda) = \mathrm{KL}(q(\mathbf{x}; \lambda^2)||p(\mathbf{x})), \tag{10.17}$$

where you can omit additive terms that do not depend on λ.

Solution. We write

$$\begin{aligned}
\mathrm{KL}(q(\mathbf{x}; \lambda^2)||p(\mathbf{x})) &= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}; \lambda^2)}{p(\mathbf{x})}\right] && \text{(S.10.50)}\\
&= \mathbb{E}_q \log q(\mathbf{x}; \lambda^2) - \mathbb{E}_q \log p(\mathbf{x}) && \text{(S.10.51)}\\
&= \mathbb{E}_q \log \mathcal{N}(x_1; \lambda^2) + \mathbb{E}_q \log \mathcal{N}(x_2; \lambda^2)\\
&\quad - \mathbb{E}_q \log \mathcal{N}(x_1; \sigma_1^2) - \mathbb{E}_q \log \mathcal{N}(x_2; \sigma_2^2) && \text{(S.10.52)}
\end{aligned}$$

We further have

$$\mathbb{E}_q \log \mathcal{N}(x_i; \lambda^2) = \mathbb{E}_q \log \left[\frac{1}{\sqrt{2\pi\lambda^2}} \exp \left[-\frac{x_i^2}{2\lambda^2} \right] \right] \tag{S.10.53}$$

$$= \log \left[\frac{1}{\sqrt{2\pi\lambda^2}} \right] - \mathbb{E}_q \left[\frac{x_i^2}{2\lambda^2} \right] \tag{S.10.54}$$

$$= -\log \lambda - \frac{\lambda^2}{2\lambda^2} + \text{const} \tag{S.10.55}$$

$$= -\log \lambda - \frac{1}{2} + \text{const} \tag{S.10.56}$$

$$= -\log \lambda + \text{const} \tag{S.10.57}$$

where we have used that for zero mean x_i, $\mathbb{E}_q[x_i^2] = \mathbb{V}(x_i) = \lambda^2$.

We similarly obtain

$$\mathbb{E}_q \log \mathcal{N}(x_i; \sigma_i^2) = \mathbb{E}_q \log \left[\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left[-\frac{x_i^2}{2\sigma_i^2} \right] \right] \tag{S.10.58}$$

$$= -\log \left[\frac{1}{\sqrt{2\pi\sigma_i^2}} \right] - \mathbb{E}_q \left[\frac{x_i^2}{2\sigma_i^2} \right] \tag{S.10.59}$$

$$= -\log \sigma_i - \frac{\lambda^2}{2\sigma_i^2} + \text{const} \tag{S.10.60}$$

$$= -\frac{\lambda^2}{2\sigma_i^2} + \text{const} \tag{S.10.61}$$

We thus have

$$\text{KL}(q(\mathbf{x}; \lambda^2) || p(\mathbf{x})) = -2 \log \lambda + \lambda^2 \left(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2} \right) + \text{const} \tag{S.10.62}$$

(b) Determine the value of λ that minimises $J(\lambda) = \text{KL}(q(\mathbf{x}; \lambda^2) || p(\mathbf{x}))$. Interpret the result and relate it to properties of the Kullback-Leibler divergence.

Solution. Taking derivatives of $J(\lambda)$ with respect to λ gives

$$\frac{\partial J(\lambda)}{\partial \lambda} = -\frac{2}{\lambda} + \lambda \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) \tag{S.10.63}$$

Setting it zero yields

$$\frac{1}{\lambda^2} = \frac{1}{2} \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) \tag{S.10.64}$$

so that

$$\lambda^2 = 2 \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{S.10.65}$$

or

$$\lambda = \sqrt{2}\sqrt{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \tag{S.10.66}$$

This is a minimum because the second derivative of $J(\lambda)$

$$\frac{\partial^2 J(\lambda)}{\partial \lambda^2} = \frac{2}{\lambda^2} + \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right) \tag{S.10.67}$$

is positive for all $\lambda > 0$.

The result has an intuitive explanation: the optimal variance λ^2 is the harmonic mean of the variances σ_i^2 of the true posterior. In other words, the optimal precision $1/\lambda^2$ is given by the average of the precisions $1/\sigma_i^2$ of the two dimensions.

If the variances are not equal, e.g. if $\sigma_2^2 > \sigma_1^2$, we see that the optimal variance of the variational distribution strikes a compromise between two types of penalties in the KL-divergence: the penalty of having a bad fit because the variational distribution along dimension two is too narrow; and along dimension one, the penalty for the variational distribution to be nonzero when p is small.

Bibliography

David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012. URL http://www.cs.ucl.ac.uk/staff/d.barber/brml/.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. URL https://link.springer.com/book/9780387310732.

Nicolas Chopin and Omiros Papaspiliopoulos. *An introduction to Sequential Monte Carlo*. Springer, 2020. URL https://link.springer.com/book/10.1007/978-3-030-47845-2.

Brendan J. Frey. Extending factor graphs so as to unify directed and undirected graphical models. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2003. URL https://arxiv.org/abs/1212.2486.

Mohinder S. Grewal and Angus P. Andrews. Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, 30(3): 69–78, 2010. URL https://ieeexplore.ieee.org/document/5466132.

Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999. URL https://ieeexplore.ieee.org/document/761722.

Aapo Hyvärinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005. URL http://jmlr.org/papers/volume6/hyvarinen05a/hyvarinen05a.pdf.

Aapo Hyvärinen, Erkki Oja, and Juha Karhunen. *Independent Component Analysis*. John Wiley & Sons, 2001. URL https://www.cs.helsinki.fi/u/ahyvarin/papers/bookfinal_ICA.pdf.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999.

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013. URL https://artowen.su.domains/mc/.

Christian Robert and George Casella. *Introducing Monte Carlo Methods with R*. Springer, 2010. URL https://link.springer.com/book/10.1007/978-1-4419-1576-4.

Walter Rudin. *Principles of Mathematical Analysis*. McGraw Hill, 3rd edition edition, 1976.