Probabilistic Machine Learning

Joachim Giesen

Contents

Introduction 5

I Models 13

1 Multivariate Categoricals 15

2 Pairwise Categoricals 23

3 Markov Random Fields 27

4 Bayesian Networks 35

5 Exercises 39

II Model Selection 41

6 Maximum Likelihood and MAP 43

7 Pairwise Categoricals 47

8 Iterative Proportional Scaling 59

9 Maximum Entropy Principle 67

10 Bayesian Model Selection 87

11 Exercises 97

III Inference **101**

12 Complexity of Inference Queries **103**

13 Knowledge Compilation **111**

14 Variational Inference **123**

15 Gibbs Sampling **131**

16 Exercises **137**

Introduction

In this lecture we consider multivariate probability distributions. Let us consider a simple example that models three customers, namely, *Alice*, *Bob* and *Claire*, of an online shop. The shop is planning another marketing campaign and wants to find out how likely Alice, Bob and Claire are to respond positively to the campaign. We can model the response of $A(lice), B(ob)$ and $C(laire)$ as a vector (A, B, C) of three random variables that each can take the values 0 or 1, where 1 means that they respond positively and 0 means that they do not respond positively to the campaign. The probability distribution of the random vector is specified by the probabilities for the possible outcomes. Here, we use a simple estimate for the probabilities that is based on the outcomes of 96 previous marketing campaigns. The probability distribution is summarized in the following probability table

A	1	1	1	1	0	0	0	0
B	1	1	0	0	1	1	0	0
C	1	0	1	0	1	0	1	0
$p(A, B, C)$	$\frac{21}{96}$	$\frac{8}{96}$	$\frac{7}{96}$	$\frac{8}{96}$	$\frac{3}{96}$	$\frac{24}{96}$	$\frac{1}{96}$	$\frac{24}{96}$

on the sample space $\{0, 1\}^3 = \{0, 1\} \times \{0, 1\} \times \{0, 1\}$.

Inference Queries

Given the joint probability function

$$p : \{0, 1\}^3 \to [0, 1]$$

that is encoded in the probability table we can ask *inference questions* like "*Given that Claire responded positively to the campaign, what is the chance that the response of Alice is also positive?*". A related, slighly more formal query is

> *What is the more probable response of A provided that we have observed $C = 1$?*

In mathematical terms the question can be phrased as

$$\text{argmax}_{a\in\{0,1\}}\ p(A = a \mid C = 1).$$

For answering the question, note that it does not depend on Bob's response. Hence, we can disregard Bob by *marginalizing* the variable B out of the probability function, that is, computing a probability table for only (A, C) by summing over the probabilities where $B = 0$ and $B = 1$. This gives,

	A	1	1	0	0
	C	1	0	1	0
	$p(A, B = 1, C)$	$\frac{7}{32}$	$\frac{1}{12}$	$\frac{1}{32}$	$\frac{1}{4}$
$+$	$p(A, B = 0, C)$	$\frac{7}{96}$	$\frac{1}{12}$	$\frac{1}{96}$	$\frac{1}{4}$
$=$	$p(A, C)$	$\frac{7}{24}$	$\frac{4}{24}$	$\frac{1}{24}$	$\frac{12}{24}$

Next we have to consider the *condition* that $C = 1$. This information is often called *evidence* in probabilistic inference problems. From the table above we can read off

$$p(A = 1, C = 1) = \frac{7}{24} \quad \text{and} \quad p(A = 0, C = 1) = \frac{1}{24},$$

which means that it is more likely that Alice will also react positively to the campaign. However, we cannot interpret $p(A = 1, C = 1)$ (or $p(A = 0, C = 1)$, respectively) as the probability to observe $A = 1$ (or $A = 0$) under the condition that we have observed $C = 1$, because by the law of total probability the conditional probabilities also should sum up to 1. In order to turn $p(A = 1, C = 1)$ and $p(A = 0, C = 0)$ into proper conditional probabilities, denoted as $p(A \mid C)$, we have to normalize them and get

$$\begin{aligned} p(A = 1 \mid C = 1) &= \frac{p(A = 1, C = 1)}{p(A = 1, C = 1) + p(A = 0, C = 1)} \\ &= \frac{p(A = 1, C = 1)}{p(C = 1)} \end{aligned}$$

and

$$p(A = 0 \mid C = 1) = \frac{p(A = 0, C = 1)}{p(C = 1)}.$$

In our example we get $p(C = 1) = \frac{8}{24}$ and thus

$A \mid C = 1$	1	0
$p(A \mid C = 1)$	$\frac{7}{8}$	$\frac{1}{8}$

That is, provided that we have observed $C = 1$, the more probable of the two possible outcomes is $A = 1$. The conditional probabilities satisfy *Bayes' theorem*

$$p(A \mid C) = \frac{p(A, C)}{p(C)} = \frac{p(C \mid A)p(A)}{p(C)}$$

that in the Bayesian interpretation of probability tells how we should update our *belief* given the *evidence*. Prior to observing the evidence, our belief of observing $A = 1$ is the marginal probability

$$p(A = 1) = p(A = 1, C = 1) + p(A = 1, C = 0) = \frac{7}{24} + \frac{4}{24} = \frac{11}{24}$$

that increases to $p(A = 1 \mid C = 1) = \frac{7}{8}$ after observing $C = 1$. Similarly, our prior belief $p(A = 0) = \frac{13}{24}$ of observing $A = 0$ decreases to $p(A = 0 \mid C = 1) = \frac{1}{8}$ after observing $C = 1$.

Note that to answer our query we had

1. to *marginalize*, that is, summing out the probabilities of non-relevant variables in the joint probability distribution of the vector of random variables,

2. to *condition*, that is, to specialize the probability distributions such that it conforms to the given evidence, and

3. to compute the outcome that *maximizes* the probability.

These three operations and combinations thereof make up the core of probabilistic inference queries. The main tasks are to learn the probability distribution from observations and to implement inference queries efficiently.
The size of our basic data structure, the probability table, becomes a challenge when addressing both tasks, because we need $2^n - 1$ parameters to fully specify a probability table of dimension n. In our online shop example n is the number of customers, which is set to $n = 3$ in the example.

Independencies and Graphical Models

In many cases, a probability table can be described with fewer parameters than the number of table elements, or more accurately the number of table elements minus one. The reason for this is that some of the random variables are independent or conditionally independent of each other. For instance, in our

example the variables A and B are independent of each other given C. Formally, this means

$$p(A, B \mid C) = p(A \mid C)p(B \mid C).$$

By marginalizing and conditioning we can directly calculate from the probability table for (A, B, C) that

$A \mid C = 1$	1	1	0	0	$A \mid C = 0$	1	1	0	0
$B \mid C = 1$	1	0	1	0	$B \mid C = 0$	1	0	1	0
$p(A \mid C = 1)$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$p(A \mid C = 0)$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
$p(B \mid C = 1)$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$p(B \mid C = 0)$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$p(A, B \mid C = 1)$	$\frac{21}{32}$	$\frac{7}{32}$	$\frac{3}{32}$	$\frac{1}{32}$	$p(A, B \mid C = 0)$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$

Since

$$p(A, B, C) = p(A, B \mid C)p(C) = p(A \mid C)p(B \mid C)p(C),$$

where $p(C)$ is the marginal probability function for C, that is, in our case

C	1	0
$p(C)$	$\frac{1}{3}$	$\frac{2}{3}$

,

the probability distribution $p(A, B, C)$ is fully specified by the probability functions

$$p(A \mid C = 1),\ p(A \mid C = 0), p(B \mid C = 1),\ p(B \mid C = 0) \text{ and } p(C)$$

which amounts to only $5 \cdot (2 - 1) = 5$ instead of seven parameters.
Here, we have started with the joint probability distribution for responses by Alice, Bob and Claire, and only observed later that the probabilities for Alice and Bob are conditional independent. Typically, one takes (conditional) independencies already into account when constructing the model (probability table). For instance, we could have observed that Alice and Claire as well as Bob and Claire are friends on a social network. It is plausible to assume that friends can influence each others preferences. Therefore, we could construct the response model such that responses for customers who are not friends on the network are conditionally independent.
Conditional independence has ramifications for inference. We already know that Carol responded positively to the marketing campaign. What happens if we also know Bob's response? We compute

$$p(A = 1 \mid B = 1, C = 1) = \frac{7/32}{1/4} = \frac{7}{8} = \frac{7/96}{1/12} = p(A = 1 \mid B = 0, C = 1).$$

That is, the additional knowledge about Bob's response does not affect the probabilities that only incorporate the knowledge about Claire's response. This is not a coincidence. Using the conditional independence of A and B given C, we can compute

$$p(A \mid B, C) = \frac{p(A, B, C)}{p(B, C)} = \frac{p(A, B \mid C)p(C)}{p(B \mid C)p(C)} = \frac{p(A \mid C)p(B \mid C)}{p(B \mid C)} = p(A \mid C).$$

Similarly, $p(B \mid A, C) = p(B \mid C)$. Hence, (conditional) independencies limit the exploitation of partial observations.

Another concept of independence is marginal independence. Formally, marginal independence of A and B just means that

$$p(A, B) = p(A)p(B).$$

Marginal independence is equivalent to the following factorization

$$p(A, B, C) = p(C \mid A, B)p(A, B) = p(C \mid A, B)p(A)p(B).$$

Note that, given the marginal independence of A and B, six parameters are enough to specify $p(A, B, C)$, namely, four parameters for $p(C \mid A, B)$ and one parameters for each $p(A)$ and $p(B)$. Obviously, marginal independence implies the factorization. The implication in the other direction follows from

$$\begin{aligned} p(A, B) &= \sum_{c \in \{0,1\}} p(A, B, C = c) = \sum_{c \in \{0,1\}} p(C = c \mid A, B)p(A)p(B) \\ &= p(A)p(B) \sum_{c \in \{0,1\}} p(C = c \mid A, B) = p(A)p(B), \end{aligned}$$

where the last equality is the law of total probability that states that the probabilities need to sum up to 1.
Neither independence concept does imply the other in general, that is, conditional independence does not imply marginal independence and vice versa. For instance, in our example A and B are conditional independent given the value of C, but they are not marginally independent since (for instance)

$$p(A = 1)p(B = 1) = \frac{44}{96} \cdot \frac{56}{96} \neq \frac{29}{96} = p(A = 1, B = 1).$$

There are several ways to encode independencies among the random variables in a vector of random variables in form of a graph whose vertex set is the set of random variables. The graph describes constraints that the joint probability function of the random vector, or more precisely its marginals and conditionals, has to satisfy.

Exercises

Exercise 1. Tom is a tall, handsome man with mild manners. In the evenings he likes to have a glass of red wine while he listens to classical music. Is Tom more likely to be

A. a professor of medieval history, or

B. a truck driver?

Argue for your answer.

Exercise 2. A blood test for some cancer has the following properties:

1. The probability that the test positive is negative given that the patient has cancer is
$$p(+|C = true) = 0.98.$$
2. The probability that the test result is negative given that the patient is healthy is
$$p(-|C = false) = 0.97.$$
3. The probability for person to suffer from cancer is
$$p(C = true) = 0.01.$$

(1) What is the probability that the test result is negative although the patient suffers from cancer?
(2) What is the probability that the patient suffers from cancer given that the test result is positive?
(3) What is the probability that the patient is healthy although the test result is positive?
(4) What is the probability that the patient suffers from cancer although the test result is negative?

Exercise 3. A pregnant woman finds out from a sonogram that she is expecting twin girls. Now she is wondering if the girls are identical or fraternal twins. Her doctor tells her that one third of twin-births are identical and two-thirds are fraternal. Compute the probability that the woman is expecting identical twins.

Hint: You can assume that for non-identical twins it is as likely that they have the same sex as that they have opposite sexes.

Exercise 4. Given are two lottery wheels. The first wheel contains two blanks and the second wheel contains one blank and one winning ticket. Assume that you have chosen one of the lottery wheels at random, each wheel has a probability of $\frac{1}{2}$ to be chosen. From the wheel you have chosen one of the lottery tickets at random, each ticket has a probability of $\frac{1}{2}$ to be chosen. The ticket is a blank. For drawing your next lottery ticket, should you choose the same lottery wheel, switch to the other wheel, or does it not matter.

Exercise 5. Let $p(x, y)$ be a bivariate probability distribution. Is it true that the marginal distributions $p(x)$ and $p(y)$ uniquely determine the joint distribution $p(X, Y)$?

Exercise 6. Let $p(x, y)$ be a bivariate probability distribution. Argue that $p(x, y)$ is determined by the conditional distributions $p(x|y)$ and $p(y|x)$ (under some technical assumptions).

Remark: The proof should generalize to vectors of more than two random variables.

Part I

Models

Chapter 1

Multivariate Categoricals

Let $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ be the Cartesian product of finite sets $\mathcal{X}_i$ with $|\mathcal{X}_i| \geq 2$. A *multivariate categorical* probability distribution p on $\mathcal{X}$ satisfies two conditions:

$$p(x) \geq 0 \text{ for all } x \in \mathcal{X} \quad \text{and} \quad \sum_{x \in \mathcal{X}} p(x) \,=\, 1.$$

The probability distribution p is thus specified by the vector $\mathbf{p} = \big(p(x)\big)_{x \in \mathcal{X}} \in [0,1]^{|\mathcal{X}|}$ that satisfies the constraint

$$\sum_{x \in \mathcal{X}} p(x) \,=\, 1.$$

That is, $\mathbf{p}$ is contained in the $(|\mathcal{X}| - 1)$-dimensional unit simplex $\Delta_{\mathcal{X}}$ whose vertices correspond to the points in $\mathcal{X}$.

Interaction Parameters

Multivariate categoricals become impractical for larger number of variables, because the number of parameters that need to be estimated is too large to handle even for a moderately large number of variables. Already for a few dozen variables model selection becomes practically infeasible, because (1) we typically do not have enough data points to estimate the parameters reliably, and (2) even if we have, in principle, enough data points, the time to compute the estimates, and (3) for doing inference on the resulting, model can become prohibitively large.

For scaling probabilistic modeling to larger numbers of variables, we are looking for models with fewer parameters. In the introduction we have seen that independencies among the variables decrease the effective number of parameters.

Here, we are studying a different parameterization of multivariate categoricals that can be reduced effectively, while still being flexible enough to model many real-world situations. Specifically, we study the following canonical representation of n-variate categoricals. For $k \in [n]$ let

$$\mathcal{I}_k \;=\; \big\{\{i_1, \ldots, i_k\} \subseteq [n] \;:\; i_1 < i_2 < \ldots < i_k\big\}$$

be parameter index sets and define $p(x) = \exp(q(x))$, where the

$$q(x) \;=\; \sum_{k=1}^{n} \sum_{I=\{i_1,\ldots,i_k\}\in\mathcal{I}_k} q_I(x_{i_1}, \ldots, x_{i_k}) \;-\; a(\mathbf{q})$$

The parameters

$$q_I(x_{i_1}, \ldots, x_{i_k}),\; k \in [n],\; I = \{i_1, \ldots, i_k\} \in \mathcal{I}_k, \;\text{ and } (x_{i_1}, \ldots, x_{i_k}) \in \mathcal{X}_I,$$

are called *canonical* parameters, whereas the $p(x), x \in \mathcal{X}$ are known as *natural* parameters. Notably, $a(\mathbf{q})$ is not a parameter, but the log-normalizing constant

$$a(\mathbf{q}) \;=\; \log\Big(\sum_{x\in\mathcal{X}} \exp\Big(\sum_{k=1}^{n} \sum_{I=\{i_1,\ldots,i_k\}\in\mathcal{I}_k} q_I(x_{i_1}, \ldots, x_{i_k})\Big)\Big).$$

Nevertheless, the number of canonical parameters is much larger than the number $|\mathcal{X}|$ of natural parameters. Therefore, this parameterization of multivariate categoricals is called *overcomplete*, meaning that there are more parameters than necessary and we can remove redundant parameters from the parameterization. It is also not *identifiable*, as we will explain next.

Remarks: (1) Note that with this parameterization we have $p(x) > 0$ for all $x \in \mathcal{X}$. In other words, we can use this type of parameterization only for *strictly positive* distributions. (2) Remember that $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_1$ is a Cartesian product of finite sets $\mathcal{X}_i$ with $|\mathcal{X}_i| \geq 2$. In the following we are using the convention that $\mathcal{X}_i = [n_i]$ with $n_i \geq 2$.

Identifiability

Let

$$\mathcal{P} \;=\; \{p_\theta \;:\; \theta \in \Theta\}$$

be a parameterized family of probability distributions. We call $\mathcal{P}$ *identifiable*, if the mapping $\theta \to p_\theta$ is injective, that is, if

$$p_\theta \;=\; p_{\hat{\theta}} \quad \text{implies that} \quad \theta \;=\; \hat{\theta}.$$

An important consequence of identifiability is that it allows us to interpret the parameters, that is, to assign semantics (meaning) to them.

The natural parameterization $p(x), x \in \mathcal{X}$ of a multivariate categorical p is identifiable. It is, however, overcomplete, because there is one redundant parameter that can be recovered from the others. Here, we arbitrarily choose $p(1,\ldots,1)$ to be the redundant parameter, and only consider the natural parameters $p(x), x \in \mathcal{X} \setminus \{(1,\ldots,1)\}$. Canoncial parameterizations of p are, in contrast to its natural parameterization, not identifiable. Adding some constant c to every parameter

$$q_I(x_{i_1},\ldots,x_{i_k}),\ k \in [n],\ I = \{i_1,\ldots,i_k\} \in \mathcal{I}_k, \text{ and } (x_{i_1},\ldots,x_{i_k}) \in \mathcal{X}_I$$

changes the log-normalization constant $a(\mathbf{q})$ by the additive term

$$\log|\mathcal{X}| + \sum_{k=1}^{n} \sum_{I=\{i_1,\ldots,i_k\} \in \mathcal{I}_k} c,$$

but not $p(x) = \exp\big(q(x)\big)$ for all $x \in \mathcal{X}$. To make canoncial parameterizations identifiable, we have to impose constraints. Here, we impose the constraints that $q_I(x_{i_1},\ldots,x_{i_k}) = 0$, whenever at least one of the x_{i_j} takes the value 1. We call a canonical parameterization that satisfies these constraints an *interaction* parameterization. The interaction parameters of an n-variate categorical p are given by $q_I(x_{i_1},\ldots,x_{i_k})$, where

$$(x_{i_1},\ldots,x_{i_k}) \in \{2,\ldots,n_{i_1}\} \times \ldots \times \{2,\ldots,n_{i_k}\} =: \hat{\mathcal{X}}_I$$

and $I = \{i_1,\ldots,i_k\} \in \mathcal{I}_k, k \in [n]$. With this choice, we no longer have the freedom to change the log-normalization constant, because for any two interaction parameterizations $\mathbf{q}$ and $\hat{\mathbf{q}}$, we have

$$a(\mathbf{q}) \;=\; -q(1,\ldots,1) \;=\; -\log p(1,\ldots,1) \;=\; -\hat{q}(1,\ldots,1) \;=\; a(\hat{\mathbf{q}}).$$

For showing that interaction parameterizations are identifiable we first show that the number of interaction parameters is the same as the number of natural parameters $p(x), x \in \mathcal{X} \setminus \{(1,\ldots,1)\}$, that is, $|\mathcal{X}| - 1$.

Lemma 1. *The number of interaction parameters for a n-variate categorical p is $n_1 \cdot \ldots \cdot n_n - 1$, where n_i is the size of the domain $\mathcal{X}_i$ of the i-th variable.*

Proof. Let the functions $q_I, I = \{i_1,\ldots,i_k\} \in \mathcal{I}_k, k \in [n]$ define an interaction parameterization of p. Each function $q_I, I = \{i_1,\ldots,i_k\} \in \mathcal{I}_k, k \in [n]$ contributes $(n_{i_1}-1)\cdot\ldots\cdot(n_{i_k}-1)$ parameters, because for the argument $x_{i_j}, j \in [k]$

of q_I we have $n_{i_j} - 1$ choices that are different from 1. Therefore, the total number of parameters is

$$\sum_{i=1}^{n}(n_i - 1) + \sum_{i=1}^{n}\sum_{j>i}^{n}(n_i - 1)(n_j - 1) + \ldots + (n_1 - 1) \cdot \ldots \cdot (n_n - 1).$$

We prove by induction over the dimension n that this number is equal to the number $n_1 \cdot \ldots \cdot n_n - 1$ of natural parameters. For the base case $n = 1$, the sum reduces to $\sum_{i=1}^{1}(n_i - 1) = n_1 - 1$. For the induction step, assume that the claim holds for dimension $n - 1$. We can rewrite the sum for dimension n as follows:

$$\begin{aligned}
&(n_1 - 1)\left(\quad 1 \quad + \sum_{j=2}^{n}(n_j - 1) \quad + \sum_{j=2}^{n}\sum_{k>j}^{n}(n_j - 1)(n_k - 1) + \ldots\right) \\
&\qquad + \left(\sum_{i=2}^{n}(n_i - 1) + \sum_{i=2}^{n}\sum_{j>i}^{n}(n_i - 1)(n_j - 1) + \ldots\right) \\
&= (n_1 - 1)(1 + n_2 \cdot \ldots \cdot n_n - 1) + (n_2 \cdot \ldots \cdot n_n - 1) \\
&= n_1 \cdot \ldots \cdot n_n - n_2 \cdot \ldots \cdot n_n + n_2 \cdot \ldots \cdot n_n - 1 \\
&= n_1 \cdot \ldots \cdot n_n - 1,
\end{aligned}$$

where we have used the inductive hypothesis for the $n - 1$ dimensions $i = 2, \ldots, n$. □

Furthermore, for the identifiability proof of interaction parameterizations, we need the following inclusion-exclusion type-of lemma.

Lemma 2. *Let I be a finite set. For any subset $J \subset I$ it holds true that*

$$\sum_{K \subseteq I : J \subseteq K} (-1)^{|K \setminus J|} = 0.$$

Here, $K \subseteq I : J \subseteq K$ means that we sum over all subsets K of I that contain J, that is, the statement after the colon is a condition that needs to be satisfied by the subset K.

Proof. Let $k = |I \setminus J|$. We get that

$$\sum_{K \subseteq I\,:\,J \subseteq K} (-1)^{|K \setminus J|} = \sum_{i=0}^{k}(-1)^i \binom{k}{i} = (1 - 1)^k = 0,$$

when we use the binomial theorem and the observation that, for $i \in [k] \cup \{0\}$, there are $\binom{k}{i}$ subsets K such that $|K \setminus J| = i$. □

Now, we are prepared to prove the identifiability proof of interaction parameterizations.

Theorem 1. *Interaction paramterizations are identifiable, that is, there is exactly one interaction parameterization for every multivariate categorical.*

Proof. For a given multivariate categorical, the number of interaction parameters is the same as the number of natural parameters $p(x), x \in \mathcal{X} \setminus \{(1,\dots,1)\}$ by Lemma 1. Therefore, it is enough to show that any multivariate categorical has at most one interaction parameterization, because then it follows from Lemma 1 that the multivariate categorical has exactly one interaction parameterization. For proving that any multivariate categorical has at most one interaction parameterization we use an argument from linear algebra, namely, the regularity of some matrix that we need to establish first.
Let the functions $q_I, I = \{i_1,\dots,i_k\} \in \mathcal{I}_k, k \in [n]$ define an interaction parameterization of a multivariate categorical p. For $I = \{i_1,\dots,i_k\} \in \mathcal{I}_k, k \in [n]$, let $\hat{\mathcal{X}}_I = \{2,\dots,n_{i_1}\} \times \ldots \times \{2,\dots,n_{i_k}\}$ and

$$\mathbf{v}_I(x) = \Big(\mathbf{1}[x_I = \hat{x}]\Big)_{\hat{x}\in\hat{\mathcal{X}}_I} \in \{0,1\}^{(n_{i_1}-1)\cdot\ldots\cdot(n_{i_k}-1)},\ x \in \mathcal{X} \setminus \big\{(1,\dots,1)\big\},$$

where x_I is the projection of $x \in \mathcal{X}$ onto $\mathcal{X}_I = \mathcal{X}_{i_1} \times \ldots \times \mathcal{X}_{i_k}$. We concatenate the vectors $\mathbf{v}_I(x)$ into a single vector

$$\mathbf{v}(x) = \Big(\mathbf{v}_I(x) : I \in \mathcal{I}_k, k \in [n]\Big) \in \{0,1\}^{n_1\cdot\ldots\cdot n_n - 1},\ x \in \mathcal{X} \setminus \big\{(1,\dots,1)\big\}.$$

There are $|\mathcal{X} \setminus \{(1,\dots,1)\} = n_1 \cdot \ldots \cdot n_n - 1$ many vectors $\mathbf{v}(x), x \in \mathcal{X} \setminus \big\{(1,\dots,1)\big\}$, and, by Lemma 1, the vectors have

$$\sum_{k=1}^{n} \sum_{I\in\mathcal{I}_k} |\hat{\mathcal{X}}_I| = n_1 \cdot \ldots \cdot n_n - 1$$

many entries each. Therefore, $V \in \{0,1\}^{(n_1\cdot\ldots\cdot n_n-1)\times(n_1\cdot\ldots\cdot n_n-1)}$, with columns $\mathbf{v}(x), x \in \mathcal{X} \setminus \big\{(1,\dots,1)\big\}$ is a square matrix. If we can show that the vectors $\mathbf{v}(x), x \in \hat{\mathcal{X}}$ are linearly independent, then V is regular, that is, invertible. The linear independence of the vectors $\mathbf{v}(x)$ would immediately follow, if we can show that any standard basis vector of $\mathbb{R}^{n_1\cdot\ldots\cdot n_k-1}$ can be expressed as a linear combination of vectors $\mathbf{v}(x)$, because this implies that the vectors $\mathbf{v}(x)$ themselves form a basis of $\mathbb{R}^{n_1\cdot\ldots\cdot n_k-1}$. Observe that the entries of the vectors $\mathbf{v}(x)$ can be indexed by the elements $\hat{x} \in \hat{\mathcal{X}}_I, I \in \mathcal{I}_k, k \in [n]$. We show that the standard basis vector $\mathbf{e}_{\hat{x}}$ that corresponds to $\hat{x} \in \hat{\mathcal{X}}_I$ for some fixed

$I \in \mathcal{I}_k, k \in [n]$ as a linear combination of the vectors $\mathbf{v}(x)$. For $J \in 2^I \setminus \{\emptyset\}$, where 2^I denotes the power set of I, define $x^J \in \mathcal{X} \setminus \{(1,\dots,1)\}$ as

$$x_i^J = \begin{cases} \hat{x}_i & : \quad i \in J \\ 1 & : \quad i \notin J \end{cases} \qquad \text{for } i \in [n].$$

We claim that $\mathbf{e}_{\hat{x}}$ can be written as

$$\sum_{i=0}^{k-1} \sum_{J \subseteq I \,:\, |J| = k-i} (-1)^i \, \mathbf{v}\big(x^J\big) =: \mathbf{v},$$

that is, as a linear combination of vectors $\mathbf{v}(x), x \in \mathcal{X} \setminus \big\{(1,\dots,1)\big\}$. Note that $\mathbf{v}\big(x^J\big)$ has the entry 1 at the positions $\hat{x}_K, K \subseteq J \neq \emptyset$ and the entry 0 at the remaining positions. Particularly, $\mathbf{v}\big(x^I\big)$ is the only one of these vectors that has the entry 1 at the position $\hat{x}$. Thus, $\mathbf{v}$ has the entry 1 at the position $\hat{x}$. For a fixed subset $J \subset I$ all vectors $\mathbf{v}\big(x^K\big)$ with $J \subseteq K \subseteq I$ also have the entry 1 at position $\hat{x}_J$, whereas all vectors $\mathbf{v}\big(x^K\big)$ with $K \subset J$ have the entry 0 at position $\hat{x}_J$. Therefore, by Lemma 2, $\mathbf{v}$ has the entry 0 at position $\hat{x}_J$, which shows that, indeed, $\mathbf{e}_{\hat{x}} = \mathbf{v}$.
Since the vectors $\mathbf{v}(x), x \in \mathcal{X} \setminus \big\{(1,\dots,1)\big\}$ are linearly independent, the matrix $V \in \{0,1\}^{(n_1 \cdot \ldots \cdot n_n - 1) \times (n_1 \cdot \ldots \cdot n_n - 1)}$, with columns $\mathbf{v}(x), x \in \mathcal{X} \setminus \big\{(1,\dots,1)\big\}$, is regular.
Now, let $q_I, I \in \mathcal{I}_k, k \in [n]$ define an interaction parameterization of the multivariate categorical p. The interaction parameters can be summarized in the vector

$$\mathbf{q} = \Big(\mathbf{q}_I \,:\, I \in \mathcal{I}_k, k \in [n]\Big) \in \mathbb{R}^{n_1 \cdot \ldots \cdot n_k - 1},$$

which is the concatenation of the vectors

$$\mathbf{q}_I = \Big(q_I(x_{i_1}, \dots, x_{i_k})\Big)_{x \in \hat{\mathcal{X}}_I} \in \mathbb{R}^{(n_{i_1}-1) \cdot \ldots \cdot (n_{i_k}-1)}.$$

Then, we have, for all $x \in \mathcal{X}$,

$$\mathbf{v}(x)^\top \mathbf{q} = \sum_{k=1}^{n} \sum_{I = \{i_1,\dots,i_k\} \in \mathcal{I}_k} q_I(x_{i_1}, \dots, x_{i_k}) = q(x) + a(\mathbf{q}) = \log p(x) + a(\mathbf{q}),$$

which can be summarized as $V^\top \mathbf{q} = \log \mathbf{p} + \mathrm{vec}(a(\mathbf{q}))$.
Assume now that we have another interaction parameterization that is defined by $\hat{q}_I, I \in \mathcal{I}_k, k \in [n]$ of p. Then, we also have $V^\top \hat{\mathbf{q}} = \log \mathbf{p} + \mathrm{vec}(a(\hat{\mathbf{q}}))$. Furthermore, as shown before, we have

$$a(\mathbf{q}) = -q(1,\dots,1) = -\log p(1,\dots,1) = -\hat{q}(1,\dots,1) = a(\hat{\mathbf{q}}).$$

Therefore, $V^\top(\mathbf{q} - \hat{\mathbf{q}}) = 0$, and by the regularity of V that $\mathbf{q} - \hat{\mathbf{q}} = 0$, which implies $\mathbf{q} = \hat{\mathbf{q}}$, and thus the identifiability of interaction parameterizations. $\square$

Remarks: From the proof of Theorem 1 we also get that for any natural parameter vector $\mathbf{p}$ with $p(x) > 0$ for all $x \in \mathcal{X}$, we have a unique interaction parameter vector $\mathbf{q}$ that can be computed from $\mathbf{p}$ as

$$\mathbf{q} \;=\; \big(V^\top\big)^{-1}\Big(\log \mathbf{p} - \mathsf{vec}\big(\log p(1,\ldots,1)\big)\Big).$$

Moreover, the interaction parameterization is not only identifiable, but also stable, because, by the continuity of the logarithm and linear map $\big(V^\top\big)^{-1}$, if $\mathbf{p}$ and $\hat{\mathbf{p}}$ are two natural parameter vectors that are close, then the corresponding interaction parameter vectors $\mathbf{q}$ and $\hat{\mathbf{q}}$ are also close.

Chapter 2

Pairwise Categoricals

Our motivation for introducing interaction parameterizations was providing a means for defining meaningful models with a significantly smaller number of parameters. Pairwise categoricals are such models. Pairwise categoricals are multivariate categoricals, where all interaction parameters that correspond to interactions of three or more variables are set to zero. Note that, for interesting models, we need at least pairwise interactions, because otherwise we would end up with a model where all the variables are independent of each other. In models where all the variables are independent of each other inference is not possible, that is, knowledge about some of the variables does not help us in predicting the values of the remaining variables.

Formally, an n-variate categorical of the form

$$p(x) \;=\; \exp\big(q(x)\big),$$

where

$$q(x) \;=\; \sum_{i=1}^{n} q_i(x_i) + \sum_{i=1}^{n} \sum_{j>i}^{n} q_{ij}(x_i, x_j) - a(\mathbf{q}),$$

with parameters $q_i(x_i), i \in [n]$ and $q_{ij}(x_i, x_j), i \in [n], j > i$, is called a *pairwise categorical*.

Computing Interaction Parameters

We still need our identifiability assumption $q_i(1) = 0, i \in [n]$ and $q_{ij}(x_i, x_j) = 0$, when either $x_i = 1$ or $x_j = 1$, because otherwise we could have different pairwise categorical parameterizations that parameterize the same distribution. Sometimes, however, we are given canoncial pairwise parameters for a multivariate

categorical, and need to compute an equivalent interaction parameterization, which is identifiable. Therefore, let $q_i(x_i), i \in [n]$ and $q_{ij}(x_i, x_j), i \in [n], j > i$ canonical pairwise parameters. We define

$$c_i(x_j) \;=\; q_{ij}(1, x_j) - \frac{q_{ij}(1,1)}{2} \quad \text{and} \quad d_j(x_i) \;=\; q_{ij}(x_i, 1) - \frac{q_{ij}(1,1)}{2}$$

for $x_j \in \mathcal{X}_j$ and $x_i \in \mathcal{X}_i$, and

$$f_i(x_i) \;=\; \sum_{j=1}^{i-1} c_j(x_i) \;+\; \sum_{j>i}^{n} d_j(x_i).$$

Note that

$$\begin{aligned}\sum_{i=1}^{n} f_i(x_i) \;&=\; \sum_{i=1}^{n}\sum_{j=1}^{i-1} c_j(x_i) \;+\; \sum_{i=1}^{n}\sum_{j>i}^{n} d_j(x_i) \\ &=\; \sum_{i=1}^{n}\sum_{j>i}^{n} c_i(x_j) \;+\; \sum_{i=1}^{n}\sum_{j>i}^{n} d_j(x_i) \;=\; \sum_{i=1}^{n}\sum_{j>i}^{n} \big(c_i(x_j) + d_j(x_i)\big).\end{aligned}$$

If we set

$$\hat{q}_i(x_i) = q_i(x_i) + f_i(x_i) - (q_i(1) + f_i(1))$$

and

$$\hat{q}_{ij}(x_i, x_j) = q_{ij}(x_i, x_j) - c_i(x_j) - d_j(x_i),$$

then we get

$$\hat{q}_i(1) \;=\; q_i(1) + f_i(1) - (q_i(1) + f_i(1)) \;=\; 0$$

and

$$\begin{aligned}\hat{q}_{ij}(1, x_j) \;&=\; q_{ij}(1, x_j) - c_i(x_j) - d_j(1) \\ &=\; q_{ij}(1, x_j) - q_{ij}(1, x_j) + \frac{q_{ij}(1,1)}{2} - q_{ij}(1,1) + \frac{q_{ij}(1,1)}{2} \;=\; 0\end{aligned}$$

and

$$\begin{aligned}\hat{q}_{ij}(x_i, 1) \;&=\; q_{ij}(x_i, 1) - c_i(1) - d_j(x_i) \\ &=\; q_{ij}(x_i, 1) - q_{ij}(1,1) + \frac{q_{ij}(1,1)}{2} - q_{ij}(x_i, 1) + \frac{q_{ij}(1,1)}{2} \;=\; 0.\end{aligned}$$

That is, $\hat{q}_i(x_i), i \in [n]$ and $\hat{q}_{ij}(x_i, x_j), i \in [n], j > i$ are interaction parameters. The following calculation shows that $q(x)$ and $\hat{q}(x)$ differ only by an additive

constant, that is, a term that does not depend on x,

$$\begin{aligned}
\hat{q}(x) &= \sum_{i=1}^{n} \hat{q}_i(x_i) + \sum_{i=1}^{n}\sum_{j>i}^{n} \hat{q}_{ij}(x_i, x_j) \\
&= \sum_{i=1}^{n} \big(q_i(x_i) + f_i(x_i) - (q_i(1) + f_i(1))\big) + \sum_{i=1}^{n}\sum_{j>i}^{n} \big(q_{ij}(x_i, x_j) - c_i(x_j) - d_j(x_i)\big) \\
&= \sum_{i=1}^{n} q_i(x_i) + \sum_{i=1}^{n}\sum_{j>i}^{n} q_{ij}(x_i, x_j) - \sum_{i=1}^{n} \big(q_i(1) + f_i(1)\big) \\
&\quad + \sum_{i=1}^{n} f_i(x_i) - \sum_{i=1}^{n}\sum_{j>i}^{n} \big(c_i(x_j) - d_j(x_i)\big) \\
&= \sum_{i=1}^{n} q_i(x_i) + \sum_{i=1}^{n}\sum_{j>i}^{n} q_{ij}(x_i, x_j) - \sum_{i=1}^{n} \big(q_i(1) + f_i(1)\big) \\
&= q(x) - \sum_{i=1}^{n} \big(q_i(1) + f_i(1)\big).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
p(x) &= \frac{\exp\big(q(x)\big)}{\sum_{\bar{x}\in\mathcal{X}} \exp\big(q(\bar{x})\big)} \\
&= \frac{\exp\big(-\sum_{i=1}^{n} \big(q_i(1) + f_i(1)\big)\big)}{\exp\big(-\sum_{i=1}^{n} \big(q_i(1) + f_i(1)\big)\big)} \cdot \frac{\exp\big(q(x)\big)}{\sum_{\bar{x}\in\mathcal{X}} \exp\big(q(\bar{x})\big)} \\
&= \frac{\exp\big(q(x) - \sum_{i=1}^{n} (q_i(1) + f_i(1))\big)}{\sum_{\bar{x}\in\mathcal{X}} \exp\big(q(\bar{x}) - \sum_{i=1}^{n} (q_i(1) + f_i(1))\big)} \\
&= \frac{\exp\big(\hat{q}(x)\big)}{\sum_{\bar{x}\in\mathcal{X}} \exp\big(\hat{q}(\bar{x})\big)},
\end{aligned}$$

which means that $q(x)$ and $\hat{q}(x)$ are equivalent parameterizations of $p(x)$. In the following we assume an identifiable interaction parameterization.

Chapter 3

Markov Random Fields

For pairwise categoricals we set all interaction parameters for three or more variables to zero. Here, we study models where we set all interaction parameters for selected interaction parameter index sets to zero, that is, for a selected index set $I = \{i_1, \ldots, i_k\}$ we set all the parameters $q_I(x_{i_1}, \ldots, x_{i_k}), x_{i_j} \in \hat{\mathcal{X}}_{i_j}, j \in [k]$, to zero. More specifically, let p be a n-variate categorical, and let $\mathcal{I} \subseteq 2^{[n]}$, where $2^{[n]}$ is the power set of $[n]$. We say that p *factorizes* over the interaction parameter index sets in $\mathcal{I}$ if

$$p(x) \;=\; \exp\Big(\sum_{I \in \mathcal{I}} q_I(x_I) - a(\mathbf{q})\Big),$$

where x_I is the projection of the vector x onto the coordinates in I. In the following, we add the emptyset to $\mathcal{I}$ and define $q_\emptyset(x_\emptyset) = -a(\mathbf{q})$. Furthermore, we set $a_I(x_I) = \exp\big(q_I(x_I)\big)$ for $I \in \mathcal{I}$, which makes the factorization more explicit

$$p(x) \;=\; \exp\Big(\sum_{I \in \mathcal{I}} q_I(x_I)\Big) \;=\; \prod_{I \in \mathcal{I}} \exp\big(q_I(x_I)\big) \;=\; \prod_{I \in \mathcal{I}} a_I(x_I).$$

The factorization of a Markov random field has interpretable probabilistic consequences that we are studying in the following.

Markov Blankets and Conditional Independence

Let $J \subset [n]$ and $I \subseteq [n] \setminus J$. Then x_J is called a *Markov blanket* for x_I, if

$$p(x_I \mid x_{-I}) \;=\; p(x_I \mid x_J)$$

for all $x \in \mathcal{X}$. Here, x_I is the projection of x onto the coordinates in I and x_{-I} is the projection of x onto the coordinates in $[n] \setminus I$.

Note that if x_J is a Markov blanket for x_I, then x_K for $K \subset [n] \setminus (I \cup J)$ does does not provide additional information beyond x_J, that is,

$$\begin{aligned}
p(x_I \mid x_J, x_K) &= \sum_{x_L \in \mathcal{X}_{[n]\setminus(I\cup J\cup K)}} p(x_I, x_L \mid x_J, x_K) \\
&= \sum_{x_L \in \mathcal{X}_{[n]\setminus(I\cup J\cup K)}} p(x_I \mid x_J, x_K, x_L) p(x_L \mid x_J, x_K) \\
&= \sum_{x_L \in \mathcal{X}_{[n]\setminus(I\cup J\cup K)}} p(x_I \mid x_{-I}) p(x_L \mid x_J, x_K) \\
&= \sum_{x_L \in \mathcal{X}_{[n]\setminus(I\cup J\cup K)}} p(x_I \mid x_J) p(x_L \mid x_J, x_K) \\
&= p(x_I \mid x_J) \sum_{x_L \in \mathcal{X}_{[n]\setminus(I\cup J\cup K)}} p(x_L \mid x_J, x_K) \\
&= p(x_I \mid x_J).
\end{aligned}$$

The following lemma shows that a factorization induces Markov blankets, which also justifies the name *Markov random field* for factorized distributions. For $i \in [n]$, let $N(i)$ be the index set of *neighbors* of x_i, that is,

$$j \in N(i), \quad \text{if and only if} \quad \exists I \in \mathcal{I} : \{i, j\} \subseteq I.$$

Lemma 3. *Let $\mathcal{I} \subset 2^{[n]}$ such that the distribution p factorizes over $\mathcal{I}$, and let $N(i) \subset [n]$ be the neighbors of x_i. Then $x_{N(i)}$ is a Markov blanket for x_i.*

Proof. Since p factorizes over $\mathcal{I}$, we have that

$$p(x) = \prod_{I \in \mathcal{I}} a_I(x_I) = \Big(\prod_{I \in \mathcal{I} :\, i \in I} a_I(x_I) \Big) \Big(\prod_{I \in \mathcal{I} :\, i \notin I} a_I(x_I) \Big) =: a(x_i, x_{N(i)}) b(x_{-i}).$$

We also have

$$\begin{aligned}
p(x_{-i}) &= \sum_{x_i \in \mathcal{X}_i} p(x) = \sum_{x_i \in \mathcal{X}_i} a(x_i, x_{N(i)}) b(x_{-i}) = b(x_{-i}) \sum_{x_i \in \mathcal{X}_i} a(x_i, x_{N(i)}) \\
&=: b(x_{-i}) \hat{a}(x_{N(i)})
\end{aligned}$$

and

$$\begin{aligned}
p(x_i, x_{N(i)}) &= \sum_{x' \in \mathcal{X}_{[n]\setminus(\{i\}\cup N(i))}} p(x) = \sum_{x' \in \mathcal{X}_{[n]\setminus(\{i\}\cup N(i))}} a(x_i, x_{N(i)}) b(x_{-i}) \\
&= a(x_i, x_{N(i)}) \sum_{x' \in \mathcal{X}_{[n]\setminus(\{i\}\cup N(i))}} b(x_{-i}) =: a(x_i, x_{N(i)}) \hat{b}(x_{N(i)})
\end{aligned}$$

and

$$p(x_{N(i)}) = \sum_{x' \in \mathcal{X}_{[n]\setminus(\{i\}\cup N(i))}} p(x_{-i}) = \sum_{x' \in \mathcal{X}_{[n]\setminus(\{i\}\cup N(i))}} b(x_{-i})\hat{a}(x_{N(i)})$$
$$= \hat{a}(x_{N(i)}) \sum_{x' \in \mathcal{X}_{[n]\setminus(\{i\}\cup N(i))}} b(x_{-i}) = \hat{a}(x_{N(i)})\hat{b}(x_{N(i)}).$$

Therefore,

$$p(x_i \mid x_{-i}) = \frac{p(x)}{p(x_{-i})} = \frac{a(x_i, x_{N(i)})b(x_{-i})}{\hat{a}(x_{N(i)})b(x_{-i})} = \frac{a(x_i, x_{N(i)})}{\hat{a}(x_{N(i)})}$$
$$= \frac{a(x_i, x_{N(i)})\hat{b}(x_{N(i)})}{\hat{a}(x_{N(i)})\hat{b}(x_{N(i)})} = \frac{p(x_i, x_{N(i)})}{p(x_{N(i)})} = p(x_i \mid x_{N(i)}),$$

and $(x_j)_{j\in N(i)}$ is a Markov blanket for x_i. □

We can encode the neighborhood relation among the random variables in the *interaction graph* $G = \big(V = [n], E\big)$, where

$$\{i, j\} \in E, \quad \text{if and only if} \quad \exists I \in \mathcal{I} : \{i, j\} \subseteq I.$$

The induced graphs $G \mid I$ for $I \in \mathcal{I}$ are *complete* and thus the $I \in \mathcal{I}$ are *cliques* of G.

Corollary 1. *If $\{i, j\} \notin E$, then x_i and x_j are independent conditioned on $x_K, K = [n] \setminus \{i, j\}$.*

Proof. We have

$$p(x_i, x_j \mid x_{-\{i,j\}}) = p(x_i \mid x_j, x_{-\{i,j\}})\, p(x_j \mid x_{-\{i,j\}})$$
$$= p(x_i \mid x_{-\{i,j\}})\, p(x_j \mid x_{-\{i,j\}}),$$

where we have used for the second equality that $j \notin N(i)$ and that $x_{N(i)}$ is a Markov blanket for x_i.

□

Remark: The interaction graph thus has an interpretation as *conditional dependence* graph, that is, the absence of an edge in the graph signifies the conditional independence of the corresponding variables.

Hammersley-Clifford Theorem

In many cases it is more natural to provide a conditional dependence graph or a proxy of it instead of a factorization. However, conditional dependence graphs seem to be a weaker concept than factorizations. After all, we have derived the conditional dependence graph from a factorization. The Hammersley-Clifford Theorem, however, asserts that also the converse of Corollary 1 is true for strictly positive distributions. An elegant proof of the Hammersley-Clifford theorem makes use of the Moebius Inversion Lemma that we are proving first.

Lemma 4. [Moebius Inversion] *Let φ and ψ be real valued functions on $2^{[n]}$. The following statements are eqivalent:*

1. *For all $I \in 2^{[n]}$: $\psi(I) = \sum_{J:J\subseteq I} \varphi(J)$*
2. *For all $I \in 2^{[n]}$: $\varphi(I) = \sum_{J:J\subseteq I} (-1)^{|I\setminus J|}\, \psi(J)$*

Proof. Assume that the second statement holds. The first statement can be derived as follows

$$\begin{aligned}
\sum_{J:J\subseteq I} \varphi(J) &= \sum_{J:J\subseteq I} \sum_{K:K\subseteq J} (-1)^{|J\setminus K|}\psi(K) \\
&= \sum_{K:K\subseteq I} \sum_{J:J\subseteq I\wedge K\subseteq J} (-1)^{|J\setminus K|}\psi(K) \\
&= \sum_{K:K\subseteq I} \psi(K) \sum_{J:J\subseteq I\wedge K\subseteq J} (-1)^{|J\setminus K|} \\
&= \sum_{K:K\subseteq I} \psi(K) \sum_{J:J\subseteq I\setminus K} (-1)^{|J|} \\
&= \psi(I),
\end{aligned}$$

where we have used the second statement in the first equality, changed the order of summation in the second equality, and exploited the fact that any non-empty set has the same number of subsets with odd and even (also counting the empty set) cadinality in the last equality, that is,

$$\sum_{J:J\subseteq I\setminus K} (-1)^{|J|} = 0,$$

except for $K = I$, when it is 1. Assume now that the first statement holds. By using similar arguments as above, the second statement can be derived as

follows

$$\begin{aligned}
\sum_{J:J\subseteq I}(-1)^{|I\setminus J|}\psi(J) &= \sum_{J:J\subseteq I}(-1)^{|I\setminus J|}\sum_{K:K\subseteq J}\varphi(K)\\
&= \sum_{J:J\subseteq I}\sum_{K:K\subseteq J}(-1)^{|I\setminus J|}\varphi(K)\\
&= \sum_{K:K\subseteq I}\sum_{J:J\subseteq I\wedge K\subseteq J}(-1)^{|I\setminus J|}\varphi(K)\\
&= \sum_{K:K\subseteq I}\varphi(K)\sum_{J:J\subseteq I\wedge K\subseteq J}(-1)^{|I\setminus J|}\\
&= \sum_{K:K\subseteq I}\varphi(K)\sum_{J:J\subseteq I\setminus K}(-1)^{|I\setminus(K\cup J)|}\\
&= \sum_{K:K\subseteq I}\varphi(K)\sum_{J:J\subseteq I\setminus K}(-1)^{|(I\setminus K)\setminus J)|}\\
&= \sum_{K:K\subseteq I}\varphi(K)\sum_{J:J\subseteq I\setminus K}(-1)^{|J|}\\
&= \varphi(I).
\end{aligned}$$

□

Theorem 2. [Hammersley and Clifford] *Let p be a positive n-variate categorical, and let $G([n],E)$ be the graph for which $\{i,j\}\notin E$ if the variables x_i and x_j are independent conditioned on $x_K, K=[n]\setminus\{i,j\}$. Then p factorizes over the cliques of G, that is,*

$$p(x) = \prod_{I:\,G|I \text{ is complete}} a_I(x_I).$$

Proof. Here, we are proving the following statement

$$\log p(x) = \sum_{I:\,G|I \text{ is complete}} \log a_I(x_I),$$

which is equivalent to the factorization claim. Fix $\hat{x}=(\hat{x}_1,\ldots,\hat{x}_n)\in\mathcal{X}$, and, for every $I\subseteq[n]$, define

$$\psi_I(x) := \log p\big(x_I,\hat{x}_{-I}\big),$$

where $\hat{x}_{-I}$ is the projection of $\hat{x}$ onto the coordinates in $[n] \setminus I$. Hence $\psi_I(x)$ depends only on the projection of x onto the coordinates in I. Let

$$\varphi_I(x) := \sum_{J:J\subseteq I} (-1)^{|I\setminus J|}\psi_J(x).$$

By construction φ_I also depends only on the projection of x onto the coordinates in I. With the definitions $\psi_I(x) =: \psi(I)$ and $\varphi_I(x) =: \varphi(I)$ for any fixed x, we can apply the Moebius Inversion Lemma which gives

$$\psi_I(x) = \sum_{J:J\subseteq I} \varphi_J(x).$$

Since $\psi_{[n]}(x) = \log p(x)$ this implies for the special case $I = [n]$,

$$\log p(x) = \psi_{[n]}(x) = \sum_{I:I\subseteq[n]} \varphi_I(x).$$

This is almost what we set out to prove. It remains to show that the functions φ_I are identical zero for all $I \subseteq [n]$ such that $G|I$ is not complete, that is, not all vertices in I are connected in the graph G. Assume that $i, j \in I$ and $\{i, j\} \notin E$, that is, x_i and x_j are independent conditioned on $x_{[n]\setminus\{i,j\}}$. Let $J = I \setminus \{i, j\}$. We have

$$\begin{aligned}
\varphi_I(x) &= \sum_{K:K\subseteq I} (-1)^{|I\setminus K|}\psi_K(x) \\
&= \sum_{K:K\subseteq J\cup\{i,j\}} (-1)^{|I\setminus K|}\psi_K(x) \\
&= \sum_{K:K\subseteq J} (-1)^{|I\setminus K|}\psi_K(x) + \sum_{K:K\subseteq J} (-1)^{|I\setminus (K\cup\{i\})|}\psi_{K\cup\{i\}}(x) \\
&\quad + \sum_{K:K\subseteq J} (-1)^{|I\setminus (K\cup\{j\})|}\psi_{K\cup\{j\}}(x) + \sum_{K:K\subseteq J} (-1)^{|I\setminus (K\cup\{i,j\})|}\psi_{K\cup\{i,j\}}(x) \\
&= \sum_{K:K\subseteq J} (-1)^{|I\setminus K|}\psi_K(x) - \sum_{K:K\subseteq J} (-1)^{|I\setminus K|}\psi_{K\cup\{i\}}(x) \\
&\quad - \sum_{K:K\subseteq J} (-1)^{|I\setminus K|}\psi_{K\cup\{j\}}(x) + \sum_{K:K\subseteq J} (-1)^{|I\setminus K|}\psi_{K\cup\{i,j\}}(x) \\
&= \sum_{K:K\subseteq J} (-1)^{|I\setminus K|}\left(\psi_K(x) - \psi_{K\cup\{i\}}(x) - \psi_{K\cup\{j\}}(x) + \psi_{K\cup\{i,j\}}(x)\right) \\
&= \sum_{K:K\subseteq J} (-1)^{|J\setminus K|}\left(\psi_K(x) - \psi_{K\cup\{i\}}(x) - \psi_{K\cup\{j\}}(x) + \psi_{K\cup\{i,j\}}(x)\right).
\end{aligned}$$

We are done if we can show that

$$\psi_K(x) - \psi_{K\cup\{i\}}(x) - \psi_{K\cup\{j\}}(x) + \psi_{K\cup\{i,j\}}(x) \;=\; 0.$$

Using $\overline{K} = [n] \setminus (K \cup \{i,j\})$, this follows immediately from

$$\begin{aligned}
\psi_{K\cup\{i,j\}}(x) - \psi_{K\cup\{i\}}(x) \;&=\; \log p\big(x_K, x_i, x_j, \hat{x}_{\overline{K}}\big) \;-\; \log p\big(x_K, x_i, \hat{x}_j, \hat{x}_{\overline{K}}\big) \\
&= \log \frac{p\big(x_K, x_i, x_j, \hat{x}_{\overline{K}}\big)}{p\big(x_K, x_i, \hat{x}_j, \hat{x}_{\overline{K}}\big)} \\
&= \log \frac{p\big(x_i, x_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)}{p\big(x_i, \hat{x}_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)} \\
&= \log \frac{p\big(x_i \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)}{p\big(x_i \mid x_K, \hat{x}_{\overline{K}}\big)p\big(\hat{x}_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)} \\
&= \log \frac{p\big(\hat{x}_i \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)}{p\big(\hat{x}_i \mid x_K, \hat{x}_{\overline{K}}\big)p\big(\hat{x}_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)} \\
&= \log \frac{p\big(\hat{x}_i, x_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)}{p\big(\hat{x}_i, \hat{x}_j \mid x_K, \hat{x}_{\overline{K}}\big)p\big(x_K, \hat{x}_{\overline{K}}\big)} \\
&= \log \frac{p\big(x_K, \hat{x}_i, x_j, \hat{x}_{\overline{K}}\big)}{p\big(x_K, \hat{x}_i, \hat{x}_j, \hat{x}_{\overline{K}}\big)} \;=\; \psi_{K\cup\{j\}}(x) - \psi_K(x),
\end{aligned}$$

where, for the fourth and sixth equalities, we have used the conditional independence of x_i and x_j. Note that all quotients are well defined since the distribution p is strictly positive. This completes the proof of the factorization claim. □

Remarks: In practice, the interaction graph is often given as background information. For the example in the introduction, the interaction graph could be a social network that includes Alice, Bob, and Claire. A fairly natural assumption could be that two variables, that correspond to social network nodes that are not connected, are independent conditioned on the other nodes in the network. In other settings, for example biology or chemistry interactions between proteins or, more generally, molecules can be observed directly. Again, it is fairly natural to assume that to variables that correspond to non-interacting molecules are conditionally independent.
Note that an interaction graph does only provide information about conditional (in)dependencies, but not on parameters.

Chapter 4

Bayesian Networks

Bayesian networks are special Markov random fields. A *Bayesian network* is a multivariate categorical where certain conditional independencies hold that are specified by a directed acyclic graph (DAG) on the index set $[n]$ of the variables. More specifically, the DAG encodes conditional independency constraints on probability distributions p on the sample space $\mathcal{X}$, namely, any variable x_i is conditionally independent from its *non-descendants* given the values of its *parents*. Here, parents and (non-)descendants are defined with respect to the DAG. The parents of node i are all nodes j such that (j, i) is an edge in the DAG. Any node j is a descendent of node i if there exists a directed path in the DAG that connects i to j. The non-descendants of node i are all nodes that are neither i nor its parents nor descendants of i.

Conditional Probability Tables

A probability distribution p that is compatible with the DAG constraints is defined by *conditional probability tables* (CPTs) for all nodes given the values of their parents nodes. The size of the CPT for node i is $|\mathcal{X}_{pa(i)} \times \mathcal{X}_i|$, where $pa(i) \subset [n]$ is the set of parents of node i. Actually, since for any fixed $x_{pa(i)} \in \mathcal{X}_{pa(i)}$ one of the probabilities $p(x_i \mid x_{pa(i)})$ can be inferred from the others, because

$$\sum_{x_i \in \mathcal{X}_i} p(x_i \mid x_{pa(i)}) \;=\; 1,$$

it is sufficient to store only $|\mathcal{X}_{pa(i)}| \cdot \big(|\mathcal{X}_i| - 1\big)$ entries of the full CPT In case that a node i is root of the DAG, that is, a node without parents, the full CPT contains just $|\mathcal{X}_i|$ values, one of which is redundant.

We use the CPTs to define a probability distribution on the whole sample space that satisfies the independencies that are specified by the DAG. For the definition, observe that the DAG specifies a partial order on $[n]$ that we can complete into a total order. Without loss of generality we can assume that this total order is the natural order on $[n]$, otherwise we just relable the random variables and permute the dimensions of the sample space accordingly. Now the indices of the parents of node i have to be smaller than i.

Chain rule. By the definition of conditional probabilities we always have the so called *chain rule* of probabilities

$$\begin{aligned} p(x_1, \ldots, x_n) &= p(x_n \mid x_{<n}) p(x_{<n}) \\ &= p(x_n \mid x_{<n}) p(x_{n-1} \mid x_{<n-1}) p(x_{<n-1}) \\ &= \ldots \\ &= \prod_{i=1}^{n} p(x_i \mid x_{<i}). \end{aligned}$$

Here, we do not know the $p(x_i \mid x_{<i})$ but $p(x_i \mid x_{pa(i)})$. Hence, we just define

$$p(x_i \mid x_{<i}) := p(x_i \mid x_{pa(i)}),$$

that is, changing the values of the variables whose corresponding nodes are not parents of node i does not affect the value of the conditional distribution $p(x_i \mid x_{<i})$, or in other words $x_{pa(i)}$ is a Markov blanket for x_i with respect to the the marginal distribution $p(x_i, x_{<i})$. The chain rule now simplifies to

$$p(x_1, \ldots, x_n) := \prod_{i=1}^{n} p(x_i \mid x_{pa(i)}).$$

Validity and Conditional Independencies

We need to verify that p is valid probability distribution that satisfies the independencies that are specified by the DAG. We start by showing that p is valid. By definition, we have $p(x) \geq 0$ for all elements of the sample space. Hence, it remains to show that these probabilities sum up to 1. We do so by induction on the dimension n of the sample space. For $n = 1$, that this, one-dimensional sample spaces, the probabilities sum up to 1 since, by definition, the single CPT is a valid probability table. Now assume that the assertion holds

for $n-1$. We have

$$\begin{aligned}
\sum_{x\in\mathcal{X}}\prod_{i=1}^{n} p(x_i \mid x_{pa(i)}) &= \sum_{x\in\mathcal{X}} p(x_n \mid x_{pa(n)}) \prod_{I=1}^{n-1} p(x_i \mid x_{pa(i)}) \\
&= \sum_{x_{-n}\in\mathcal{X}_1\times\ldots\times\mathcal{X}_{n-1}} \sum_{x_n\in\mathcal{X}_n} p(x_n \mid x_{pa(n)}) \prod_{i=1}^{n-1} p(x_i \mid x_{pa(i)}) \\
&= \sum_{x_{-n}\in\mathcal{X}_1\times\ldots\times\mathcal{X}_{n-1}} \Big(\sum_{x_n\in\mathcal{X}_n} p(x_n \mid x_{pa(n)}) \Big) \prod_{i=1}^{n-1} p(x_i \mid x_{pa(i)}) \\
&= \sum_{x_{-n}\in\mathcal{X}_1\times\ldots\times\mathcal{X}_{n-1}} \prod_{i=1}^{n-1} p(x_i \mid x_{pa(i)}) \;=\; 1,
\end{aligned}$$

where we have used in the second to last equality that the CPT for x_n is valid, and the induction hypothesis in the last equality.
It remains to show the independencies that are implied by the DAG. The non-descendants of node i are the nodes whose index is smaller than i that are not parents of node i and nodes with index larger than i that are not connected to i by an oriented path in the DAG. Let $nd(i) \subset [n]$ be the index set of the non-descendants of i, $nd^+(i) \subseteq nd(i)$ be the subset of indices greater than i, and $nd^-(i) \subseteq nd(i)$ be the subset of indices smaller than i. We have

$$\begin{aligned}
&p(x_i, x_{nd(i)} \mid x_{pa(i)}) \\
&\quad= p(x_{nd^+(i)}, x_i, x_{nd^-(i)} \mid x_{pa(i)}) \\
&\quad= p(x_{nd^+(i)} \mid x_i, x_{pa(i)}, x_{nd^-(i)}) p(x_i \mid x_{pa(i)}, x_{nd^-(i)}) p(x_{nd^-(i)} \mid x_{pa(i)}) \\
&\quad= p(x_{nd^+(i)} \mid x_i, x_{pa(i)}, x_{nd^-(i)}) p(x_i \mid x_{pa(i)}) p(x_{nd^-(i)} \mid x_{pa(i)}) \\
&\quad= p(x_{nd^+(i)} \mid x_{pa(i)}, x_{nd^-(i)}) p(x_i \mid x_{pa(i)}) p(x_{nd^-(i)} \mid x_{pa(i)}) \\
&\quad= p(x_i \mid x_{pa(i)}) p(x_{nd^+(i)} \mid x_{pa(i)}, x_{nd^-(i)}) p(x_{nd^-(i)} \mid x_{pa(i)}) \\
&\quad= p(x_i \mid x_{pa(i)}) p(x_{nd^-(i)}, x_{nd^+(i)} \mid x_{pa(i)}) \\
&\quad= p(x_i \mid x_{pa(i)}) p(x_{nd(i)} \mid x_{pa(i)}),
\end{aligned}$$

where we have used $nd(i) = nd^-(i) \cup nd^+(i)$ for the first and for the last equality, and the definition of conditional probabilities for the second and sixth equality. The fifth equality just commutes the first two factors. The third equality is implied by $p(x_i \mid x_{<i}) = p(x_i \mid x_{pa(i)})$. The fourth equality follows from the same principle and the observations that the indices in $\{i\} \cup pa(i) \cup nd^-(i)$ are all smaller than the indices in $nd^+(i)$, and that the index sets of the parents of the variables $x_{nd^+(i)}$ are contained in $pa(i) \cup nd^-(i)$, because by the definition

of $nd^+(i)$ neither x_i nor any of its descendants can be a parents of the variables $x_{nd^+(i)}$.

Remark: (1) The DAG and the CPTs uniquely define the probability distribution p, but there can be more DAGs and associated CPTs that also define p. That is, the DAG is not *identifiable* since several DAGs can lead to the same probability distribution. Hence, it is problematic to assign a meaning to the DAG like for instance that the edges signify *causal* dependencies. Nevertheless, Bayesian networks can serve as a substrate for casual models, but this requires some work and care. (2) The conditional independencies implied by the DAG are typically not the only independencies that hold for the corresponding distribution. The additional independencies can be discovered by a purely graph theoretic concept called *d-separation*. (3) The DAG structure makes it more or less straightforward to sample from a Bayesian network, while sampling from a Markov random field is, in general, more complicated.

Chapter 5

Exercises

Chapter 3

Exercise 1. Going back to the example from the introduction. What is interaction graph of this model?

Part II

Model Selection

Chapter 6

Maximum Likelihood and MAP

The maximum likelihood approach is an optimization approach for the model selection problem. We illustrate the maximum likelihhod and the related maximum a posteriori approach on the model selection problem for multivariate categorical distributions by estimating the parameter vector $\mathbf{p}$ of natural parameters from observed data points. We assume that we are given m data points $x^{(1)}, \ldots, x^{(m)}$, with n entries each, drawn *independently* from the multivariate categorical distribution p on $\mathcal{X}$. We denote the sequence of data points also as X.

For our introductory example we would have to learn a vector $\mathbf{p}$ of eight probability values, one for each element of the sample space $\{0,1\}^3$. Of course, later we realized that it is enough to learn five other parameters from which we can compute the eight elements of $\mathbf{p}$. Still, in the end we need to know $\mathbf{p}$. In the Bayesian philosophy the vector $\mathbf{p}$ itself is treated as a random variable with *(prior) distribution* $p(\mathbf{p})$ on a sample space $\mathcal{P}$ for $\mathbf{p}$. The evidence, that is, independently sampled data points $x^{(1)}, \ldots, x^{(m)} \in \mathcal{X}^m$, is used for updating our understanding of the distribution of $\mathbf{p}$ from the prior to the *posterior distribution* $p(\mathbf{p} \mid x^{(1)}, \ldots, x^{(m)})$. This assumes that we have a joint distribution p on the Cartesian product of simplex $\Delta_{\mathcal{X}}$ and the sample space $\mathcal{X}^m$. Note that here the meaning of p is overloaded but should be clear from the context. The update is then facilitated by Bayes theorem, that is,

$$p(\mathbf{p} \mid x^{(1)}, \ldots, x^{(m)}) = \frac{p(\mathbf{p}, x^{(1)}, \ldots, x^{(m)})}{p(x^{(1)}, \ldots, x^{(m)})} = \frac{p(x^{(1)}, \ldots, x^{(m)} \mid \mathbf{p})\, p(\mathbf{p})}{p(x^{(1)}, \ldots, x^{(m)})},$$

and referred to as *Bayesian inference*. Of course, we are still left with the freedom to model the prior distribution $p(\mathbf{p})$. The controversy about the Bayesian method

is that there is no generally accepted principle for choosing the prior distribution. Nevertheless, the method has proven to be effective in many applications.

Dirichlet prior. A computionally attractive choice for the prior distribution is a *Dirichlet distribution*. A Dirichlet distribution is given by the following probability distribution on the $(|\mathcal{X}| - 1)$-dimensional unit simplex $\Delta_{\mathcal{X}}$

$$p(\mathbf{p}) \;=\; \frac{1}{D(\boldsymbol{\alpha})} \prod_{x \in \mathcal{X}} p(x)^{\alpha(x)-1} \;\sim\; \mathsf{Dir}(\boldsymbol{\alpha}),$$

where X is the sequence of data points and $\boldsymbol{\alpha} = \big(\alpha(x)\big)_{x \in \mathcal{X}} \in \mathbb{R}^{|\mathcal{X}|}$ is a parameter vector,

$$D(\boldsymbol{\alpha}) \;=\; \frac{\prod_{x \in \mathcal{X}} \Gamma(\boldsymbol{\alpha}(x))}{\Gamma\left(\sum_{x \in \mathcal{X}} \alpha(x)\right)}$$

is a normalizing constant and $\Gamma(\cdot)$ is the Gamma function.

In general, the posterior distribution is given as

$$p(\mathbf{p} \mid X) \;=\; \frac{p(X \mid \mathbf{p})\, p(\mathbf{p})}{p(X)},$$

where $p(\cdot \mid \mathbf{p})$ is a multinomial distribution with parameters $\mathbf{p}$. The likelihood term $p(X \mid \mathbf{p})$ is given as

$$p(X \mid \mathbf{p}) \;=\; p(x^{(1)}, \ldots, x^{(m)} \mid \mathbf{p}) \;=\; \prod_{i=1}^{m} p(x^{(i)} \mid \mathbf{p}),$$

where the second equality follows from the independence of the data points. Furthermore, we have

$$\prod_{i=1}^{m} p(x^{(i)} \mid \mathbf{p}) \;=\; \prod_{x \in \mathcal{X}} p(x)^{m(x)},$$

where $m(x)$ is the number of observations for the vector $x \in \mathcal{X}$. Obviously, we have $\sum_{x \in \mathcal{X}} m(x) = m$, where m is the number of data points. The vector $\mathbf{m} = \big(m(x)\big)_{x \in \mathcal{X}} \in \mathbb{N}^{|\mathcal{X}|}$ is called a *contingency table*.

For the special case of a Dirichlet prior, the posterior distribution has the form

$$\begin{aligned} p(\mathbf{p} \mid X) \;&=\; \frac{p(X \mid \mathbf{p})\, p(\mathbf{p})}{p(X)} \\ &=\; \left(\frac{1}{p(X)} \prod_{x \in \mathcal{X}} p(x)^{m(x)}\right) \left(\frac{1}{D(\boldsymbol{\alpha})} \prod_{x \in \mathcal{X}} p(x)^{\alpha(x)-1}\right) \;\sim\; \mathsf{Dir}\big(\mathbf{m} + \boldsymbol{\alpha}\big), \end{aligned}$$

that is, it is again a Dirichlet distribution. That is why we call the Dirichlet distribution a *conjugate prior distribution* for the parameter vector $\mathbf{p}$ of multivariate categorical distribution.

Maximum a posteriori estimate. Sometimes a point estimate for the parameters is preferred over a distribution on the space of parameters. A natural choice for a point estimate is the maximum of the posterior distribution, which is called the *maximum a posteriori (MAP)* estimate of $\mathbf{p}$. For a sequence X of m data points $x^{(1)}, \ldots, x^{(m)}$, it is given as

$$\begin{aligned}
\mathbf{p}_{MAP} &= \mathsf{argmax}_{\mathbf{p}}\ p(\mathbf{p} \mid X) \\
&= \mathsf{argmax}_{\mathbf{p}}\ \frac{p(X \mid \mathbf{p})\, p(\mathbf{p})}{p(X)} \\
&= \mathsf{argmax}_{\mathbf{p}}\ p(X \mid \mathbf{p})\, p(\mathbf{p}) \\
&= \mathsf{argmax}_{\mathbf{p}}\ \log p(X \mid \mathbf{p}) + \log p(\mathbf{p}),
\end{aligned}$$

where the last equality holds true, because the logarithm is a monotonously increasing function. The first term of the last objective function, that is, the function

$$\ell(\mathbf{p}) = \log p(X \mid \mathbf{p}) = \log \left(\prod_{x \in \mathcal{X}} p(x)^{m(x)} \right) = \sum_{x \in \mathcal{X}} m(x) \log p(x)$$

is called the *log-likelihood function*.

Note that $\mathbf{p}$ must satisfy the constraint $\sum_{x \in \mathcal{X}} p(x) = 1$. By the Lagrange multiplier theorem, a necessary condition for an optimum is that the gradient of the objective function, in our case

$$\log p(X \mid \mathbf{p}) + \log p(\mathbf{p})$$

is a multiple of the gradient of the constraint function at the optimal solution. Therefore, we get that

$$\nabla_{\mathbf{p}} \big(\ell(\mathbf{p}) + \log p(\mathbf{p}) \big) = \mathsf{vec}(\lambda)$$

for some vector with entry λ in each component. Using $\sum_{x \in \mathcal{X}} p(x) = 1$ and $\sum_{x \in \mathcal{X}} m(x) = m$, this solves to

$$\mathbf{p}_{MAP} = \left(\frac{m(x) + \alpha(x) - 1}{m + \sum_{x' \in \mathcal{X}} \alpha(x') - |\mathcal{X}|} \right)_{x \in \mathcal{X}}.$$

Maximum likelihood estimate. Without the (Dirichlet) prior the MAP optimization problem reduces to a likelihood optimization problem

$$\mathbf{p}_{ML} \;=\; \mathrm{argmax}_{\mathbf{p}}\; p(X \mid \mathbf{p}),$$

which selects the parameter vector $\mathbf{p} = \big(p(x)\big)_{x\in\mathcal{X}}$ that maximizes the probability of observing the sequence X. Maximizing the log-likelihood function $\ell(\mathbf{p})$ under the constraints

$$\sum_{x\in\mathcal{X}} m(x) \;=\; m \qquad \text{and} \qquad \sum_{x\in\mathcal{X}} p(x) \;=\; 1$$

gives the maximum likelihood estimate $\mathbf{p}_{ML} \;=\; \Big(\frac{m(x)}{m}\Big)_{x\in\mathcal{X}}$.

Remark: The maximum a posteriori estimate can be seen as a regularized variant of the maximum likelihood estimate. The special case $\alpha(x) = k + 1$ for every $x \in \mathcal{X}$, where we assume that every x has occurred $k \geq 1$ times more than it actually did, is known as Laplace smoothing.

Chapter 7

Pairwise Categoricals

The number of data points per parameter obviously becomes better when moving from general multivariate categoricals to pairwise categoricals. Therefore, we should be able to learn the parameters of pairwise models more reliably. It is, however, not obvious that the pairwise parameters can also be learned more efficiently. Our standard approach for parameter estimation is the maximum likelihood approach. It turns out that the maximum likelihood approach for pairwise categoricals behaves nicely in statistical sense, but cannot be implemented efficiently.
For deriving the maximum likelihood and studying its properties, it helps to summarize the pairwise interaction parameters in a block parameter matrix

$$Q = \left(\begin{array}{c|c|c} Q_{11} & \cdots & Q_{1n} \\ \hline \vdots & \ddots & \vdots \\ \hline Q_{n1} & \cdots & Q_{nn} \end{array}\right)$$

with

$$\mathbb{R}^{n_i \times n_i} \ni Q_{ii} = \mathsf{diag}\big(2\mathbf{q}_i\big), \quad \mathbf{q}_i = \big(q_i(1) = 0, q_i(2), \ldots, q_i(n_i)\big),\ i \in [n],$$

and

$$\mathbb{R}^{n_i \times n_j} \ni Q_{ij} \ = \ \big(q_{ij}(k,l)\big)_{k \in [n_i], l \in [n_j]} \ \text{ for } i < j, \ \text{ and } Q_{ji} = Q_{ij}^\top \ \text{ for } j > i.$$

Now, by using indicator encodings of $x \in \mathcal{X}$ that are defined as

$$\overline{x} = (\overline{x}_1, \ldots, \overline{x}_n) \in \{0,1\}^{n_1 + \ldots + n_n},$$

with $\overline{x}_i = \big(\mathbf{1}[x_i = 1], \ldots, \mathbf{1}[x_i = n_i]\big) \in \{0,1\}^{n_i}$, $i \in [n]$, we can rewrite pairwise categoricals as

$$p(x) = \exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{x}\,\overline{x}^\top\Big) - a(Q)\right),$$

where $a(Q) = a(\mathbf{q})$, because

$$\begin{aligned}
\mathsf{Trace}\Big(Q\,\overline{x}\,\overline{x}^\top\Big) &= \mathsf{Trace}\big(\overline{x}^\top Q\,\overline{x}\big) = \overline{x}^\top Q\,\overline{x} \\
&= 2\sum_{i=1}^{n}\sum_{j>i}^{n} \overline{x}_i{}^\top Q_{ij}\,\overline{x}_j + \sum_{i=1}^{n} \overline{x}_i{}^\top Q_{ii}\,\overline{x}_i \\
&= 2\sum_{i=1}^{n}\sum_{j>i}^{n} q_{ij}(x_i, x_j) + 2\sum_{i=1}^{n} \mathbf{q}_i^\top \overline{x}_i \\
&= 2\sum_{i=1}^{n}\sum_{j>i}^{n} q_{ij}(x_i, x_j) + 2\sum_{i=1}^{n} q_i(x_i) = 2q(x) + 2\cdot a(Q).
\end{aligned}$$

Now, the likelihood function for the parameters of pairwise categoricals is given as

$$\begin{aligned}
L(Q) &= \prod_{i=1}^{m} \exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{x}^{(i)}\,\overline{x}^{(i)\top}\Big) - a(Q)\right) \\
&= \exp\left(\frac{1}{2}\sum_{i=1}^{m}\mathsf{Trace}\Big(Q\,\overline{x}^{(i)}\,\overline{x}^{(i)\top}\Big) - m\,a(Q)\right) \\
&= \exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\sum_{i=1}^{m}\Big(\overline{x}^{(i)}\,\overline{x}^{(i)\top}\Big)\Big) - m\,a(Q)\right),
\end{aligned}$$

where

$$a(Q) = \sum_{x\in\mathcal{X}} \exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{x}\,\overline{x}^\top\Big)\right).$$

Consequently, the log-likelihood function is given as

$$\ell(Q) = \frac{1}{2}\mathsf{Trace}\Big(Q\sum_{i=1}^{m}\Big(\overline{x}^{(i)}\,\overline{x}^{(i)\top}\Big)\Big) - m\log\left(\sum_{x\in\mathcal{X}}\exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{x}\,\overline{x}^\top\Big)\right)\right).$$

Here, the second term that corresponds to the normalizer $a(Q)$ is problematic, because it is a sum over the whole sample space, which has $\prod_{i=1}^{n} n_i \geq 2^n$-many elements.

To gain additional structural insights, we can rewrite the likelihood function as

$$\begin{aligned} L(Q) &= \exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\sum_{i=1}^{m}\left(\overline{x}^{(i)}\,\overline{x}^{(i)\top}\right)\Big) - m\,a(Q)\right) \\ &= \exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\sum_{x\in\mathcal{X}} m(x)\,\overline{x}\,\overline{x}^{\top}\Big) \;-\; m\,a(Q)\right) \end{aligned}$$

where $\overline{x}^{(i)}$ is the indicator encoding of the data point $x^{(i)}$, $i \in [m]$ and $m(x)$ is number of observations of $x \in \mathcal{X}$. The log-likelihood function is then given as

$$\ell(Q) = \frac{1}{2}\mathsf{Trace}\Big(Q\sum_{x\in\mathcal{X}} m(x)\,\overline{x}\,\overline{x}^{\top}\Big) - m\log\left(\sum_{x\in\mathcal{X}}\exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{x}\,\overline{x}^{\top}\Big)\right)\right).$$

That is, the log-likelihood function is the sum of a linear function and a concave function in its parameters, which are summarized in the matrix Q. Here, we use that log-sum-exp of linear functions is known to be convex. However, in contrast to the natural parameters, the log-likelihood optimization problem for the (pairwise) interaction parameters has not a closed form solution. Nevertheless, it can be solved by numerical methods for satisfying the optimality criterion, that is, a vanishing gradient. The gradient of the log-likelihood function also gives us further insights into the parameter estimation problem:

$$\begin{aligned} \nabla\ell(Q) &= \frac{1}{2}\sum_{x\in\mathcal{X}} m(x)\,\overline{x}\,\overline{x}^{\top} \;-\; \frac{m}{2}\,\frac{\sum_{x\in\mathcal{X}}\overline{x}\,\overline{x}^{\top}\exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{x}\,\overline{x}^{\top}\Big)\right)}{\sum_{\bar{x}\in\mathcal{X}}\exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{\bar{x}}\,\overline{\bar{x}}^{\top}\Big)\right)} \\ &= \frac{1}{2}\sum_{x\in\mathcal{X}} m(x)\,\overline{x}\,\overline{x}^{\top} \;-\; \frac{m}{2}\sum_{x\in\mathcal{X}} p(x)\,\overline{x}\,\overline{x}^{\top} \\ &= \frac{1}{2}\sum_{x\in\mathcal{X}}\big(m(x) - m\,p(x)\big)\,\overline{x}\,\overline{x}^{\top}. \end{aligned}$$

The optimality condition for the maximum log-likelihood problem is a vanishing gradient. Setting the gradient to zero and considering that that the $\Big(\big(\sum_{i=1}^{n} n_i\big) \times \big(\sum_{i=1}^{n} n_1\big)\Big)$-matrices $\overline{x}\,\overline{x}^{\top}$ are symmetric gives us $\binom{\sum_{i=1}^{n} n_1}{2}$ equations for the same number of parameters in the matrix Q. Each equation, however, is a sum over the whole sample space $\mathcal{X}$ of size $n_1 \cdot \ldots \cdot n_n$, which becomes prohibitively large already for a moderately large number n of variables. Moreover, the equations are non-linear in the parameters, and thus cannot be

solved in closed form. The form of the equations, however, provides us some statistical insight into their solutions for large numbers of samples.
Assume that $p(x), x \in \mathcal{X}$ are the unknown *true* natural parameters $p^\star(x), x \in \mathcal{X}$, that is, assume that the data points are independently drawn from the pairwise categorical p with natural parameters $p^\star(x), x \in \mathcal{X}$. By the weak law of large numbers, more specifically, the weak consistency of the maximum likelihood estimate of the natural parameters, we have that

$$\lim_{m\to\infty} \mathsf{P}\Big[\,\Big|\frac{m(x)}{m} - p^\star(x)\Big| \geq \varepsilon\Big] \;=\; 0,$$

for any $\varepsilon > 0$. Here, the probability of the event is computed with respect to the product distribution on $\mathcal{X}^m$ derived from the distribution $p^\star$ on $\mathcal{X}$.
The weak law of large numbers implies that the gradient of the log-likelihood function with respect to the true parameters converges to zero, in probability. Therefore, by the strict concavity of the log-likelihood function, the maximum likelihood estimate of the parameter matrix Q converges in probability to the true parameter matrix, which is why the maximum likelihood estimate of the parameters is called *(weakly) consistent*.

Maximum Pseudo-Likelihood Estimates

The computational bottleneck in the maximum likelihood approach is computing the log-normalization constant $a(Q)$, which requires to sum over the whole sample space $\mathcal{X}$, which is prohibitively large for high-dimensional models, that is, models with many variables. In the pseudo-likelihood approach the expensive to compute log-likelihood function is approximated by a simpler log-pseudo-likelihood function whose log-normalization constant is, in contrast to the maximum likelihood approach, data dependent. The log-normalization constant can be computed by a nested sum over the number m of data points, the number n of dimensions, and the size n_i of the slices of the sample space. That is, the log-normalization constant can be computed in time $O\Big(m\big(\sum_{i=1}^n n_i\big)^2\Big)$. It turns out that the pseudo-likelihood approach is still (weakly) consistent.

Given independent data points $x^{(1)}, \ldots, x^{(m)}$. For the pseudo-likelihood function, we are replacing the probabilities $p(x^{(j)})$ in the likelihood function by a product $\prod_{i=1}^n p\big(x_i^{(j)} \mid x_{-i}^{(j)}\big)$ of node conditionals

$$p\big(x_i^{(j)} \mid x_{-i}^{(j)}\big) \;=\; \exp\left(\frac{1}{2}\,\overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}^{\top} Q\,\overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)} \;-\; a\big(Q, x_{-i}^{(j)}\big)\right),$$

for $x_i^{(j)} \in \mathcal{X}_i = [n_i]$. Here, the entries of

$$x^{(j)} \;=\; \big(x_1^{(j)} \;\ldots, x_n^{(j)}\big) \;=\; (x_i^{(j)}, x_{-i}^{(j)}) \in \mathcal{X},$$

for $-i := [n] \setminus \{i\}$, are not permuted. Moreover, $\overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}$ is the indicator encoding of $x^{(j)}$. The log-normalization constant of a univariate node conditional is given as

$$a\big(Q, x_{-i}^{(j)}\big) \;=\; \log\left(\frac{1}{2}\sum_{x_i=1}^{n_i} \exp\left(\overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}^{\top} Q\, \overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}\right)\right).$$

Hence, the log-pseudo-likelihood function is given as

$$\begin{aligned}
\hat{\ell}(Q) \;&=\; \sum_{j=1}^{m}\sum_{i=1}^{n} \log p\big(x_i^{(j)} \mid x_{-i}^{(j)}\big) \\
&=\; \sum_{j=1}^{m}\sum_{i=1}^{n} \left(\frac{1}{2}\overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}^{\top} Q\, \overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)} \;-\; a(Q, x_{-i}^{(j)})\right).
\end{aligned}$$

The log-pseudo-likelihood function is again the sum of a linear and a concave function, and thus concave itself. The gradient of the log-pseudo-likelihood function reads as

$$\begin{aligned}
\nabla\hat{\ell}(Q) \;=\;& \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{n} \overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}\, \overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}^{\top} \\
&- \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{x_i=1}^{n_i} \frac{\exp\left(\frac{1}{2}\overline{\big(x_i, x_{-i}^{(j)}\big)}^{\top} Q\, \overline{\big(x_i, x_{-i}^{(j)}\big)}\right) \overline{\big(x_i, x_{-i}^{(j)}\big)}\, \overline{\big(x_i, x_{-i}^{(j)}\big)}^{\top}}{\sum_{\bar{x}_i=1}^{n_i} \exp\left(\frac{1}{2}\overline{\big(\bar{x}_i, x_{-i}^{(j)}\big)}^{\top} Q\, \overline{\big(\bar{x}_i, x_{-i}^{(j)}\big)}\right)} \\
=\;& \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{n} \overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}\, \overline{\big(x_i^{(j)}, x_{-i}^{(j)}\big)}^{\top} \\
&- \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{x_i=1}^{n_i} p\big(x_i^{(j)} \mid x_{-i}^{(j)}\big)\, \overline{\big(x_i, x_{-i}^{(j)}\big)}\, \overline{\big(x_i, x_{-i}^{(j)}\big)}^{\top}
\end{aligned}$$

Setting the gradient to zero gives us, as in the maximum likelihood approach, $\binom{\sum_{i=1}^{n} n_1}{2}$ equations for the same number of parameters in the matrix Q. However, in contrast to the maximum likelihood approach, each equation is now a sum of only $m\sum_{i=1}^{n} n_i$ terms. That is, the equations can be evaluated much more efficiently.

Fortunately, in the log-pseudo-likelihood approach for pairwise categoricals, we do not have to sacrifice consistency for efficiency. To see this, we bring the log-pseudo-likelihood functions into a form that differs from the log-likelihood function only in the normalization term and scaling factor of n in the first term, which is realized by summing over n indentical terms. As in the maximum likelihood approach, we can write the log-pseudo-likelihood function also as

$$\begin{aligned}
\hat{\ell}(Q) &= \sum_{j=1}^{m}\sum_{i=1}^{n}\left(\frac{1}{2}\overline{\left(x_i^{(j)}, x_{-i}^{(j)}\right)}^{\top} Q\, \overline{\left(x_i^{(j)}, x_{-i}^{(j)}\right)} \;-\; a(Q, x_{-i}^{(j)})\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{1}{2}\overline{x^{(j)}}^{\top} Q\, \overline{x^{(j)}} \;-\; a(Q, x_{-i}^{(j)})\right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{x\in\mathcal{X}}\Big(m(x)\,\overline{x}^{\top} Q\,\overline{x}\Big) \;-\; \hat{a}(Q) \\
&= \frac{1}{2}\sum_{i=1}^{n}\mathsf{Trace}\Big(Q\sum_{x\in\mathcal{X}} m(x)\,\overline{x}\,\overline{x}^{\top}\Big) \;-\; \hat{a}(Q),
\end{aligned}$$

with

$$\begin{aligned}
\hat{a}(Q) &:= \sum_{j=1}^{m}\sum_{i=1}^{n} a(Q, x_{-i}^{(j)}) \\
&= \sum_{j=1}^{m}\sum_{i=1}^{n}\log\left(\sum_{x_i=1}^{n_i}\exp\left(\frac{1}{2}\overline{\left(x_i, x_{-i}^{(j)}\right)}^{\top} Q\, \overline{\left(x_i, x_{-i}^{(j)}\right)}\right)\right) \\
&= \sum_{x_{-i}\in\mathcal{X}_{-i}} m(x_{-i})\sum_{i=1}^{n}\log\left(\sum_{x_i=1}^{n_i}\exp\left(\frac{1}{2}\overline{(x_i, x_{-i})}^{\top} Q\, \overline{(x_i, x_{-i})}\right)\right) \\
&= \sum_{i=1}^{n}\sum_{x_{-i}\in\mathcal{X}_{-i}} m(x_{-i})\log\left(\sum_{x_i=1}^{n_i}\exp\left(\frac{1}{2}\mathsf{Trace}\big(Q\,\overline{(x_i, x_{-i})}\,\overline{(x_i, x_{-i})}^{\top}\big)\right)\right)
\end{aligned}$$

and

$$m(x_{-i}) \;=\; \sum_{\hat{x}\in\mathcal{X}:\,\hat{x}_{-i}=x_{-i}} m(\hat{x}).$$

Using this representation of the log-pseudo-likelihood function, we get the

following representations of its gradient

$$
\begin{aligned}
&\nabla\hat{\ell}(Q)\\
&= \frac{1}{2}\sum_{i=1}^{n}\Bigg(\sum_{x\in\mathcal{X}} m(x)\overline{xx}^{\top} - \sum_{x_{-i}\in\mathcal{X}_{-i}}\sum_{x_i=1}^{n_i} m(x_{-i})\,p(x_i \mid x_{-i})\,\overline{(x_i,x_{-i})}\,\overline{(x_i,x_{-i})}^{\top}\Bigg)\\
&= \frac{1}{2}\sum_{i=1}^{n}\Bigg(\sum_{x\in\mathcal{X}} m(x)\overline{xx}^{\top} - \sum_{x\in\mathcal{X}} m(x_{-i})\,p(x_i \mid x_{-i})\,\overline{xx}^{\top}\Bigg)\\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{x\in\mathcal{X}}\big(m(x) - m(x_{-i})\,p(x_i \mid x_{-i})\big)\overline{xx}^{\top},
\end{aligned}
$$

where we have used for the first equality that

$$
p(x_i \mid x_{-i}) \;=\; \frac{\exp\Big(\frac{1}{2}\mathsf{Trace}\big(Q\,\overline{(x_i,x_{-i})}\,\overline{(x_i,x_{-i})}^{\top}\big)\Big)}{\sum_{\bar{x}_i=1}^{n_i}\exp\Big(\frac{1}{2}\mathsf{Trace}\big(Q\,\overline{(\bar{x}_i,x_{-i})}\,\overline{(\bar{x}_i,x_{-i})}^{\top}\big)\Big)}.
$$

For the unknown *true* natural parameters $p^{\star}(x), x\in\mathcal{X}$, we get by the weak law of large numbers for the one-dimensional marginals that

$$
\lim_{m\to\infty}\mathsf{P}\Big[\Big|\frac{m(x_{-i})}{m} - p^{\star}(x_{-i})\Big| \geq \varepsilon\Big] \;=\; 0,
$$

for any $\varepsilon > 0$. Therefore, $\frac{m(x_{-i})}{m} = p^{\star}(x_{-i})$ asymptotically in probability, and thus

$$
m(x_{-i})\,p^{\star}(x_i \mid x_{-i}) \;=\; m(x_{-i})\frac{p^{\star}(x_i,x_{-i})}{p^{\star}(x_{-i})} \;=\; \frac{m(x_{-i})}{p^{\star}(x_{-i})}p^{\star}(x) \;=\; m\,p^{\star}(x).
$$

We have seen before that, also by the weak law of large numbers, asymptotically in probability $m(x) = m\,p^{\star}(x)$. Therefore, the weak law of large numbers implies that the gradient of the log-pseudo-likelihood function with respect to the true parameters converges to zero, in probability. That is, in probability, the maximum pseudo-likelihood estimates of the parameters Q converge to the true parameters, and thus, also the maximum pseudo-likelihood estimate is (weakly) consistent.

Appendix: Weak Law of Large Numbers

For stating and proving the weak law of large numbers, we start with the elementary *Markov Inequality* for non-negative random variables.

Lemma 5. [Markov Inequality] *Let X be a non-negative random variable, i.e., $\mathcal{X} \subseteq [0, \infty)$. Then we have for any $t > 0$ that*

$$P[X \geq t] \leq \frac{E[X]}{t}.$$

Proof. We have

$$\begin{aligned}
\mathsf{E}[X] &= \int_{\mathcal{X}} x p(x)\, dx \\
&= \int_{\mathcal{X}} x \cdot \big(\mathbf{1}[x < t] + \mathbf{1}[x \geq t]\big)\, p(x) dx \\
&= \int_{\mathcal{X}} x \cdot \mathbf{1}[x < t]\, p(x) dx \;+\; \int_{\mathcal{X}} x \cdot \mathbf{1}[x \geq t]\, p(x) dx \\
&\geq \int_{\mathcal{X}} x \cdot \mathbf{1}[x \geq t]\, p(x) dx \\
&\geq \int_{\mathcal{X}} t \cdot \mathbf{1}[x \geq t]\, p(x) dx \\
&= t \cdot \int_{\mathcal{X}} \mathbf{1}[x \geq t]\, p(x) dx \\
&= t \cdot \mathsf{P}[X \geq t]
\end{aligned}$$

□

From Markov's inequality we also get inequalities for a real valued random variable X, that is, $\mathcal{X} \subseteq \mathbb{R}$. An immediate consequence is Chebyshev's inequality

$$\mathsf{P}\big[|X - \mathsf{E}[X]| \geq t\big] \;=\; \mathsf{P}\big[(X - \mathsf{E}[X])^2 \geq t^2\big] \;\leq\; \frac{\mathsf{E}\big[\big(X - \mathsf{E}[X]\big)^2\big]}{t^2} \;=\; \frac{\mathsf{Var}[X]}{t^2}.$$

The weak law of large numbers is a consequence of Chebyshev's inequality.

Theorem 3. [Weak law of large numbers] *Let $X_1, \ldots, X_m$ be independent, real-valued random variables with the same expectation, that is, $E[X_1] = \ldots = E[X_m]$, and bounded variance $Var[X_i] \leq c < \infty$ for all $i \in [m]$. Then we have for $X = \frac{1}{m} \sum_{i=1^m} X_i$ and all $\varepsilon > 0$ that*

$$P\Big[\big|X - E[X]\big| \geq \varepsilon\Big] \;\leq\; \frac{c}{m\varepsilon^2}.$$

Proof. By the linearity of expectation we have $\mathsf{E}[X] = \mathsf{E}[X_i]$ for all $i \in [n]$ and

$$\begin{aligned}
\mathsf{Var}[X] &= \mathsf{E}\big[(X - \mathsf{E}[X])^2\big) = \mathsf{E}[X^2] - \mathsf{E}[X]^2 \\
&= \frac{1}{m^2}\mathsf{E}\left[\sum_{i=1}^{m} X_i^2\right] + \frac{2}{m^2}\mathsf{E}\left[\sum_{i=1}^{m}\sum_{j>i}^{m} X_i X_j\right] - \mathsf{E}[X]^2 \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{E}[X_i^2] + \frac{2}{m^2}\sum_{i=1}^{m}\sum_{j>i}^{m}\mathsf{E}[X_i X_j] - \mathsf{E}[X]^2 \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{E}[X_i^2] + \frac{2}{m^2}\sum_{i=1}^{m}\sum_{j>i}^{m}\mathsf{E}[X_i]\mathsf{E}[X_j] - \mathsf{E}[X]^2 \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{E}[X_i^2] + \frac{2}{m^2}\sum_{i=1}^{m}(m-i)\mathsf{E}[X]^2 - \mathsf{E}[X]^2 \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{E}[X_i^2] + \left(\frac{2}{m^2}\frac{m(m-1)}{2} - 1\right)\mathsf{E}[X]^2 \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{E}[X_i^2] - \frac{1}{m}\mathsf{E}[X]^2 \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{E}[X_i^2] - \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{E}[X_i]^2 \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\big(\mathsf{E}[X_i^2] - \mathsf{E}[X_i]^2\big) = \frac{1}{m^2}\sum_{i=1}^{m}\mathsf{Var}[X_i] \leq \frac{c}{m}.
\end{aligned}$$

The weak law of large numbers now follows from Chebyshev's inequality. □

We can use the weak law of large numbers for proving the consistency of estimators. Therefore, we first formally introduce estimators for parameterized families of distributions. Let $\mathcal{X}$ be a sample space,

$$\{p_\theta \,:\, \theta \in \Theta \subseteq \mathbb{R}^k\}$$

a parameterized family of probability distributions on $\mathcal{X}$, and $\mathcal{X}^+$ be the set of all finite sequences in $\mathcal{X}$. An *estimator* is a function

$$T : \mathcal{X}^+ \to \Theta$$

that assigns a parameter value $\theta \in \Theta$ to observations $X \in \mathcal{X}^+$. Let $X^{(m)} \in \mathcal{X}$ be a sequence of m data points drawn independently from the probability

distribution p_θ for a fixed $\theta \in \Theta$. An estimator T is called *(weakly) consistent* if for any $\varepsilon > 0$

$$\lim_{m\to\infty} \mathsf{P}_\theta\big(|T(X^{(m)}) - \theta| \geq \varepsilon\big) \;=\; 0,$$

that is, it is required that in probability the parameter θ can be estimated more accurately with a growing number of data points.

We can show that the maximum likelihood estimator for multivariate categoricals is consistent. To do so, we start with the simple case of a Bernoulli distribution on the sample space $\mathcal{X} = \{0, 1\}$. The natural parameter of a Bernoulli distribution is given by $p = p(x = 1)$. In the exercises you will show that, given m data points that have been sampled independently, the maximum likelihood estimator for the parameter p is given as

$$T_{ML}(X^{(m)}) \;=\; \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}[x^{(i)} = 1].$$

By the linearity of expectation, we get that

$$\mathsf{E}_p\big[T_{ML}(X^{(m)})\big] \;=\; \frac{1}{m}\mathsf{E}_p\left[\sum_{i=1}^{m} \mathbf{1}[x^{(i)} = 1]\right] \;=\; \frac{1}{m}\sum_{i=1}^{m} \mathsf{E}_p\big[\mathbf{1}[x^{(i)} = 1]\big] \;=\; p.$$

Furthermore, we have for all $i \in [m]$ that

$$\begin{aligned}\mathsf{Var}_p\big[\mathbf{1}[x^{(i)} = 1]\big] \;&=\; \mathsf{E}_p\big[\mathbf{1}[x^{(i)} = 1]^2\big] - \mathsf{E}_p\big[\mathbf{1}[x^{(i)} = 1]\big]^2 \\ &=\; \mathsf{E}_p\big[\mathbf{1}[x^{(i)} = 1]\big] - p^2 \;=\; p - p^2 \;=\; p(1-p).\end{aligned}$$

Therefore, by setting $X_i = \mathbf{1}[x^{(i)} = 1]$ we get from the weak law of large numbers that

$$\mathsf{P}\Big[\big|T_{ML}(X^{(m)}) - p\big| \geq \varepsilon\Big] \;\leq\; \frac{p(1-p)}{m\varepsilon^2}$$

which goes to zero as m goes to infinity. Hence, the maximum likelihood estimator T_{ML} for the Bernoulli parameter p is (weakly) consistent.

Now, let p be a multivariate categorical. The maximum likelihood estimator for the vector $\mathbf{p} = \big(p(x)\big)_{x\in\mathcal{X}}$ of natural parameters assigns to any sequence $X^{(m)} \in \mathcal{X}^+$ of points $x^{(1)}, \ldots, x^{(m)}$ the parameter vector

$$T_{ML}(X^{(m)}) \;=\; \frac{1}{m}\sum_{i=1}^{m} \Big(\mathbf{1}[x^{(i)} = x]\Big)_{x\in\mathcal{X}} \;\in\; \Delta_{\mathcal{X}},$$

Since all the random variables

$$\mathbf{1}[x^{(i)} = x], \quad x \in \mathcal{X},\, i \in [m]$$

are Bernoulli variables, we can reuse our results for Bernoulli variables and get that

$$\mathsf{E}_{\mathbf{p}}\big[T_{ML}(X^{(m)})\big] \;=\; \mathbf{p},$$

and

$$\mathsf{Var}_{\mathbf{p}}\Big[\Big(\mathbf{1}[x^{(i)} = x]\Big)_{x\in\mathcal{X}}\Big] \;=\; \Big(p(x)(1-p(x))\Big)_{x\in\mathcal{X}} \qquad \text{for all } i \in [m].$$

Thus, by the weak law of large numbers and a union bound argument we get for the maximum norm

$$\mathsf{P}\Big[\big\|T_{ML}(X^{(m)}) - \mathbf{p}\big\|_\infty \geq \varepsilon\Big] \;\leq\; \frac{|\mathcal{X}|\max_{x\in\mathcal{X}} p(x)(1-p(x))}{m\varepsilon^2} \;\leq\; \frac{|\mathcal{X}|}{4m\varepsilon^2},$$

which goes to zero as m goes to infinity. Hence, the maximum likelihood estimator T_{ML} is (weakly) consistent. It is easy to see that T_{MAP} is also (weakly) consistent.

Chapter 8

Iterative Proportional Scaling

Iterative proportional scaling is an algorithm for computing a maximum likelihood estimate of the natural parameters of a positive multivariate categorical that factorizes over a set of multi-index sets. When the interaction (conditional dependence) graph is known, the algorithm can be used to learn natural parameters $\mathbf{p} = \big(p(x)\big)_{x\in\mathcal{X}}$ that factorize as

$$p(x) \;=\; \prod_{I\in\mathcal{I}} a_I(x_I), \text{ where } \mathcal{I} \subset 2^{[n]},$$

and that maximize the likelihood function $L(\mathbf{p}) = \prod_{i=1}^{m} p(x^{(i)}) = \prod_{x\in\mathcal{X}} p(x)^{m(x)}$, where $x^{(1)}, \ldots, x^{(m)} \in \mathcal{X}$ are the given data points and $m(x)$ is the count of the observations of $x \in \mathcal{X}$.

We denote the set of all probability distributions that factorizes as above as $\mathcal{P}$. Note that $\mathcal{P}$ is not empty, because it contains the distribution $p(x) = 1/|\mathcal{X}|, x \in \mathcal{X}$, that is, the uniform distribution. The uniform distribution can be obtained by setting $a_I(x_I) = c$ for all $I \in \mathcal{I}$ and all $x_I \in \hat{\mathcal{X}}_I$ such that

$$1 \;=\; \sum_{x\in\mathcal{X}} p(x) \;=\; \sum_{x\in\mathcal{X}} \prod_{I\in\mathcal{I}} a_I(x_I) \;=\; \sum_{x\in\mathcal{X}} \prod_{I\in\mathcal{I}} c \;=\; \sum_{x\in\mathcal{X}} c^{|\mathcal{I}|} \;=\; |\mathcal{X}|c^{|\mathcal{I}|},$$

which implies $c = \exp\big(-\log(|\mathcal{X}|)/|\mathcal{I}|\big)$. To make sure that a maximum likelihood solution exists we maximize the likelihood over the compact closure $\overline{\mathcal{P}}$ of $\mathcal{P}$, where $p \in \overline{\mathcal{P}}$, if and only if there exists a sequence $(p_i), p_i \in \mathcal{P}$ such that

$$p(x) \;=\; \lim_{i\to\infty} p_i(x) \quad \text{ for all } x \in \mathcal{X}.$$

In fact, we can optimize over the data dependent set

$$\mathcal{P}^\star \;=\; \Big\{p \in \overline{\mathcal{P}} \,:\, m(x) > 0 \Rightarrow p(x) > 0\Big\},$$

where we only consider distributions that are strictly positive for any $x \in \mathcal{X}$ that has been actually observed. Note that, of course, the distribution p from the data have been generated must have this property. The condition $m(x) > 0 \Rightarrow p(x) > 0$ is logically equivalent to the practically more important condition

$$p(x) = 0 \;\Rightarrow\; m(x) = 0.$$

Restricting ourselves to $\mathcal{P}^\star$ is not a restriction, because for $\mathbf{p}_{ML} = \text{argmax}_{\mathbf{p} \in \overline{\mathcal{P}}} L(\mathbf{p})$, $p_{ML}(x) = 0$ implies $m(x) = 0$ (and thus $p_{ML}(x)^{m(x)} = 1$). Otherwise, we would have $L(\mathbf{p}_{ML}) = 0$, but then $\mathbf{p}_{ML}$ cannot maximize the likelihood function, because the uniform distribution, for example, has a positive likelihood.

Adjusting a Marginal

Adjusting a marginal is the core operation that needs to be implemented in the iterative proportional scaling algorithm. The operation is defined for every $I \in \mathcal{I}$ and the iterative proportional scaling algorithm iteratively applies the operations for the different $I \in \mathcal{I}$ one after the other in a loop until convergence is reached. The *adjusting a marginal* operation is a mapping $T_I : \mathcal{P}^\star \ni p \mapsto T_I p$, where

$$T_I p(x) = \begin{cases} \frac{m(x_I)}{m} \frac{p(x)}{p(x_I)} = \frac{m(x_I)/m}{p(x_I)} p(x) & : \quad p(x_I) > 0 \\ 0 & : \quad p(x_I) = 0 \end{cases}$$

that, as we will see below, maps distributions in $\mathcal{P}^\star$ onto distributions in $\mathcal{P}^\star$. In the following we only write $T_I p(x) = \frac{m(x_I)/m}{p(x_I)} p(x)$ and implicitly assume that this expression is 0 whenever $p(x_I) = 0$.

Lemma 6. *The adjusting a marginal operator T_I has the following properties:*

1. *$T_I p$ is a probability distribution on $\mathcal{X}$ that factorizes over $\mathcal{I}$.*
2. *T_I is continuous on $\mathcal{P}^\star$.*
3. *$T_I(\mathcal{P}^\star) \subseteq \mathcal{P}^\star$.*
4. *$T_I p(y) = \frac{m(y)}{m}$ for all $y \in \mathcal{X}_I$ and all $p \in \mathcal{P}^\star$.*
5. *$L(T_I p) \geq L(p)$, with equality if and only if $p(y) = m(y)/m$ for all $y \in \mathcal{X}_I$, that is, $T_I p = p$.*

Proof. 1. $T_I p$ factorizes over $\mathcal{I}$, because p factorizes over $\mathcal{I}$ and the multiplication with $b_I(x_I) := \frac{m(x_I)/m}{p(x_I)}$ just changes the factor $a_I(x_I)$ into $a_I(x_I) b_I(x_I)$. Moreover,

$$T_I p(x) = \frac{m(x_I)/m}{p(x_I)} p(x) \geq 0 \quad \text{for all } x \in \mathcal{X}.$$

Thus it remains to show that $\sum_{x\in\mathcal{X}} T_I p(x) = 1$. We have

$$\sum_{x\in\mathcal{X}} T_I p(x) \;=\; \sum_{x\in\mathcal{X}} \frac{m(x_I)/m}{p(x_I)} p(x) \;=\; \sum_{y\in\mathcal{X}_I} \frac{m(y)/m}{p(y)} \Big(\sum_{x\in\mathcal{X}:\, x_I = y} p(x) \Big)$$
$$= \sum_{y\in\mathcal{X}_I} \frac{m(y)/m}{p(y)} p(y) \;=\; \frac{1}{m} \sum_{y\in\mathcal{X}_I} m(y) \;=\; 1.$$

Hence, $T_I p$ is a probability distribution.

2. Let (p_i) be a sequence in $\mathcal{P}^\star$ and let $\lim_{i\to\infty} p_i = p \in \mathcal{P}^\star$. Using that $p(x_I) = 0$ implies that $m(x_I) = 0$ for $p \in \mathcal{P}^\star$ we get

$$\lim_{i\to\infty} T_I p_i(x) \;=\; \lim_{i\to\infty} p_i(x) \frac{m(x_I)/m}{p_i(x_I)}$$
$$= \begin{cases} p(x)\frac{m(x_I)/m}{p(x_I)} & : \quad p(x_I) > 0 \\ 0 & : \quad p(x_I) = 0 \end{cases}$$
$$= T_I p(x)$$

for all $x \in \mathcal{X}$. Hence,

$$\lim_{i\to\infty} T_I p_i \;=\; T_I p \;=\; T_I\Big(\lim_{i\to\infty} p_i \Big),$$

which means that T_I is continuous on $\mathcal{P}^\star$.

3. Let $\varepsilon > 0$ and define

$$q_\varepsilon(x) \;=\; \frac{m(x) + \varepsilon}{m + \varepsilon|\mathcal{X}|} \quad \text{for all} \quad x \in \mathcal{X}.$$

Then $q_\varepsilon(x) > 0$ for all $x \in \mathcal{X}$. For $p \in \mathcal{P}$ let

$$p_\varepsilon(x) \;=\; \frac{q_\varepsilon(x_I)}{p(x_I)} p(x) \;=:\; b_I(x_I) p(x) \quad \text{for all} \quad x \in \mathcal{X}.$$

Then we have

$$T_I p(x) \;=\; \lim_{\varepsilon\to 0} p_\varepsilon(x).$$

If we can show that $p_\varepsilon \in \mathcal{P}$, then it follows immediately that $T_I p \in \overline{\mathcal{P}}$, that is, $T_I(\mathcal{P}) \subseteq \overline{\mathcal{P}}$. Obviously, p_ε is positive and factorizes over $\mathcal{I}$. Thus, it remains to

show that p_ε is a probability measure on $\mathcal{X}$, which follows from

$$\begin{aligned}
\sum_{x\in\mathcal{X}} p_\varepsilon(x) &= \sum_{x\in\mathcal{X}} \frac{q_\varepsilon(x_I)}{p(x_I)} p(x) = \sum_{y\in\mathcal{X}_I} \frac{q_\varepsilon(y)}{p(y)} \left(\sum_{x\in\mathcal{X}:\, x_I=y} p(x) \right) \\
&= \sum_{y\in\mathcal{X}_I} \frac{q_\varepsilon(y)}{p(y)} p(y) = \sum_{y\in\mathcal{X}_I} q_\varepsilon(y) = \sum_{y\in\mathcal{X}_I} \sum_{x\in\mathcal{X}:\, x_I=y} q_\varepsilon(x) \\
&= \sum_{x\in\mathcal{X}} q_\varepsilon(x) = \sum_{x\in\mathcal{X}} \frac{m(x)+\varepsilon}{m+\varepsilon|\mathcal{X}|} \\
&= \frac{1}{m+\varepsilon|\mathcal{X}|} \sum_{x\in\mathcal{X}} \big(m(x)+\varepsilon\big) \\
&= \frac{m+\varepsilon|\mathcal{X}|}{m+\varepsilon|\mathcal{X}|} = 1.
\end{aligned}$$

Therefore, we have, by the continuity of T_I on $\mathcal{P}^\star$, for any sequence (p_i) in $\mathcal{P} \subseteq \mathcal{P}^\star$ with $\lim_{i\to\infty} p_i = p \in \mathcal{P}^\star$ that

$$\lim_{i\to\infty} T_I p_i = T_I p \in \mathcal{P}^\star,$$

and thus $T_I(\mathcal{P}^\star) \subseteq \overline{\mathcal{P}}$. In order to show that $T_I(\mathcal{P}^\star) \subseteq \mathcal{P}^\star$ it suffices to show that $T_I p(x) = 0$ implies $m(x) = 0$ for all $p \in \mathcal{P}^\star$. Since

$$T_I p(x) = \frac{m(x_I)/m}{p(x_I)} p(x)$$

$T_I p(x) = 0$ implies $p(x) = 0$ or $m(x_I) = 0$. In the first case we get $m(x) = 0$ since $p \in \mathcal{P}^\star$, and in the second case we get $m(x) = 0$ since $m(x) \le m(x_I) = 0$. Thus $T_I p \in \mathcal{P}^\star$.

4. Let $y \in \mathcal{X}_I$ and $p \in \mathcal{P}^\star$. We have for the marginal distribution on $\mathcal{X}_I$ that

$$\begin{aligned}
T_I p(y) &= \sum_{x\in\mathcal{X}:\, x_I=y} T_I p(x) = \sum_{x\in\mathcal{X}:\, x_I=y} \frac{m(x_I)/m}{p(x_I)} p(x) = \sum_{x\in\mathcal{X}:\, x_I=y} \frac{m(y)/m}{p(y)} p(x) \\
&= \frac{m(y)/m}{p(y)} \sum_{x\in\mathcal{X}:\, x_I=y} p(x) = \frac{m(y)/m}{p(y)} p(y) = \frac{m(y)}{m}.
\end{aligned}$$

5. We have

$$\begin{aligned}
L(T_I\mathbf{p}) &= \prod_{x\in\mathcal{X}} \left(\frac{m(x_I)/m}{p(x_I)}p(x)\right)^{m(x)} \\
&= \left(\prod_{x\in\mathcal{X}} p(x)^{m(x)}\right)\left(\prod_{x\in\mathcal{X}} \left(\frac{m(x_I)/m}{p(x_I)}\right)^{m(x)}\right) \\
&= L(\mathbf{p}) \prod_{x\in\mathcal{X}} \left(\frac{m(x_I)/m}{p(x_I)}\right)^{m(x)} \\
&= L(\mathbf{p}) \prod_{y\in\mathcal{X}_I} \left(\frac{m(y)/m}{p(y)}\right)^{\sum_{x\in\mathcal{X}:\, x_I=y} m(x)} \\
&= L(\mathbf{p}) \prod_{y\in\mathcal{X}_I} \left(\frac{m(y)/m}{p(y)}\right)^{m(y)} \\
&= L(\mathbf{p}) \frac{\prod_{y\in\mathcal{X}_I} \left(\frac{m(y)}{m}\right)^{m(y)}}{\prod_{y\in\mathcal{X}_I} p(y)^{m(y)}} \;\geq\; L(\mathbf{p}),
\end{aligned}$$

where the inequality follows from the fact that $\big(m(y)/m\big)_{y\in\mathcal{X}_I}$ is the non-factorization-constrained maximum likelihood estimate for a categorical distribution on $\mathcal{X}_I$ given the observations $\big(m(y)\big)_{y\in\mathcal{X}_I}$ with $\sum_{y\in\mathcal{X}_v} m(y) = m$. Equality only holds if $p(y) = m(y)/m$ for all $y \in \mathcal{X}_I$. □

Iterative Proportional Scaling (IPS) algorithm

Lemma 7. *The maximum likelihood estimate* $\mathbf{p}_{ML} = \text{argmax}_{p\in\mathcal{P}^\star}\, L(\mathbf{p})$ *is a solution to the system of equations*

$$p(y) \;=\; \frac{m(y)}{m}$$

for all $I \in \mathcal{I}$ *and all* $y \in \mathcal{X}_I$.

Proof. We first show that $\mathbf{p}_{ML}$ satisfies the system of equations. By Lemma 6 (Property 5) we have for all $I \in \mathcal{I}$ that $L(T_I\mathbf{p}_{ML}) \geq L(\mathbf{p}_{ML})$. On the other hand we also have $L(\mathbf{p}_{ML}) \geq L(T_I\mathbf{p}_{ML})$ since $\mathbf{p}_{ML}$ maximizes the likelihood function. Thus, $L(T_I\mathbf{p}_{ML}) = L(\mathbf{p}_{ML})$, which implies $p_{ML}(y) = m(y)/m$ for all $y \in \mathcal{X}_I$. That is, $\mathbf{p}_{ML}$ satisfies the system of equations.

It remains to show that any solution $\hat{p} \in \mathcal{P}^\star$ of the system of equations maximizes the likelihood function. By using $m(y) = m\hat{p}(y)$, we have for any $p \in \mathcal{P}^\star$ that

$$\begin{aligned}
L(\mathbf{p}) &= \prod_{x \in \mathcal{X}} p(x)^{m(x)} = \prod_{x \in \mathcal{X}} \Big(\prod_{I \in \mathcal{I}} a_I(x_I) \Big)^{m(x)} \\
&= \prod_{x \in \mathcal{X}} \Big(\prod_{I \in \mathcal{I}} a_I(x_I)^{m(x)} \Big) = \prod_{I \in \mathcal{I}} \Big(\prod_{y \in \mathcal{X}_I} a_I(y)^{\sum_{x \in \mathcal{X}:\, x_I = y} m(x)} \Big) \\
&= \prod_{I \in \mathcal{I}} \Big(\prod_{y \in \mathcal{X}_I} a_I(y)^{m(y)} \Big) = \prod_{I \in \mathcal{I}} \Big(\prod_{y \in \mathcal{X}_I} a_I(y)^{m\hat{p}(y)} \Big) \\
&= \prod_{I \in \mathcal{I}} \Big(\prod_{y \in \mathcal{X}_I} a_I(y)^{m \sum_{x \in \mathcal{X}:\, x_I = y} \hat{p}(x)} \Big) = \prod_{x \in \mathcal{X}} \Big(\prod_{I \in \mathcal{I}} a_I(x_I)^{m\hat{p}(x)} \Big) \\
&= \prod_{x \in \mathcal{X}} \Big(\prod_{I \in \mathcal{I}} a_I(x_I) \Big)^{m\hat{p}(x)} = \prod_{x \in \mathcal{X}} p(x)^{m\hat{p}(x)}.
\end{aligned}$$

Therefore, we get for all $p \in \mathcal{P}^\star$ that

$$L(\hat{\mathbf{p}}) \;=\; \prod_{x \in \mathcal{X}} \hat{p}(x)^{m\hat{p}(x)} \;\geq\; \prod_{x \in \mathcal{X}} p(x)^{m\hat{p}(x)} \;=\; L(\mathbf{p}),$$

where the inequality follows from the fact that $\hat{\mathbf{p}}$ is the non-factorization-constrained maximum likelihood estimate of a multivariate categorical on $\mathcal{X}$ given the "observations" $\big(m\hat{p}(x)\big)_{x \in \mathcal{X}}$. That is, $\hat{\mathbf{p}}$ maximizes the likelihood function. □

For solving the system of equations, the iterative proportional scaling algorithm iteratively applies all the adjusting a marginal operations one after the other in a loop. As stated in the following theorem, the algorithm converges to a maximum likelihood estimate that satisfies the factorization constraint.

Theorem 4. *Let $I_1, \ldots, I_k$ be the elements of $\mathcal{I}$ (in arbitrary order) and define*

$$S \;=\; T_{I_k} \circ T_{I_{k-1}} \circ \ldots \circ T_{I_1}.$$

Let p_0 be the uniform distribution $p_0(x) = 1/|\mathcal{X}|$ for all $x \in \mathcal{X}$, and $p_i = S^i p_0$. Then it holds that

$$\lim_{i \to \infty} L\big(\mathbf{p}_i\big) \;=\; \max_{\mathbf{p} \in \mathcal{P}^\star} L(\mathbf{p}),$$

that is, the likelihood of the sequence converges to maximum likelihood value.

Proof. We have $p_0 \in \mathcal{P}$ and, by Lemma 6 (Property 3), that $p_i \in \mathcal{P}^\star$ for all $i \in \mathcal{N}$. By Lemma 6 (Property 5), we inductively have that

$$L(\mathbf{p}_i) \;=\; L(S\mathbf{p}_{i-1}) \;\geq\; L(\mathbf{p}_{i-1}) \;\geq\; \ldots \;\geq\; L(\mathbf{p}_0) \;>\; 0,$$

for all $i \in \mathbb{N}$. Since $\overline{\mathcal{P}}$ is compact the sequence (p_i) has a convergent subsequence (p_{i_j}) in $\overline{\mathcal{P}}$. Let $p = \lim_{j\to\infty} p_{i_j}$. By the continuity of L we have

$$L(\mathbf{p}) \;\geq\; L(\mathbf{p}_0) \;>\; 0.$$

Therefore, we can assume that $p \in \mathcal{P}^\star$, because otherwise there exists $x \in \mathcal{X}$ with $m(x) > 0$ and $p(x) = 0$, which implies $L(\mathbf{p}) = 0$, a contradiction.
Now, by the continuity of L and S, and by Lemma 6 (Property 5) we have

$$\begin{aligned} L(S\mathbf{p}) \;&=\; L\Big(S \lim_{j\to\infty} \mathbf{p}_{i_j}\Big) \;=\; L\Big(\lim_{j\to\infty} S\mathbf{p}_{i_j}\Big) \;=\; \lim_{j\to\infty} L\big(S\mathbf{p}_{i_j}\big) \\ &=\; \lim_{j\to\infty} L\big(S(S^{i_j}\mathbf{p}_0)\big) \;=\; \lim_{j\to\infty} L\big(S^{i_j+1}\mathbf{p}_0\big) \;\leq\; \lim_{j\to\infty} L\big(S^{i_{j+1}}\mathbf{p}_0\big) \\ &=\; L\Big(\lim_{j\to\infty} S^{i_{j+1}}\mathbf{p}_0\Big) \;=\; L\Big(\lim_{j\to\infty} \mathbf{p}_{i_{j+1}}\Big) \;=\; L(\mathbf{p}). \end{aligned}$$

On the other hand, we have also from Lemma 6 (Property 5) that $L(S\mathbf{p}) \geq L(\mathbf{p})$. Thus, we have $L(S\mathbf{p}) = L(\mathbf{p})$. Then we also have $L(T_{I_i}\mathbf{p}) = L(\mathbf{p})$ for all $i \in [k]$, because $L(T_{I_i}\mathbf{p}) > L(\mathbf{p})$ would imply that $L(S\mathbf{p}) > L(\mathbf{p})$. That is, $\mathbf{p}$ satisfies the system of marginal equations

$$p(y) \;=\; \frac{m(y)}{m} \qquad \text{for all} \qquad y \in \mathcal{X}_{I_i},\, i \in [k].$$

Note that this holds for any convergent subsequence of (p_i). For a contradiction, assume now that $L\big(\mathbf{p}_i\big)$ does not converge to $\max_{\mathbf{p}\in\mathcal{P}^\star} L(\mathbf{p})$. Then there exists $\varepsilon > 0$ and for every $i \in \mathbb{N}$ there exists $j_i \geq i$ such that

$$\big|L(\mathbf{p}_{j_i}) - \max_{\mathbf{p}\in\mathcal{P}^\star} L(\mathbf{p})\big| \;\geq \varepsilon.$$

Again, by the compactness of $\overline{\mathcal{P}}$, the sequence (p_{j_i}) has a convergent subsequence that converges to an element in $\mathcal{P}^\star$. Thus, there exists a convergent subsequence of (p_i) such that $\big(L(\mathbf{p}_i)\big)$ does not converge to $\max_{\mathbf{p}\in\mathcal{P}^\star} L(\mathbf{p})$, but the limit satisfies the system of marginal equations. This is a contradiction to Lemma 7, which states that any solution of the system of marginal equations is an element of $\operatorname{argmax}_{\mathbf{p}\in\mathcal{P}^\star}$. Therefore, we must have $\lim_{i\to\infty} L\big(\mathbf{p}_i\big) = \max_{\mathbf{p}\in\mathcal{P}^\star} L(\mathbf{p})$. $\square$

Remark: The proof does not guarantee that the sequence (p_i) converges to a single element in $\operatorname{argmax}_{\mathbf{p}\in\mathcal{P}^\star} L(\mathbf{p})$, but only that the likelihood values converge to the maximum likelihood value. However, when we iteratively compute the sequence (p_i) one element after the other, then, once the likelihood value has converged, that is, satisfies some convergence criterion, although the sequence itself has not converged, we can just stop the algorithm and output the last computed element to get a maximum likelihood estimate.

Finally, note that the running time of the iterative proportional scaling algorithm is in $\Omega\big(|\mathcal{X}\big)$, because it computes all the natural parameters $\mathbf{p} = \big(p(x)\big)_{x\in\mathcal{X}}$. It is practically feasible only for small and medium sized models.

Chapter 9

Maximum Entropy Principle

Here, we use the maximum entropy principle for parameter and structure learning of multivariate categoricals. The maximum entropy principle was popularized by E.T. Jaynes:

> ... in making inferences on the basis of partial information we must use that probability distribution which has maximum entropy subject to whatever is known. This is the only unbiased assignment we can make; to use any other would amount to arbitrary assumption of information which by hypothesis we not have. [E.T. Jaynes, 1957]

Sufficient Statistics

For invoking the maximum entropy principle, once again, we slightly rewrite the interaction parameters of a multivariate categorical as

$$q_I(\hat{x}_I) \;=\; \sum_{x \in \mathcal{X}} q_{\hat{x}_I} \varphi_{\hat{x}_I}(x),$$

where the $q_{\hat{x}_I} \in \mathbb{R}$ are (identifiable) interaction parameters and the functions

$$\varphi_{\hat{x}_I}(x) \;=\; \mathbf{1}[x_I = \hat{x}_I]$$

for $\hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I} \subset 2^{[n]}$ are called *sufficient statistic*.

Sufficient statistics. Let $\mathcal{P} = \{p_\theta \,:\, \theta \in \Theta\}$ be a parameterized family of probability distributions on the sample space $\mathcal{X}$, and let

$$\varphi : \mathcal{X} \to \mathbb{R}^k,\, x \mapsto \big(\varphi_1(x), \ldots, \varphi_k(x)\big).$$

The function φ is called a *sufficient statistic* for θ, if the conditional probability distribution

$$p_\theta(x \mid \varphi(x)) \;=\; \frac{p_\theta(x, \varphi(x))}{p_\theta(\varphi(x))} \;=\; \frac{p_\theta(x)}{p_\theta(\varphi(x))},$$

is independent of θ, that is, $p_\theta(x \mid \varphi(x)) = p(x \mid \varphi(x))$. Note that we have $p_\theta(x, \varphi(x)) = p_\theta(x)$, because $\varphi(x)$ is determined by x. However, φ is not necessarily injective, that is, knowing the value of $\varphi(x)$ does not necessarily mean also knowing the value of x. The interesting aspect of sufficient statistics is, that we can compute the same maximum likelihood estimate for θ from $\varphi(x^{(1)}), \ldots, \varphi(x^{(m)})$ as from the data points $x^{(1)}, \ldots, x^{(m)}$ that have been sampled from p_θ. The maximum likelihood estimate of θ given the data points is computed as follows

$$\begin{aligned}
\theta_{ML} \;&=\; \mathrm{argmax}_{\theta\in\Theta} \prod_{j=1}^{m} p_\theta\big(x^{(j)}\big) \\
&=\; \mathrm{argmax}_{\theta\in\Theta} \prod_{j=1}^{m} p_\theta\big(\varphi(x^{(j)})\big) p\big(x^{(j)} \mid \varphi(x^{(j)})\big) \\
&=\; \mathrm{argmax}_{\theta\in\Theta} \Big(\prod_{j=1}^{m} p\big(x^{(j)} \mid \varphi(x^{(j)})\big) \Big) \Big(\prod_{j=1}^{m} p_\theta\big(\varphi(x^{(j)})\big) \Big) \\
&=\; \mathrm{argmax}_{\theta\in\Theta} \prod_{j=1}^{m} p_\theta\big(\varphi(x^{(j)})\big).
\end{aligned}$$

Sufficient statistics have been characterized by Fisher and Neyman.

Theorem 5. [Fisher-Neyman Factorization Theorem] *The function*

$$\varphi : \mathcal{X} \to \mathbb{R}^k$$

is a sufficient statistic for θ, if and only if there exists functions $g_\theta : \mathbb{R}^k \to [0, \infty)$ and $h : \mathcal{X} \to [0, \infty)$ (that does not depend on θ) such that

$$p_\theta(x) \;=\; g_\theta(\varphi(x))\, h(x)$$

for all $x \in \mathcal{X}$ and all $\theta \in \Theta$.

Proof. First, assume that p_θ satisfies the factorization property. For $x \in \mathcal{X}$ let

$$\mathcal{X}_{\varphi(x)} \;=\; \varphi^{-1}\big(\varphi(x)\big) \;=\; \big\{\hat{x} \in \mathcal{X} \,:\, \varphi(\hat{x}) = \varphi(x)\big\}.$$

We compute

$$\begin{aligned} p_\theta(\varphi(x)) \;&=\; \sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} p_\theta(\hat{x}) \;=\; \sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} g_\theta(\varphi(\hat{x}))h(\hat{x}) \\ &=\; \sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} g_\theta(\varphi(x))h(\hat{x}) \;=\; g_\theta(\varphi(x)) \sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} h(\hat{x}), \end{aligned}$$

and get that

$$p_\theta(x \mid \varphi(x)) \;=\; \frac{p_\theta(x)}{p_\theta(\varphi(x))} \;=\; \frac{g_\theta(\varphi(x))\,h(x)}{g_\theta(\varphi(x))\sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} h(\hat{x})} \;=\; \frac{h(x)}{\sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} h(\hat{x})},$$

which is independent of θ. Therefore, we can conclude that φ is a sufficient statistic.

For the other direction, suppose that φ is a sufficient statistic. Then

$$h(x) \;:=\; p(x \mid \varphi(x)) \;=\; \frac{p_\theta(x)}{p_\theta(\varphi(x))} \;=\; \frac{p_\theta(x)}{\sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} p_\theta(\hat{x})}$$

does not depend on θ. Hence, $p_\theta(x) = g_\theta(\varphi(x))\,h(x)$, if we set

$$g_\theta(y) \;=\; \sum_{\hat{x}\in\mathcal{X}_y} p_\theta(\hat{x}) \quad \text{for } y \in \mathbb{R}^k,$$

because then $g_\theta(\varphi(x)) = \sum_{\hat{x}\in\mathcal{X}_{\varphi(x)}} p_\theta(\hat{x})$. □

For multivariate categoricals

$$p(x) \;=\; \exp\left(\sum_{\hat{x}_I\in\hat{\mathcal{X}}_I\; I\in\mathcal{I}} q_{\hat{x}_I}\varphi_{\hat{x}_I}(x) \;-\; a(\mathbf{q})\right)$$

we get from the Fisher-Neyman Factorization Theorem that $\boldsymbol{\varphi} = \left(\varphi_{\hat{x}_I}\right)_{\hat{x}_I\in\hat{\mathcal{X}}_I\; I\in\mathcal{I}}$ is a sufficient statistics for the parameter vector $\mathbf{q} = \left(q_{\hat{x}_I}\right)_{\hat{x}_I\in\hat{\mathcal{X}}_I\; I\in\mathcal{I}}$ if we set $h(x) = 1$ for all $x \in \mathcal{X}$ and

$$g_{\mathbf{q}} : \mathbb{R}^k \to [0,\infty),\; y \mapsto \exp\left(\mathbf{q}^\top y - a(\mathbf{q})\right),$$

where k is the number of parameters.

The Maximum Entropy Problem

The maximum entropy principle suggest an alternative approach for parameter estimation from data points that are encoded by sufficient statistics, namely, among all distributions where the expected sufficient statistics matches their sample means choose the distribution that maximizes the entropy. The entropy

$$\mathsf{H}(p) \;=\; -\sum_{x\in\mathcal{X}} p(x)\log p(x) \;=\; -\mathsf{E}_p\big[\log \mathbf{p}\big]$$

of the distribution p, which is represented by the parameter vector $\mathbf{p}$, is a measure of the *uncertainty* of p. The sample means of the sufficient statistics $\varphi_{\hat{x}_I}, I \in \mathcal{X}_I, I \in \mathcal{I}$ for data points $x^{(1)}, \ldots, x^{(m)}$ drawn independently from p are given as

$$\mu_{\hat{x}_I} \;=\; \frac{1}{m}\sum_{j=1}^{m} \varphi_{\hat{x}_I}\big(x^{(j)}\big).$$

The *maximum entropy estimate* is given as the solution of the constrained optimization problem

$$\begin{array}{ll} \max_{\mathbf{p}\in\mathbb{R}^{|\mathcal{X}|}} & \mathsf{H}(p) \\ \text{subject to} & \mathsf{E}_p[\varphi_{\hat{x}_I}(x)] \;=\; \mu_{\hat{x}_I},\ \hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I} \\ & \sum_{x\in\mathcal{X}} p(x) \;=\; 1 \;=:\; \mu_0. \end{array}$$

In the following it turns out to be slightly more convenient to work with the equivalent problem, where we minimize the negative entropy under the same constraints.

Maximum Entropy – Maximum Likelihood Duality

The Lagrangian dual of the negative entropy minimization problem is the optimization problem

$$\max_{\boldsymbol{\theta}} \min_{\mathbf{p}} \; L(\mathbf{p}, \boldsymbol{\theta}),$$

where the Lagrangian of the negative entropy minimization problem is defined as

$$\begin{aligned} L(\mathbf{p},\boldsymbol{\theta}) &= -\mathsf{H}(p) + \theta_0\bigg(\sum_{x\in\mathcal{X}} p(x) - 1\bigg) + \sum_{\hat{x}_I\in\hat{\mathcal{X}}_I, I\in\mathcal{I}} \theta_{\hat{x}_I}\bigg(\mu_{\hat{x}_I} - \sum_{x\in\mathcal{X}} \varphi_{\hat{x}_I}(x)p(x)\bigg) \\ &= -\mathsf{H}(p) + \theta_0\bigg(\sum_{x\in\mathcal{X}} p(x) - 1\bigg) + \boldsymbol{\theta}^\top\boldsymbol{\mu} - \boldsymbol{\theta}^\top\mathsf{E}_p\big[\boldsymbol{\varphi}\big] \\ &= -\mathsf{H}(p) + \theta_0\bigg(\sum_{x\in\mathcal{X}} p(x) - 1\bigg) + \boldsymbol{\theta}^\top\boldsymbol{\mu} - \mathsf{E}_p\big[\boldsymbol{\theta}^\top\boldsymbol{\varphi}\big], \end{aligned}$$

where the $\boldsymbol{\theta} = \big(\theta_{\hat{x}_I}\big)_{\hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I}}$ are the *Lagrange multipliers* or *dual variables*, $\boldsymbol{\mu} = \big(\mu_{\hat{x}_I}\big)_{\hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I}}$, and $\boldsymbol{\varphi} = \big(\varphi_{\hat{x}_I}\big)_{\hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I}}$ is the vector of sufficient statistics. If we assume that the negative entropy minimization problem has an optimal solution in the relative interior of the probability simplex $\Delta_{\mathcal{X}}$, then, by the *saddle point condition* (see the appendix), the derivative of the Lagrangian with respect to the primal variables vanishes, that is, we get for every $p(x) \in \mathcal{X}$ that

$$0 \;=\; \frac{\partial L(\mathbf{p}, \boldsymbol{\theta})}{\partial p(x)} \;=\; 1 + \log p(x) \;+\; \theta_0 \;-\; \sum_{\hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I}} \theta_{\hat{x}_I} \varphi_{\hat{x}_I}(x).$$

The equations are solved to

$$p(x) \;=\; \exp\Bigg(\sum_{\hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I}} \theta_{\hat{x}_I} \varphi_{\hat{x}_I}(x) - (\theta_0 + 1) \Bigg) \;=\; \exp\Big(\boldsymbol{\theta}^\top \boldsymbol{\varphi}(x) - (\theta_0 + 1) \Big).$$

The dual variables $\theta_{\hat{x}_I}$ are obviously the canoncial parameters $q_{\hat{x}_I}$. If we set $\theta_0 + 1 = a(\mathbf{q})$, then p is normalized. That is, $\sum_{x \in \mathcal{X}} p(x) = 1$ is automatically satisfied and the corresponding term can be eliminated from the Lagrangian. Plugging the normalized solution $p(x) = \exp\Big(\mathbf{q}^\top \boldsymbol{\varphi}(x) - a(\mathbf{q}) \Big)$ back into the Lagrangian gives

$$L(\mathbf{p}, \mathbf{q}) \;=\; -\mathsf{H}(p) \;+\; \mathbf{q}^\top \boldsymbol{\mu} \;-\; \mathsf{E}_p\big[\mathbf{q}^\top \boldsymbol{\varphi}\big],$$

which can be rewritten further by plugging the normalized solution for $p(x)$ also into the expression for the negative entropy, which gives

$$-\mathsf{H}(p) \;=\; \mathsf{E}_p[\log p] \;=\; \mathsf{E}_p\big[\mathbf{q}^\top \boldsymbol{\varphi}(x) - a(\mathbf{q})\big] \;=\; \mathsf{E}_p\big[\mathbf{q}^\top \boldsymbol{\varphi}(x)\big] - a(\mathbf{q}).$$

Therefore, the Lagrangian becomes $L(\mathbf{q}) = \mathbf{q}^\top \boldsymbol{\mu} - a(\mathbf{q})$. Finally, from

$$\boldsymbol{\mu} \;=\; \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{\varphi}\big(x^{(j)}\big),$$

we get that

$$\begin{aligned} L(\mathbf{q}) \;=\; \mathbf{q}^\top \boldsymbol{\mu} - a(\mathbf{q}) \;&=\; \frac{1}{m} \sum_{j=1}^{m} \Big(\mathbf{q}^\top \boldsymbol{\varphi}(x^{(j)}) - a(\mathbf{q}) \Big) \\ &=\; \frac{1}{m} \sum_{j=1}^{m} \log p_{\mathbf{q}}(x^{(j)}) \;=\; \frac{\ell(\mathbf{q})}{m}. \end{aligned}$$

Hence, the dual problem becomes the log-likelihood maximization problem

$$\mathsf{argmax}_{\mathbf{q}} \; \frac{\ell(\mathbf{q})}{m}, \qquad \text{or equivalently just} \qquad \mathsf{argmax}_{\mathbf{q}} \; \ell(\mathbf{q}).$$

Relaxed Maximum Entropy Principle

The relaxed entropy maximization problem is given as

$$\begin{array}{ll} \max_{\mathbf{p}\in\mathbb{R}^{|\mathcal{X}|}} & \mathsf{H}(p) \\ \text{subject to} & \left|\mathsf{E}_p[\varphi_{\hat{x}_I}] - \mu_{\hat{x}_I}\right| \leq c,\ \hat{x}_I \in \hat{\mathcal{X}}_I, I \in \mathcal{I} \\ & \sum_{x\in\mathcal{X}} p(x) = 1 =: \mu_0. \end{array}$$

The difference to the regular entropy maximization problem is in the constraints

$$\mathsf{E}_p[\varphi_{\hat{x}_I}] = \mu_{\hat{x}_I} \quad \text{that have been relaxed to} \quad \left|\mu_{\hat{x}_I} - \mathsf{E}_p[\varphi_{\hat{x}_I}]\right| \leq c$$

for some constant $c > 0$. The inequality constraints can be equivalently written as

$$\mu_{\hat{x}_I} - \mathsf{E}_p[\varphi_{\hat{x}_I}] \leq c \quad \text{and} \quad \mathsf{E}_p[\varphi_{\hat{x}_I}] - \mu_{\hat{x}_I} \leq c.$$

Using these constraints, the Lagrangian of the relaxed negative entropy minimization problem is given as

$$\begin{aligned} L(\mathbf{p},\mathbf{q}) = &- \mathsf{H}(p) + q_0\Bigg(\sum_{x\in\mathcal{X}} p(x) - 1\Bigg) + \sum_{\hat{x}_I\in\hat{\mathcal{X}}_I, I\in\mathcal{I}} \left(q^+_{\hat{x}_I} - q^-_{\hat{x}_I}\right)\left(\mu_{\hat{x}_I} - \mathsf{E}_p[\varphi_{\hat{x}_I}]\right) \\ &- c\sum_{\hat{x}_I}\left(q^+_{\hat{x}_I} + q^-_{\hat{x}_I}\right), \end{aligned}$$

where the $q^{\pm}_{\hat{x}_I} \geq 0$ are the dual variables (Lagrange multipliers). We can assume that in a solution of the dual problem $\max_{\mathbf{q}}\min_{\mathbf{p}} L(\mathbf{p},\mathbf{q})$ at most one of each pair of variables $q^+_{\hat{x}_I}, q^-_{\hat{x}_I}$ is non-zero, because, otherwise, we could decrease both by the same amount resulting in a solution with a larger objective value. Note that if both variables are decreased by the same amount, the value of the fourth term increases, because a smaller value gets subtracted, while the values of the other terms remain constant. Setting

$$q_{\hat{x}_I} = q^+_{\hat{x}_I} - q^-_{\hat{x}_I} \quad \text{gives} \quad |q_{\hat{x}_I}| = q^+_{\hat{x}_I} + q^-_{\hat{x}_I}$$

allows us to rewrite the Lagrangian as

$$\begin{aligned} L(\mathbf{p},\mathbf{q}) &= -\mathsf{H}(\boldsymbol{p}) + q_0\Bigg(\sum_{x\in\mathcal{X}} p(x) - 1\Bigg) \\ &\quad + \sum_{\hat{x}_I} q_{\hat{x}_I}\left(\mu_{\hat{x}_I} - \mathsf{E}_p[\varphi_{\hat{x}_I}]\right) - c\sum_{\hat{x}_I}|q_{\hat{x}_I}| \\ &= -\mathsf{H}(\boldsymbol{p}) + q_0\Bigg(\sum_{x\in\mathcal{X}} p(x) - 1\Bigg) + \mathbf{q}^\top(\boldsymbol{\mu} - \mathsf{E}_p[\boldsymbol{\varphi}]) - c\|\mathbf{q}\|_1. \end{aligned}$$

As in the non-relaxed case, we get from the saddle point condition that the derivative of Lagrangian with respect to the primal variables $\mathbf{p}$ vanishes, and thus $p(x) = \exp\Big(\mathbf{q}^\top \boldsymbol{\varphi}(x) - a(\mathbf{q})\Big)$. Plugging this expression for $p(x)$ in the Lagrangian and into the negative entropy gives, as in the non-relaxed case, that

$$L(\mathbf{q}) \;=\; \mathbf{q}^\top \boldsymbol{\mu} - a(\mathbf{q}) - c\|\mathbf{q}\|_1 \;=\; \frac{\ell(\mathbf{q})}{m} - c\|\mathbf{q}\|_1.$$

If we rescale the regularization parameter $c > 0$ by m, we get the following dual regularized maximum log-likelihood problem

$$\mathsf{max}_{\mathbf{q}} \quad \ell(\mathbf{q}) \;-\; c\|\mathbf{q}\|_1.$$

Structured Relaxation for Pairwise Categoricals

Let p be an n-variate pairwise categorical, that is,

$$p(x) \;=\; \exp\left(\frac{1}{2}\mathsf{Trace}\Big(Q\,\overline{x}\,\overline{x}^\top\Big) \;-\; a(Q)\right),$$

when using indicator encodings of $x \in \mathcal{X}$. Here, $\mathsf{Trace}\big(Q\,\overline{x}\,\overline{x}^\top\big) = Q \bullet \overline{x}\,\overline{x}^\top$, where $A \bullet B \;=\; \mathsf{Trace}(AB^\top)$ is an inner product for matrices of matching dimensions. From data points $x^{(1)}, \ldots, x^{(m)}$ that are drawn independently from p we compute

$$M \;=\; \frac{1}{m}\sum_{i=1}^{m} \overline{x}^{(i)}\,\overline{x}^{(i)\top},$$

where

$$M = \left(\begin{array}{c|c|c} M_{11} & \cdots & M_{1n} \\ \hline \vdots & \ddots & \vdots \\ \hline M_{n1} & \cdots & M_{nn} \end{array}\right) \qquad \text{with } M_{ij} \;=\; M_{ji}.$$

The structured, relaxed entropy maximization problem for estimating the parameter matrix Q and thus a Markov random field is given as

$$\begin{array}{ll} \mathsf{min}_{\mathbf{p},C} & -\mathsf{H}(p) \\ \text{subject to} & \mathsf{E}_p\big[\overline{x}\,\overline{x}_{ij}^\top\big] - M_{ij} \;\leq\; C_{ij},\; i < j \in [n] \\ & M_{ij} - \mathsf{E}_p\big[\overline{x}\,\overline{x}_{ij}^\top\big] \;\leq\; C_{ij},\; i < j \in [n] \\ & \|C_{ij}\|_F^2 \;\leq\; c^2,\; i < j \in [n] \\ & \mathsf{E}_p\big[\mathsf{diag}\big(\overline{x}\,\overline{x}_{ii}^\top\big)\big] = \mathsf{E}_p[\overline{x}_i] \;=\; M_{ii} =: \boldsymbol{\mu}_i,\; i \in [n] \\ & \sum_{x \in \mathcal{X}} p(x) \;=\; 1, \end{array}$$

where the matrix inequalities mean elementwise inequalities, the Frobenius norm of a matrix A is $\|A\|_F = \mathsf{Trace}(AA^\top)$, and $c > 0$ is again neither a primal nor dual variable, but a regularization parameter. The Frobenius norm of A is derived from the inner product $A \bullet A$, analogously to the Euclidean norm of a vector.

The Lagrangian dual of the negative entropy minimization problem is given as

$$\begin{aligned} L(\mathbf{p}, C, Q^+, Q^-, \mathbf{q}, \mathbf{r}) \;=\; & -\,\mathsf{H}(p) + q_0\Big(\sum_{x\in\mathcal{X}} p(x) \;-1\Big) \\ & + \sum_{i=1}^{n}\sum_{j>i}^{n} \big(Q_{ij}^+ - Q_{ij}^-\big) \bullet \big(M_{ij} - \mathsf{E}_p\big[\overline{x}\,\overline{x}_{ij}^\top\big]\big) \\ & - \sum_{i=1}^{n}\sum_{j>i}^{n} \big(Q_{ij}^+ + Q_{ij}^-\big) \bullet C_{ij} \;+ \sum_{i=1}^{n} \mathbf{q}_i^\top \big(\boldsymbol{\mu}_i - \mathsf{E}_p[\overline{x}_i]\big) \\ & + \sum_{i=1}^{n}\sum_{j>i}^{n} r_{ij}(\|C_{ij}\|_F^2 - c^2), \end{aligned}$$

where $Q_{ij}^+, Q_{ij}^- \geq 0, r_{ij} \geq 0, q_0$ and $\mathbf{q}_i \in \mathbb{R}^{n_i}$ are the dual variables. From the saddle point condition we get that the partial derivative of the Lagrangian with respect to the primal variables $\mathbf{p}$ vanishes, that is,

$$\log p(x) + 1 \;+\; q_0 \;-\; \sum_{i=1}^{n}\sum_{j>i}^{n} \big(Q_{ij}^+ - Q_{ij}^-\big) \bullet \overline{x}\,\overline{x}_{ij}^\top \;-\; \sum_{i=1}^{n} \mathbf{q}_i^\top \overline{x}_i \;=\; 0,$$

which solves to

$$p(x) \;=\; \exp\Bigg(\sum_{i=1}^{n}\sum_{j>i}^{n} \big(Q_{ij}^+ - Q_{ij}^-\big) \bullet \overline{x}\,\overline{x}_{ij}^\top + \sum_{i=1}^{n} \mathbf{q}_i^\top \overline{x}_i - (q_0 + 1)\Bigg).$$

where, as we have discussed before, $q_0 + 1$ should be equated with the log-normalizer $a(Q, \mathbf{q})$. Plugging the solution for $p(x)$ into the Lagrangian while using

$$-\mathsf{H}(p) \;=\; \mathsf{E}_p[\log p] \;=\; \sum_{i=1}^{n}\sum_{j>i}^{n} \mathsf{E}_p\Big[\big(Q_{ij}^+ - Q_{ij}^-\big) \bullet \overline{x}\,\overline{x}_{ij}^\top\Big] + \sum_{i=1}^{n} \mathbf{q}_i^\top \mathsf{E}_p[\overline{x}_i] - a(Q, \mathbf{q}),$$

with $Q = Q^+ - Q^-$, eliminates the primal variables $\mathbf{p}$ and gives

$$L(Q^+, Q^-, C, \mathbf{q}, \mathbf{r}) = \sum_{i=1}^{n}\sum_{j>i}^{n} \big(Q_{ij}^+ - Q_{ij}^-\big) \bullet M_{ij} + \sum_{i=1}^{n} \mathbf{q}_i^\top \boldsymbol{\mu}_i - a(Q, \boldsymbol{q}) - \sum_{i=1}^{n}\sum_{j>i}^{n} \Big(\big(Q_{ij}^+ + Q_{ij}^-\big) \bullet C_{ij} - r_{ij}\big(\|C_{ij}\|_F^2 - c^2\big)\Big),$$

where as before

$$\frac{1}{m}\ell(Q, \mathbf{q}) = \sum_{i=1}^{n}\sum_{j>i}^{n} Q_{ij} \bullet M_{ij} + \sum_{i=1}^{n} \mathbf{q}_i^\top \boldsymbol{\mu}_i - a(Q, \boldsymbol{q}),$$

with $Q_{ij} = Q_{ij}^+ - Q_{ij}^-$. Furthermore, we get again from the saddle point condition that also the derivative with respect to the primal variables C_{ij} vanishes, that is,

$$\begin{aligned} 0 &= \frac{\partial}{\partial C_{ij}} \Bigg(\sum_{i=1}^{n}\sum_{j>i}^{n} \Big(\big(Q_{ij}^+ + Q_{ij}^-\big) \bullet C_{ij} - r_{ij}\big(\|C_{ij}\|_F^2 - c^2\big)\Big)\Bigg) \\ &= Q_{ij}^+ + Q_{ij}^- - 2r_{ij}C_{ij}, \end{aligned}$$

and thus $C_{ij} = \frac{1}{2r_{ij}}(Q_{ij}^+ + Q_{ij}^-)$, which gives

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{j>i}^{n} \Big(\big(Q_{ij}^+ + Q_{ij}^-\big) \bullet C_{ij} - r_{ij}\big(\|C_{ij}\|_F^2 - c^2\big)\Big) \\ &\qquad = \sum_{i=1}^{n}\sum_{j>i}^{n} \Big(\frac{1}{2r_{ij}} \big\|Q_{ij}^+ + Q_{ij}^-\big\|_F^2 - \frac{1}{4r_{ij}} \big\|Q_{ij}^+ + Q_{ij}^-\big\|_F^2 + c^2 r_{ij}\Big) \\ &\qquad = \sum_{i=1}^{n}\sum_{j>i}^{n} \Big(\frac{1}{4r_{ij}} \big\|Q_{ij}^+ + Q_{ij}^-\big\|_F^2 + c^2 r_{ij}\Big). \end{aligned}$$

Therefore, the Lagrangian becomes the following function of the dual variables,

$$L(Q^+, Q^-, \mathbf{q}, \mathbf{r}) = \frac{1}{m}\ell(Q, \mathbf{q}) - \sum_{i=1}^{n}\sum_{j>i}^{n} \Big(\frac{1}{4r_{ij}} \big\|Q_{ij}^+ + Q_{ij}^-\big\|_F^2 + c^2 r_{ij}\Big).$$

For eliminating the dual variables $\mathbf{r}$, we are invoking the saddle point condition once more, this time for $\mathbf{r}$. Setting the derivative with respect to r_{ij} to zero gives

$$r_{ij} = \frac{\|Q_{ij}^+ + Q_{ij}^-\|_F}{2c},$$

which is, as required, non-negative, and

$$\frac{1}{4r_{ij}}\left\|Q_{ij}^{+}+Q_{ij}^{-}\right\|_F^2+c^2r_{ij} \;=\; c\left\|Q_{ij}^{+}+Q_{ij}^{-}\right\|_F.$$

Thus, the Lagrangian becomes

$$L(Q^{+},Q^{-},\mathbf{q}) \;=\; \frac{1}{m}\ell(Q,\mathbf{q}) \;-\; c\sum_{i=1}^{n}\sum_{j>i}^{n}\left\|Q_{ij}^{+}+Q_{ij}^{-}\right\|_F.$$

Finally, since $Q_{ij}^{+}+Q_{ij}^{-}=|Q|_{ij}$ we have $\left\|Q_{ij}^{+}+Q_{ij}^{-}\right\|_F=\|Q_{ij}\|_F$, from which we get

$$L(Q,\mathbf{q}) \;=\; \frac{1}{m}\ell(Q,\mathbf{q}) \;-\; c\sum_{i=1}^{n}\sum_{j>i}^{n}\left\|Q_{ij}\right\|_F.$$

The regularization term is a 1-norm of Frobenius norms. Similarly as a 1-norm regularization term induces sparsity, this term induces group sparsity by setting whole matrices Q_{ij} to zero. Sometimes, the regularization term is written in terms of a group norm. If we additionally scale the regularization parameter c by m we get the following dual $\max_{Q,\mathbf{q}} L(Q,\mathbf{q})$ of the negative entropy minimization problem

$$\max_{Q,\mathbf{q}} \quad \ell(Q,\mathbf{q}) \;-\; c\left\|Q\right\|_{1,F}.$$

Appendix: Lagrangian Duality

We consider optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \; f(x) \quad \text{subject to} \quad c_i(x) \;\leq\; 0,\; i\in[m],$$

where $f, c_i : \mathbb{R}^n \to \mathbb{R}$ are convex and continuously differentiable. The *Lagrangian* of this problem is the function

$$L : \mathbb{R}^n \times \mathbb{R}^m_{\geq 0} \to \mathbb{R},\; (x,a) \mapsto f(x) + \sum_{i=1}^{m} a_i c_i(x).$$

Let $(\hat{x},\hat{a})$ be a saddle point of the Lagrangian, i.e., we have

$$L(\hat{x},a) \;\leq\; L(\hat{x},\hat{a}) \;\leq\; L(x,\hat{a}) \;\text{ for all }\; x\in\mathbb{R}^n,\, a\in\mathbb{R}^m_{\geq 0}.$$

Note that such a saddle point does not need to exist, but if it exists then we have the following lemma.

Lemma 8. *Let $(\hat{x}, \hat{a})$ be a saddle point of the Lagrangian L. Then we have*

1. *$\hat{x}$ is a feasible solution of the constrained optimization problem.*
2. *$\hat{a}_i c_i(\hat{x}) = 0$ for all $i \in [m]$, which implies $L(\hat{x}, \hat{a}) = f(\hat{x})$.*
3. *$f(\hat{x}) \leq f(x)$ for all feasible $x \in \mathbb{R}^n$.*

Proof. 1. Assume that $c_i(\hat{x}) > 0$ for some $i \in [m]$, which renders $\hat{x}$ infeasible. Then we can increase $\hat{a}_i \geq 0$ to make $L(\hat{x}, \hat{a})$ larger. This is a contradiction to the saddle point condition, which asserts that

$$\max_{a \geq 0} L(\hat{x}, a) = L(\hat{x}, \hat{a}).$$

2. Assume for a contradiction that $\hat{a}_i c_i(\hat{x}) \neq 0$. Then we have $\hat{a}_i > 0$ and since $\hat{x}$ is feasible $c_i(\hat{x}) < 0$. Therefore can decrease $\hat{a}_i$ while keeping it non-negative to make $L(\hat{x}, \hat{a})$ larger, again in contradiction to the saddle point condition.

3. We have

$$f(\hat{x}) = L(\hat{x}, \hat{a}) \leq L(x, \hat{a}) = f(x) + \sum_{i=1}^{m} \hat{a}_i c_i(x) \leq f(x),$$

where the last inequality follows from $\hat{a}_i \geq 0$, $i \in [m]$ and the feasibility of x, i.e., we have $c_i(x) \leq 0$ for all $i \in [m]$. □

From the saddle point condition we also get

$$\begin{aligned} \max_{a \in \mathbb{R}^m_{\geq 0}} \min_{x \in \mathbb{R}^n} L(x, a) &\leq \max_{a \geq 0} L(\hat{x}, a) \\ &= L(\hat{x}, \hat{a}) \\ &= \min_{x \in \mathbb{R}^n} L(x, \hat{a}) \leq \max_{a \in \mathbb{R}^m_{\geq 0}} \min_{x \in \mathbb{R}^n} L(x, a), \end{aligned}$$

which implies

$$\max_{a \geq 0} \min_{x \in \mathbb{R}^n} L(x, a) = L(\hat{x}, \hat{a}) = f(\hat{x}).$$

Hence, the *dual optimization problem*

$$\max_{a \in \mathbb{R}^m} \min_{x \in \mathbb{R}^n} L(x, a) \quad \text{subject to} \quad a \geq 0$$

has the same optimal value as the *primal problem*. The Lagrangian is by our assumptions a convex, differentiable function. Thus we have

$$\hat{x} = \text{argmin}_{x \in \mathbb{R}^n} L(x, a) \quad \text{if and only if} \quad \nabla_x L(x, a)|_{x = \hat{x}} = 0,$$

and get the following equivalent formulation of the dual problem

$$\max_{a \in \mathbb{R}^m} L(x, a) \quad \text{subject to} \quad a \geq 0 \text{ and } \nabla_x L(x, a) = 0.$$

The existence of a saddle point can often be established by satisfying the Karush-Kuhn-Tucker condition that we state here without a proof.

Theorem 6. [Karush-Kuhn-Tucker] *Given an optimization problem*

$$\textit{min}_{x \in \mathbb{R}^n} f(x) \quad \textit{subject to} \quad c_i(x) \leq 0,\ i \in [m],$$

where $f, c_i : \mathbb{R}^n \to \mathbb{R}$ are convex and continuously differentiable. If $\hat{x} \in \mathbb{R}^n$ is an optimal solution of the primal optimization problem and there exists $x \in \mathbb{R}^n$ such that $c_(x) < 0$ for all $i \in [m]$ (Slater's condition), then there exists $\hat{a} \in \mathbb{R}^m_{\geq 0}$ such that $(\hat{x}, \hat{a})$ is a saddle point of the Lagrangian associated with the optimization problem. Moreover, there exists $i \in [m]$ such that $\hat{a}_i > 0$. □

The Karush-Kuhn-Tucker (KKT) Theorem guarantees the existence of a saddle point, if the optimization problem has an optimal solution.

We can use the KKT Theorem to establish the existence of saddle point for the negative entropy minimization problem

$$\begin{array}{ll} \max_{\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}} & -\mathsf{H}(p) \\ \text{subject to} & \mathsf{E}_p[\boldsymbol{\varphi}] = \boldsymbol{\mu} \\ & \mathbf{p} \in \Delta_{\mathcal{X}} \end{array}$$

as follows: The negative entropy function is differentiable and convex on the open set $\mathbb{R}^{|\mathcal{X}|}_{>0}$. The convexity the negative entropy function is certified by its Hessian $\text{diag}\big(\text{vec}(1) \oslash \mathbf{p}\big)$, which is positive semi-definite on $\mathbb{R}^{|\mathcal{X}|}_{>0}$. The feasible set is defined by linear, and thus differentiable and convex functions. Therefore, if the maximum entropy problem has an optimal solution in the interior of $\Delta_{\mathcal{X}}$, then, by the KKT Theorem, it has saddle point.

Appendix: Generalized Linear Models

We can build many more families of probabilities distributions, called linear exponential families, from sufficient statistics, in general. Multivariate categoricals are a just special case of an exponential family. Given a sufficient

statistics $\varphi : \mathcal{X} \to \mathbb{R}^k$, a *linear exponential family* contains all distributions whose probability density functions are of the form

$$p(x) \;=\; h(x) \exp\left(\boldsymbol{\theta}^\top \boldsymbol{\varphi}(x) - a(\boldsymbol{\theta})\right),$$

where $\boldsymbol{\theta} \in \mathbb{R}$ are *canonical parameters* and h is called the *carrier density*. Distributions from linear exponential families are also known as *log-linear models* or, in statistical physics, as *Gibbs distributions* and *Boltzmann distributions*.

Sufficient statistics can also be used to derive models for supervised learning. In fact, many of the well-known unsupervised models can be derived as generalized linear models from parameterized sufficient statistics.
In the following we discuss some specific choices for the sufficient statistics and relate the corresponding models to well-known techniques in supervised machine learning. In order to be consistent with the conventional notation in supervised machine learning, we denote the sample space by $\mathcal{Y}$, the parameter space by $\mathcal{X}$, and the parameterized probability distributions (parameterized exponential families) by

$$p(y : x) \;=\; h(y) \exp\left(\boldsymbol{\theta}^\top \boldsymbol{\varphi}(y, x) - a(\boldsymbol{\theta}, x)\right).$$

Generalized linear models are special cases of conditional, linear exponential families whose sufficient statistics is of the following form

$$\varphi_i(y : x) \;=\; \hat{\varphi}_i(y) x, \quad i \in [k],$$

where $\hat{\varphi}_i : \mathcal{Y} \to \mathbb{R}$ and $x \in \mathcal{X} = \mathbb{R}^n$. Special cases of generalized linear models lead to well-known classification and regression problems, such as, logistic regression and ordinary least squares.

Logistic regression. Let $\mathcal{Y} = \{0, 1\}$,

$$\hat{\varphi}(y) \;=\; y,$$

and $h(y) = 1$, then the corresponding general linear model has the density

$$\begin{aligned} p(y : x) \;&=\; \exp\left(\hat{\varphi}(y)\theta^\top x - a(\theta, x)\right) \\ &=\; \exp\left(y\theta^\top x - a(\theta, x)\right) \\ &=\; \frac{\exp\left(y\theta^\top x\right)}{\sum_{y' \in \{0,1\}} \exp\left(y'\theta^\top x\right)} \\ &=\; \frac{\exp\left(y\theta^\top x\right)}{1 + \exp\left(\theta^\top x\right)}. \end{aligned}$$

Estimating $\theta \in \mathbb{R}^n$ from observations $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ via the maximum likelihood principle is called *logistic regression*.

Remark: $p(y : x)$ is a Bernoulli distribution with parameters

$$p(y = 1 : x) \;=\; \phi(x) \;=\; \frac{1}{1 + \exp\big(-\theta^\top x\big)}.$$

We compute

$$\theta^\top x \;=\; \log\left(\frac{\phi(x)}{1 - \phi(x)}\right),$$

and thus

$$\begin{aligned}
\phi(x)^y (1 - \phi(x))^{1-y} \;&=\; \exp\left(y \log\left(\frac{\phi(x)}{1 - \phi(x)}\right) + \log(1 - \phi(x))\right) \\
&=\; \exp\big(y\theta^\top x - \log\big(1 + \exp(\theta^\top x)\big)\big) \\
&=\; \frac{\exp\big(y\theta^\top x\big)}{1 + \exp\big(\theta^\top x\big)} \\
&=\; p(y : x).
\end{aligned}$$

Softmax regression. We assume that $\mathcal{Y}$ is a finite set with at least two elements,

$$\hat{\varphi}_{y'}(y) \;=\; \mathbf{1}[y = y'] \quad \text{for} \quad y' \in \mathcal{Y},$$

and $h(y) = 1$, then the corresponding general linear model has the density

$$\begin{aligned}
p(y : x) \;&=\; h(y) \exp\left(\sum_{y' \in \mathcal{Y}} \hat{\varphi}(y) \theta_{y'}^\top x \;-\; a(\theta, x)\right) \\
&=\; \exp\left(\sum_{y' \in \mathcal{Y}} \mathbf{1}[y = y'] \theta_{y'}^\top x \;-\; a(\theta, x)\right) \\
&=\; \frac{\exp\left(\sum_{y' \in \mathcal{Y}} \mathbf{1}[y = y'] \theta_{y'}^\top x\right)}{\sum_{y' \in \mathcal{Y}} \exp\big(\theta_{y'}^\top x\big)}.
\end{aligned}$$

Estimating the parameter vectors $\theta_{y'} \in \mathbb{R}^n$, $y' \in \mathcal{Y}$ from observations

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$$

via the maximum likelihood principle is called *softmax regression*.

Remarks: The $p(y:x)$ are categorical distributions with canonical parameters $\log p(y:x)$, $y \in \mathcal{Y}$ (and natural parameters $p(y:x)$, $y \in \mathcal{Y}$).
A simple calculation shows, that, in the case $|\mathcal{Y}| = 2$, softmax regression reduces to logistic regression.

Ordinary least squares regression. Let $\mathcal{Y} = \mathbb{R}$ and

$$\hat{\varphi}(y) \;=\; y.$$

If we additionally assume the following carrier density function

$$h(y) \;=\; \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right),$$

then the corresponding general linear model has the density

$$\begin{aligned} p(y:x) \;&=\; h(y) \exp\left(\hat{\varphi}(y)\theta^\top x - a(\theta,x)\right) \\ &=\; \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) \exp\left(y\theta^\top x - a(\theta,x)\right) \\ &=\; \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(y-\theta^\top x\right)^2}{2}\right). \end{aligned}$$

Note, that here we have $a(\theta,x) = \left(\theta^\top x\right)^2/2$, because with this choice $p(y:x)$ becomes a standard, normalized univariate Gaussian. Estimating $\theta \in \mathbb{R}^n$ from observations $(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})$ via the maximum likelihood principle is called *ordinary least squares regression*.

Appendix: Conditional Random Fields

Again using parameterized sufficient statistics, we derive an alternative to logistic regression for binary classification. We still have the label space $\mathcal{Y} = \{0,1\}$ and the feature space $\mathcal{X} = \mathbb{R}^n$. Here, however, we assume a sufficient statistics given by the following $2d+2$ functions

$$\varphi_{y'}(y,x) \;=\; \mathbf{1}[y=y']x \qquad \text{and} \qquad \hat{\varphi}_{y'}(y) \;=\; \mathbf{1}[y=y'],$$

entails, with $h(y) \equiv 1$, the conditional exponential family

$$\begin{aligned} p(y:x) &= \exp\left(\sum_{y' \in \{0,1\}} \mathbf{1}[y=y']\theta_{y'}^\top x + \sum_{y' \in \{0,1\}} \mathbf{1}[y=y']\hat{\theta}_{y'} - a(\theta,\hat{\theta},x)\right) \\ &= \frac{\exp\left(\sum_{y' \in \{0,1\}} \mathbf{1}[y=y']\theta_{y'}^\top x + \sum_{y' \in \{0,1\}} \mathbf{1}[y=y']\hat{\theta}_{y'}\right)}{\sum_{\hat{y} \in \{0,1\}} \exp\left(\sum_{y' \in \{0,1\}} \mathbf{1}[\hat{y}=y']\theta_{y'}^\top x + \sum_{y' \in \{0,1\}} \mathbf{1}[\hat{y}=y']\hat{\theta}_{y'}\right)} \\ &= \frac{\exp\left(\theta_y^\top x + \hat{\theta}_y\right)}{\exp\left(\theta_0^\top x + \hat{\theta}_0\right) + \exp\left(\theta_1^\top x + \hat{\theta}_1\right)}. \end{aligned}$$

The model selection problem for this parameterized exponential family boils down to computing the maximum likelihood estimate of the parameters of $p(y:x)$ that we summarize into the parameter vectors $\theta \in \mathbb{R}^{2d}$ and $\hat{\theta} \in \mathbb{R}^2$. Given the observations

$$(y^{(1)}, x^{(1)}), \ldots, (y^{(m)}, x^{(m)}),$$

we get for the maximum (log-)likelihood estimate of the parameters θ and $\hat{\theta}$ that

$$\begin{aligned} (\theta,\hat{\theta})_{ML} &= \mathsf{argmax}_\theta \prod_{i=1}^m p(y^{(i)}:x^{(i)}) \\ &= \mathsf{argmax}_\theta \log \prod_{i=1}^m p(y^{(i)}:x^{(i)}) \\ &= \mathsf{argmax}_{\theta,\hat{\theta}} \log \prod_{i=1}^m \frac{\exp\left(\theta_{y^{(i)}}^\top x^{(i)} + \hat{\theta}_{y^{(i)}}\right)}{\exp\left(\theta_0^\top x^{(i)} + \hat{\theta}_0\right) + \exp\left(\theta_1^\top x^{(i)} + \hat{\theta}_1\right)} \\ &= \mathsf{argmax}_{\theta,\hat{\theta}} \sum_{i=1}^m \left(\left(\theta_{y^{(i)}}^\top x^{(i)} + \hat{\theta}_{y^{(i)}}\right) - \log\left(\sum_{y' \in \{0,1\}} \exp\left(\theta_{y'}^\top x^{(i)} + \hat{\theta}_{y'}\right)\right)\right) \\ &= \mathsf{argmax}_{\theta,\hat{\theta}} \sum_{i=1}^m \left(\left(\theta_{y^{(i)}}^\top x^{(i)} + \hat{\theta}_{y^{(i)}}\right) - \right. \\ &\qquad \left. \log\left(\exp\left(\theta_{y^{(i)}}^\top x^{(i)} + \hat{\theta}_{y^{(i)}}\right) + \exp\left(\theta_{1-y^{(i)}}^\top x^{(i)} + \hat{\theta}_{1-y^{(i)}}\right)\right)\right) \\ &=: \mathsf{argmax}_{\theta,\hat{\theta}}\ \ell(\theta,\hat{\theta}). \end{aligned}$$

Note that the log-likelihood function $\ell(\theta,\hat{\theta})$ is non-positive and concave, but not always strictly concave. To make it strictly concave we can subtract the

regularization term $\frac{1}{2c}\|\theta\|^2$, for some regularization parameter $c > 0$. This can be achieved by turning to a maximum a posteriori (MAP) estimate of θ, that is, treating θ as a random variable itself and using a suited prior distribution. If we model the prior distribution of θ by a multivariate Gaussian distribution

$$p(\theta) \sim \mathcal{N}(0, c \cdot \mathbb{1}), \text{ that is, } p(\theta) = \frac{1}{(2c\pi)^{n/2}} \exp\left(-\frac{\|\theta\|^2}{2c}\right),$$

then we get for the MAP estimate

$$\begin{aligned}
(\theta, \hat{\theta})_{MAP} &= \mathsf{argmax}_{\theta}\, p(\theta | Y : X) \\
&= \mathsf{argmax}_{\theta,\hat{\theta}}\, \frac{p(Y : X|\theta)\, p(\theta)}{p(Y : X)} \\
&= \mathsf{argmax}_{\theta,\hat{\theta}}\, \log p(Y : X|\theta) \,+\, \log p(\theta) \\
&= \mathsf{argmax}_{\theta,\hat{\theta}}\, \ell(\theta, \hat{\theta}) \,+\, \log p(\theta) \\
&= \mathsf{argmax}_{\theta,\hat{\theta}}\, \ell(\theta, \hat{\theta}) \,-\, \frac{1}{2c}\|\theta\|^2 + \frac{n}{2}\log(2c\pi) \\
&= \mathsf{argmax}_{\theta,\hat{\theta}}\, \ell(\theta, \hat{\theta}) \,-\, \frac{1}{2c}\|\theta\|^2 \\
&= \mathsf{argmin}_{\theta,\hat{\theta}} \quad \frac{1}{2}\|\theta\|^2 \,-\, c \cdot \ell(\theta, \hat{\theta}),
\end{aligned}$$

where we have used the notation $Y : X = y^{(1)} : x^{(1)}, \ldots, y^{(m)} : x^{(m)}$, the log-likelihood function (negative loss function) $-\ell(\theta, \hat{\theta}) = \log p(Y : X|\theta)$, and the Bayes theorem that states that

$$p(\theta | Y : X) \,=\, \frac{p(Y|\theta : X)\, p(\theta)}{p(Y : X)}.$$

Hinge loss and support vector machines. The negative log-likelihood function $-\ell(\theta, \hat{\theta})$ is of the form $\log\big(\exp(a) + \exp(b)\big) - a$ for $a, b \in \mathbb{R}^+$, where

$$a \,=\, \theta_{y^{(i)}}^{\top} x^{(i)} + \hat{\theta}_{y^{(i)}} \quad \text{and} \quad b \,=\, \theta_{1-y^{(i)}}^{\top} x^{(i)} + \hat{\theta}_{1-y^{(i)}}.$$

For approximating the negative log-likelihood function, we distinguish three cases:

1. $a \approx b$, then $\log\big(\exp(a) + \exp(b)\big) - a \approx \log(2\exp(a)) - a = \log(2)$, or
2. $a \gg b$, then $\log\big(\exp(a) + \exp(b)\big) - a \approx \log(\exp(a)) - a = a - a = 0$, or
3. $a \ll b$, $\log\big(\exp(a) + \exp(b)\big) - a \approx \log(\exp(b)) - a = b - a \gg 0$.

If we we do not want to make a large approximation error in the case $a \approx b$, that is, in the case that is difficult to decide, then we could use the following approximation

$$\log\big(\exp(a) + \exp(b)\big) - a \;\approx\; \mathsf{max}\{0, \log(2) + b - a\}.$$

Therefore, we can approximate the negative log-likelihood function $-\ell(\theta, \hat{\theta})$ by

$$\begin{aligned}
&\sum_{i=1}^{m} \mathsf{max}\Big\{0,\; \log(2) \;+\; \Big(\theta_{1-y^{(i)}}^{\top} x^{(i)} \;+\; \hat{\theta}_{1-y^{(i)}}\Big) \;-\; \Big(\theta_{y^{(i)}}^{\top} x^{(i)} \;+\; \hat{\theta}_{y^{(i)}}\Big)\Big\} \\
&= \sum_{i=1}^{m} \mathsf{max}\left\{0,\; \log(2) \;+\; \Big(\theta_{1-y^{(i)}} \;-\; \theta_{y^{(i)}}\Big)^{\top} x^{(i)} \;+\; \hat{\theta}_{1-y^{(i)}} \;-\; \hat{\theta}_{y^{(i)}}\right\} \\
&= \sum_{i=1}^{m} \mathsf{max}\Big\{0,\; \log(2) \;+\; \big(1 - 2y^{(i)}\big)\Big(\bar{\theta}^{\top} x^{(i)} \;+\; b\Big)\Big\},
\end{aligned}$$

where $\bar{\theta} \;=\; \theta_1 - \theta_0$ and $b \;=\; \hat{\theta}_1 - \hat{\theta}_0$. This is almost the so called *hinge loss* that is frequently used in the linear *support vector machine* (SVM) approach to binary classification. The hinge loss is given as

$$\sum_{i=1}^{m} \mathsf{max}\Big\{0,\; 1 - \big(2y^{(i)} - 1\big)\Big(\theta^{\top} x^{(i)} \;+\; b\Big)\Big\}.$$

The only difference is the scaling by a factor $\log(2)$. In the standard, regularized linear SVM, that is given as the following optimization problem

$$\mathsf{argmin}_{\theta,b}\, \frac{1}{2}\|\theta\|^2 \;+\; c \cdot \sum_{i=1}^{m} \mathsf{max}\Big\{0,\; 1 - \big(2y^{(i)} - 1\big)\Big(\theta^{\top} x^{(i)} \;+\; b\Big)\Big\},$$

the scaling factor can be absorbed into the regularization parameter c.

Multiclass classification. Let $\mathcal{Y}$ be a finite set with at least two elements and consider a sufficient statistics of the form

$$\varphi_{y'}(y, x) \;=\; \mathbf{1}[y = y']\hat{\varphi}(x)$$

for some feature function $\hat{\varphi} : \mathcal{X} \to \mathbb{R}^k$, then a simple calculation shows that the maximum likelihood estimator of the parameter vector θ, given the data points $Y \times X = \{(y^{(1)}, x^{(1)}), \ldots, (y^{(m)}, x^{(m)})\}$, is a minimum of the following loss function

$$\ell(\theta) \;=\; \sum_{i=1}^{m} \left(\log\left(\sum_{y' \in \mathcal{Y}} \exp\Big(\theta_{y'}^{\top} \hat{\varphi}\big(x^{(i)}\big)\Big)\right) \;-\; \theta_{y^{(i)}}^{\top} \hat{\varphi}\big(x^{(i)}\big)\right),$$

which is, as in in the binary case, a convex problem. Similarly to the binary case, if we use the approximation

$$\log\left(\sum_{i=1}^{k}\exp(a_i)\right) - a_j \;\approx\; \mathsf{max}\{0,\, \mathsf{const.} + \mathsf{max}\{a_1,\ldots,a_k\} - a_j\},$$

then we can approximate the loss function by

$$\mathsf{max}\big\{0,\, 1 + \mathsf{max}_{y'\in\mathcal{Y}}\left\{\theta_{y'}^{\top}\hat{\varphi}(x^{(i)})\right\} - \theta_{y^{(i)}}^{\top}\hat{\varphi}(x^{(i)})$$

after a rescaling such that the constant (const,) becomes 1. This is the well known *Crammer-Singer loss* for multiclass classification.
The same reasoning as for the binary case shows that the corresponding approximation of the MAP estimator for θ, using a standard Gaussian prior distribution

$$p(\theta) \sim \mathcal{N}(0, c\cdot\mathbb{1}), \;\; \text{that is,} \;\; p(\theta) = \frac{1}{(2c\pi)^{n/2}}\exp\left(-\frac{\|\theta\|^2}{2c}\right),$$

is computed as

$$\mathsf{argmin}_{\theta} \quad \frac{1}{2}\|\theta\|^2 \;+\; c\cdot\sum_{i=1}^{m}\mathsf{max}\big\{0,\, 1 + \mathsf{max}_{y'\in\mathcal{Y}}\left\{\theta_{y'}^{\top}\hat{\varphi}(x^{(i)})\right\} - \theta_{y^{(i)}}^{\top}\hat{\varphi}(x^{(i)}).$$

Finally, we can extend the classification approach to multiple labels, that is, we assume $\mathcal{Y} = \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_n$, with finite sets $\mathcal{Y}_i$. Here, the probability distribution $p(y : x)$ can be a Markov random field. In the latter case the model is also called a *conditional random field*.

Chapter 10

Bayesian Model Selection

The Bayesian approach to model selection does not use optimization for model selection, but samples models from a posterior distribution. Therefore, we need sampling instead of optimization algorithms for Bayesian model selection.

For generating a "random" sample from given probability distribution, we assume that we have access to a pseudo random number generator. Pseudo random numbers are well established. *Deterministic* methods exist for generating samples from the unit interval $[0, 1]$, that approximate samples drawn independently from the uniform distribution, that is, the probability to observe a sample point in any closed subinterval $[a, b] \subseteq [0, 1]$ is just $b - a$.

Sampling from a Dirichlet Distribution

As we have discussed before, a conjugate prior for the natural parameters of a multivariate categorical is a Dirichlet distribution. Assuming a Dirichlet distributed prior $p(\mathbf{p})$ on the space $\Delta_{\mathcal{X}}$ of natural parameters leads to an updated posterior Dirichlet distributed distribution on $\Delta_{\mathcal{X}}$. For Bayesian model selection we need to sample from the posterior distribution, that is, from a Dirichlet distribution.
Sampling from a Dirichlet distribution is not an easy undertaking. Here, we present a special, but limited approach that first uses pseudo-random numbers for sampling from an exponential distribution. The samples are then used to sample from a Gamma distribution. Finally, the samples from the Gamma distribution are used to sample from a Dirichlet distribution.

Exponential distribution. An exponential distribution is defined on the sample space $\mathcal{X} = (0, \infty)$. Its density function is given as $p(x) = \lambda \exp(-\lambda x)$ for

some *rate* parameter $\lambda > 0$. Sampling from an exponential distribution is easy, because if u is uniformly distributed on $[0, 1]$, then $x = -\frac{1}{\lambda}\log(u)$ is exponentially distributed, because th probability to drawn an exponentially distirbuted value x from the interval $\big[-\log(b)/\lambda, -\log(a)/\lambda\big] \subset (0, \infty)$ is

$$\lambda \int_{-\log(b)/\lambda}^{-\log(a)/\lambda} \exp(-\lambda x)dx \;=\; \Big[-\exp(-\lambda x)\Big]_{-\log(b)/\lambda}^{-\log(a)/\lambda} \;=\; b - a,$$

which equals the probability to draw an uniformly distributed value u from the interval $[a, b] \subset [0, 1]$.

Gamma distribution. Exponential distributions are special cases of Gamma distributions. Gamma distributions are a two-parameter family of distributions $\Gamma(\alpha, \beta)$ on the sample space $\mathcal{X} = (0, \infty)$. Its densities are given as

$$f(x) \;=\; \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x),$$

where $\alpha > 0$ is called the *shape* and $\beta > 0$ the *rate* parameter. The special case $\Gamma(1, \lambda)$ has the density $f(x) = \lambda \exp(-\lambda x)$, because $\Gamma(1) = 0! = 1$. That is, $\Gamma(1, \lambda)$ is an exponential distribution.

Lemma 9. *Let $X_1, \ldots, X_n$ be independent random variables that are $\Gamma(\alpha_i, \beta), i \in [n]$, distributed. Then $\sum_{i=1}^n X_i$ is $\Gamma\big(\sum_{i=1}^n \alpha_i,\ \beta\big)$ distributed.*

Proof. It is enough to show the claim for X_1 and X_2, because the general case then follows inductively. Let f_1 be the density $\Gamma(\alpha_1, \beta)$ and f_2 be the density of $\Gamma(\alpha_2, \beta)$. Since X_1 and X_2 are independent, $X_1 + X_2$ has the density

$$\begin{aligned}
f(x) \;&=\; \int_0^x f_1(z) f_2(x - z) dz \\
&=\; \frac{\beta^{\alpha_1}\beta^{\alpha_2}}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_0^x z^{\alpha_1 - 1} \exp(-\beta z)(x - z)^{\alpha_2 - 1} \exp\big(-\beta(x - z)\big) dz \\
&=\; \frac{\beta^{\alpha_1+\alpha_2} \exp(-\beta x)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_0^x z^{\alpha_1 - 1}(x - z)^{\alpha_2 - 1} dz \\
&=\; \frac{\beta^{\alpha_1+\alpha_2} \exp(-\beta x)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_0^1 (xt)^{\alpha_1 - 1}(x - xt)^{\alpha_2 - 1} x dt \\
&=\; \frac{\beta^{\alpha_1+\alpha_2} \cdot x^{\alpha_1+\alpha_2-1} \exp(-\beta x)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_0^1 t^{\alpha_1 - 1}(1 - t)^{\alpha_2 - 1} dt \\
&=\; \frac{\beta^{\alpha_1+\alpha_2}}{\Gamma(\alpha_1 + \alpha_2)} x^{\alpha_1+\alpha_2-1} \exp(-\beta x),
\end{aligned}$$

where we have used the substitution $z = xt$ for the fourth equality and that

$$f(t) \;=\; \frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} t^{\alpha_1-1}(1-t)^{\alpha_2-1}$$

is the density of the Beta distribution $B(\alpha_1, \alpha_2)$. □

It immediately follows that the sum of n independent exponentially distributed random variables with the same rate parameter λ is $\Gamma(n, \lambda)$ distributed.

Theorem 7. *Let $X_1, \ldots, X_n$ be independent random variables that are $\Gamma(\alpha_i, 1)$ distributed. Let $Z = \sum_{i=1}^{n} X_i$ and $Y_i = X_i/Z$. Then $\big(Y_1, \ldots, Y_n\big)$ is Dirichlet distributed with parameter vector $(\alpha_1, \ldots, \alpha_n)$.*

Proof. By construction, we have $0 \leq \sum_{i=1}^{n-1} Y_i \leq 1$ and the joint density function for the independent variables $X_1, \ldots, X_n$ is

$$f(x_1, \ldots, x_n) \;=\; \prod_{i=1}^{n} \frac{1}{\Gamma(\alpha_i)} x^{\alpha_i - 1} \exp(-x_i).$$

We can get back the variables X_i from the variables $Y_1, \ldots, Y_{n-1}, Z$ by the following transformation T,

$$X_i \;=\; Y_i \cdot Z \;\text{ for } i \in [n-1], \;\text{ and } X_n \;=\; \Big(1 - \sum_{i=1}^{n-1} Y_i\Big) \cdot Z.$$

For deriving the joint density function g of the new variables $Y_1, \ldots, Y_{n-1}, Z$, we use the change of variables transform for multivariate integrals

$$\begin{aligned}
&\int f(x_1, \ldots, x_n) dx_1 \ldots dx_n \\
&\qquad = \;\int f\Big(y_1 z, \ldots, y_{n-1} z, \Big(1 - \sum_{i=1}^{n-1} y_i\Big) z\Big) |J(T)| dy_1 \ldots dy_{n-1} dz \\
&\qquad =: \int g(y_1, \ldots, y_{n-1}, z) dy_1 \ldots dy_{n-1} dz
\end{aligned}$$

where $|J(T)|$ is the determinant of the Jacobian matrix of the transformation T,

which is given as

$$|J(T)| = \begin{vmatrix} Z & 0 & \dots & 0 & Y_1 \\ 0 & Z & \dots & 0 & Y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & Z & Y_{n-1} \\ -Z & -Z & \dots & -Z & 1 - \sum_{i=1}^{n-1} Y_i \end{vmatrix} = Z^{n-1},$$

where we have developed the determinant by the last column. Therefore, the joint density g of the variables $Y_1, \dots, Y_{n-1}, Z$ can be computed from the joint density f of the variables $X_1, \dots, X_n$ as

$$\begin{aligned} & f\Big(y_1 z, \dots, y_{n-1} z, \Big(1 - \sum_{i=1}^{n-1} y_i\Big) z\Big) |J(T)| \\ &= \frac{\Big(\prod_{i=1}^{n-1} (y_i z)^{\alpha_i - 1}\Big)\Big(\big(1 - \sum_{i=1}^{n-1} y_i\big) z\Big)^{\alpha_n - 1}}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \cdots \\ &\qquad \dots \exp\Big(- \sum_{i=1}^{n-1} y_i z - \Big(1 - \sum_{i=1}^{n-1} y_i\Big) z\Big) \cdot z^{n-1} \\ &= \frac{\Big(\prod_{i=1}^{n-1} y_1^{\alpha_i - 1}\Big)\Big(1 - \sum_{i=1}^{n-1} y_i\Big)^{\alpha_n - 1}}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \exp(-z) z^{\sum_{i=1}^{n} \alpha_i - 1} \\ &= \frac{\Gamma\Big(\sum_{i=1}^{n} \alpha_i\Big)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \Big(\prod_{i=1}^{n-1} y_1^{\alpha_i - 1}\Big)\Big(1 - \sum_{i=1}^{n-1} y_i\Big)^{\alpha_n - 1} \frac{1}{\Gamma\Big(\sum_{i=1}^{n} \alpha_i\Big)} z^{\sum_{i=1}^{n} \alpha_i - 1} \exp(-z) \\ &= g(y_1, \dots, y_{n-1}, z) =: h(y_1, \dots, y_n) l(z). \end{aligned}$$

That is, $g(y_1, \dots, y_{n-1}, z)$ is the product of a Dirichlet distribution with density

$$\begin{aligned} h(y_1, \dots, y_n) &= \frac{\Gamma\Big(\sum_{i=1}^{n} \alpha_i\Big)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \Big(\prod_{i=1}^{n-1} y_1^{\alpha_i - 1}\Big)\Big(1 - \sum_{i=1}^{n-1} y_i\Big)^{\alpha_n - 1} \\ &= \frac{\Gamma\Big(\sum_{i=1}^{n} \alpha_i\Big)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \Big(\prod_{i=1}^{n} y_1^{\alpha_i - 1}\Big) \end{aligned}$$

and parameter vector $(\alpha_1, \dots, \alpha_n)$ for $(Y_1, \dots, Y_n)$ and a $\Gamma\Big(\sum_{i=1}^{n} \alpha_i, 1\Big)$ distribution with density $l(z)$ for Z. □

Given natural numbers $n_1, \dots n_{n-1}$, the above reasoning can be summarized in Algorithm 1 for sampling one sample from a Dirichlet distribution with parameter vector $(n_1, \dots, n_{n-1})$.

Algorithm 1 Dirichlet sample

input $n_1, \dots, n_n \in \mathbb{N}$
output $(y_1, \dots, y_n) \in \Delta_n$ drawn from $\mathrm{Dir}\big((n_1, \dots, n_n)\big)$

for $i = 1$ to n **do**
  for $j = 1$ to n_i **do**
    $u_{ij} =$ pseudo-random number in $[0, 1]$; $v_{ij} = \exp(-u_{ij})$
    $x_i = x_i + v_{ij}$
  end for
end for
$z = \sum_{i=1}^{n} x_i$
for $i = 1$ to $n - 1$: $y_i = x_i / z$
return $\big(y_1, \dots, y_{n-1}, 1 - \sum_{i=1}^{n-1} y_i\big)$

Gaussian Prior for Interaction Parameters

A joint probability density function of the form

$$p : \mathbb{R}^n \to (0, \infty),\ x \mapsto (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2} (x - \boldsymbol{\mu})^\top \Sigma^{-1} (x - \boldsymbol{\mu}) \right),$$

for a vector $(X_1, \dots, X_n)$ of random variables is called an n-variate Gaussian. The vector $\boldsymbol{\mu} \in \mathbb{R}^n$ is called the *mean vector*, and the positive definite matrix $\Sigma \in \mathbb{R}^{n \times n}$ is called the *covariance matrix*. $|\Sigma|$ is the determinant of Σ. We also use the shorthand notation $p(x) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

We have seen in Chapter 1 that, by using interaction parameters, multivariate categoricals can be expressed as

$$p(x) \ = \ \exp\big(\mathbf{v}(x)^\top \mathbf{q} - a(\mathbf{q})\big),$$

where

$$\mathbf{v}(x) \ = \ \Big(\mathbf{v}_I(x) \ : \ I \in \mathcal{I}_k, k \in [n]\Big) \in \{0,1\}^{n_1 \cdot \dots \cdot n_n - 1},\ x \in \mathcal{X} \setminus \big\{(1, \dots, 1)\big\}$$

and

$$\mathbf{v}_I(x) \ = \ \Big(\mathbf{1}[x_I = \hat{x}]\Big)_{\hat{x} \in \hat{\mathcal{X}}_I} \in \{0,1\}^{(n_{i_1} - 1) \cdot \dots \cdot (n_{i_k} - 1)},\ x \in \mathcal{X} \setminus \big\{(1, \dots, 1)\big\}.$$

Let X be a set of independently sampled data points $x^{(1)}, \ldots, x^{(m)} \in \mathcal{X}$. If $\mathbf{q} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{1})$, then

$$\begin{aligned} p(\mathbf{q} \mid X) &= \frac{p(X \mid \mathbf{q})p(\mathbf{q})}{p(X)} \\ &= \frac{\exp\Big(\sum_{i=1}^m \mathbf{v}(x^{(i)})^\top \mathbf{q} - a(\mathbf{q})\Big) \exp\Big(-\tfrac{1}{2}(\mathbf{q}-\boldsymbol{\mu})^\top(\mathbf{q}-\boldsymbol{\mu})\Big)}{(2\pi)^{-n/2}p(X)} \\ &= \exp\Big(-\frac{1}{2}\big(\mathbf{q} - \sum_{i=1}^m \mathbf{v}(x^{(i)}) - \boldsymbol{\mu}\big)^\top \big(\mathbf{q} - \sum_{i=1}^m \mathbf{v}(x^{(i)}) - \boldsymbol{\mu}\big)\Big) \ldots \\ &\qquad \ldots \frac{\exp\Big(\tfrac{1}{2}\big\|\sum_{i=1}^m \mathbf{v}(x^{(i)}) + \boldsymbol{\mu}\big\|^2 - \tfrac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\mu}\Big) - a(\mathbf{q}}{(2\pi)^{-n/2}p(X)} \\ &\sim \mathcal{N}\Big(\sum_{i=1}^m \mathbf{v}(x^{(i)}) + \boldsymbol{\mu}, \mathbf{1}\Big). \end{aligned}$$

Therefore, $\mathcal{N}(\boldsymbol{\mu}, \mathbf{1})$ is a conjugate prior for multivariate categorical that is parameterized by interaction parameters. A "canonical" choice for $\boldsymbol{\mu}$ would be constant a vector, for instance, $\boldsymbol{\mu} = \text{vec}(0)$ the all-zero vector.

Box-Muller Transform for Sampling from a Gaussian

Let $(u, v) \in (0, 1)^2$ be drawn at random from the uniform distribution on $(0, 1)^2$. Such a random vector can be obtained by sampling u and v independently from the uniform distribution on $(0, 1)$ by using a pseudo-random number generator. The *Box-Muller transform* maps the vector (u, v) to some point in $\mathbb{R}^2$ as follows:

$$(0,1)^2 \ni (u, v) \mapsto \sqrt{-2\log u}\big(\cos(2\pi v), \sin(2\pi v)\big) \in \mathbb{R}^2.$$

It remains to show that

$$\sqrt{-2\log u}\cos(2\pi v) \quad \text{and} \quad \sqrt{-2\log u}\sin(2\pi v)$$

are independent samples from a standard univariate Gaussian.
Let (X_1, X_2) be a vector of two random variables whose joint density function p is a standard bivariate Gaussian $\mathcal{N}(0, \mathbb{1}_2)$. The following simple calculation shows that X_1 and X_2 are independent,

$$p(x_1, x_2) = \frac{1}{2\pi}\exp\left(-\frac{x_1^2 + x_2^2}{2}\right) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x_1^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x_2^2}{2}\right).$$

For any measurable subset $A \subset \mathbb{R}^2$, the probability to observe a sample point from $\mathcal{N}(0, \mathbb{1}_2)$ in A is given as

$$\begin{aligned}
P[A] &= \frac{1}{2\pi} \int_A \exp\big(-(x_1^2 + x_2^2)/2\big) dx \\
&= \frac{1}{2\pi} \int_{\mathbb{R}^2} \mathbf{1}[x \in A] \exp\big(-(x_1^2 + x_2^2)/2\big) dx \\
&= \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} \mathbf{1}\big[(r\cos\varphi, r\sin\varphi) \in A\big] \exp(-r^2/2) r \, dr d\varphi \\
&= \int_0^1 \int_0^{\infty} \mathbf{1}\Big[\big(\sqrt{2w}\cos(2\pi v), \sqrt{2w}\sin(2\pi v)\big) \in A\Big] \exp(-w) \, dw dv \\
&= \int_0^1 \int_0^1 \mathbf{1}\Big[\big(\sqrt{-2\log u}\cos(2\pi v), \sqrt{-2\log u}\sin(2\pi v)\big) \in A\Big] \, du dv \\
&= P\Big[\big(\sqrt{-2\log u}\cos(2\pi v), \sqrt{-2\log u}\sin(2\pi v)\big) \in A\Big].
\end{aligned}$$

Here we have used several coordinate transformations. First, the transformation to *polar coordinates* $(r, \varphi) \in [0, \infty) \times [0, 2\pi)$, then the transformations $w = r^2/2$, that is, $dw = rdr$, and $v = \varphi/2\pi$, that is, $dv = \frac{d\varphi}{2\pi}$, and finally $u = \exp(-w)$, that is, $du = -\exp(-w)dw$, where the minus sign is accounted for by switching the bounds of integration.
The calculation shows that the probability for the two points that we obtain from a uniform sample from $(0, 1)^2$ through the Box-Muller transform to be contained in a measurable set A is the same as to observe a vector (x, y), whose components are sampled independently from a standard univariate Gaussian, in A. Since this holds for any measurable set $A \subseteq \mathbb{R}^2$, the Box-Muller transform indeed provides us with two independent sample points from a standard univariate Gaussian.

For generating sample points from a non-standard univariate Gaussian with $p(z) \sim \mathcal{N}(\mu, \sigma^2)$ we can just sample points x from the standard univariate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, for instance, by using the Box-Muller transform, and transform them as

$$z = \sigma \cdot x + \mu.$$

Then z is sampled from $\mathcal{N}(\mu, \sigma^2)$.

Multivariate Gaussians. There is a simple and efficient method for employing a univariate Gaussian sampler to sample from a multivariate Gaussian with mean vector μ and covariance matrix Σ.

Obviously, it is easy to sample from a multivariate Gaussian with mean vector 0 and covariance matrix $\mathbb{1}_n$, because this just boils down to generating n sample points from a standard univariate Gaussian for one multivariate sample point. Let $X = (X_1, \ldots, X_n)$ be a vector of random variables whose joint density function is a standard multivariate Gaussian, that is, $p(x) \sim \mathcal{N}(0, \mathbb{1}_n)$, and let $Y = (Y_1, \ldots, Y_n)$ be a vector of random variables whose joint density function is $p(y) \sim \mathcal{N}(\mu, \Sigma)$. A non-zero mean vector μ is not a problem since we just need to add μ to sample points drawn from $\mathcal{N}(0, \Sigma)$. Thus we can assume $\mu = 0$ and it remains to turn sample points drawn from $\mathcal{N}(0, \mathbb{1}_n)$ into sample points drawn from $\mathcal{N}(0, \Sigma)$.
Assume that there exists a matrix $A \in \mathbb{R}^{n \times n}$ such that $Y = AX$. Then we get

$$\begin{aligned}\Sigma &= \mathsf{Cov}[Y] = \mathsf{E}\big[Y - \mu)(Y - \mu\big]^\top) = \mathsf{E}\big[YY^\top\big] \\ &= \mathsf{E}\big[AXX^\top A\big] = A\mathsf{E}\big[XX^\top\big]A = A\mathsf{Cov}[X, X]A^\top = AA^\top.\end{aligned}$$

Any symmetric, positive definite matrix Σ has a unique representation

$$\Sigma = LDL^\top,$$

where L is a lower triangular and D is a diagonal matrix with positive entries on the diagonal. Setting $A = LD^{1/2}$ gives $\Sigma = AA^\top$. Hence, there exists a unique A such that $Y = AX$.
The decomposition $\Sigma = AA^\top$ is called *Cholesky decomposition* and can be computed efficiently, see the appendix to this section. A sample point y from a $\mathcal{N}(\mu, \Sigma)$ distribution can now be computed from a sample point x of a $\mathcal{N}(0, \mathbb{1}_n)$ distribution by the transformation

$$z = Ax + \mu,$$

where A has been computed from a Cholesky decomposition of Σ.

Appendix: Cholesky decomposition

Theorem 8. [Cholesky decomposition] *Any symmetric, positive semi-definite matrix $\Sigma \in \mathbb{R}^{n \times n}$ can be decomposed as $\Sigma = AA^\top$, where A is a lower triangular matrix.*

Proof. The proof of the Cholesky decomposition theorem is constructive in the sense that it gives a recursive procedure for computing the decomposition.
Write Σ as

$$\Sigma = \begin{pmatrix} s & v^\top \\ v & \hat{\Sigma} \end{pmatrix},$$

where $s \geq 0$, because the positive semi-definiteness of Σ implies $s = e_1^\top \Sigma e_1 \geq 0$, where e_1 is the first standard basis vector $e_1 = (1, 0, \ldots, 0)^\top$.

We distinguish the two cases $s > 0$ and $s = 0$.

First case ($s > 0$): The matrix Σ can be written as

$$\Sigma = \begin{pmatrix} \sqrt{s} & \mathbf{0}^\top \\ v/\sqrt{s} & \mathbb{1}_{n-1} \end{pmatrix} \begin{pmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \hat{\Sigma} - vv^\top/s \end{pmatrix} \begin{pmatrix} \sqrt{s} & v^\top/\sqrt{s} \\ 0 & \mathbb{1}_{n-1} \end{pmatrix}.$$

We claim that $\hat{\Sigma} - vv^\top/s \in \mathbb{R}^{(n-1)\times(n-1)}$ is again symmetric and positive semi-definite. The symmetry is obvious since both $\hat{\Sigma}$ and $-vv^\top/s$ are symmetric. Any vector in $\mathbb{R}^{n-1}$ can be written as $\lambda v + u$, where $u^\top v = 0$ and $\lambda \in \mathbb{R}$, and we have

$$(\lambda v+u)^\top(\hat{\Sigma}-vv^\top/s)(\lambda v+u) = \begin{pmatrix} -\frac{\lambda}{s}v^\top v \\ \lambda v + u \end{pmatrix}^\top \begin{pmatrix} s & v^\top \\ v & \hat{\Sigma} \end{pmatrix} \begin{pmatrix} -\frac{\lambda}{s}v^\top v \\ \lambda v + u \end{pmatrix} \geq 0.$$

Thus, $\hat{\Sigma} - vv^\top/s$ is also positive semi-definite. Recursively, let $\hat{A} \in \mathbb{R}^{(n-1)\times(n-1)}$ be a lower triangular matrix such that $\hat{A}\hat{A}^\top = \hat{\Sigma} - vv^\top/s$, then it follows for the lower triangular matrix

$$A = \begin{pmatrix} \sqrt{s} & \mathbf{0}^\top \\ v/\sqrt{s} & \hat{A} \end{pmatrix} \in \mathbb{R}^{n\times n}$$

that $AA^\top = \Sigma$.

Second case ($s = 0$): It must hold that $v = 0$, because otherwise the following expression

$$\begin{pmatrix} \lambda \\ v \end{pmatrix}^\top \begin{pmatrix} 0 & v^\top \\ v & \hat{\Sigma} \end{pmatrix} \begin{pmatrix} \lambda \\ v \end{pmatrix} = 2\lambda\|v\|^2 + v^\top\hat{\Sigma}v$$

can become negative for $\lambda \ll 0$ which contradicts the positive semi-definiteness of Σ. Since $\hat{\Sigma}$ is symmetric and positive semi-definite, which follows immediately from the corresponding properties of Σ, the Cholesky factor A can be computed recursively. Let $\hat{A} \in \mathbb{R}^{(n-1)\times(n-1)}$ be a lower triangular matrix such that $\hat{A}\hat{A}^\top = \hat{\Sigma}$, then it follows for the lower triangular matrix

$$A = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \hat{A} \end{pmatrix} \in \mathbb{R}^{n\times n}$$

that $AA^\top = \Sigma$. □

Remark: The recursive algorithm for computing the Cholesky decomposition is given as follows: Let **Cholesky**$\big(\Sigma, n\big)$ the Cholesky factor A of Σ.

1. If $n = 1$, then return $\sqrt{s}$, where $\Sigma = s$.
2. Else, assume $\Sigma = \begin{pmatrix} s & v^\top \\ v & \hat{\Sigma} \end{pmatrix} \in \mathbb{R}^{n \times n}$ and return

$$\begin{pmatrix} \sqrt{s} & \mathbf{0}^\top \\ v/\sqrt{s} & \textbf{Cholesky}\big(\hat{\Sigma} - vv^\top/s, n-1\big) \end{pmatrix}$$

if $s > 0$ and otherwise (if $s = 0$ and thus also $v = 0$) return

$$\begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \textbf{Cholesky}\big(\hat{\Sigma}, n-1\big) \end{pmatrix}.$$

Chapter 11

Exercises

Chapter 5

Exercise 1. Multivariate categoricals are just a special (structured) case of categorical distributions that are also known as *multinoulli distributions*. A categorical distribution is defined on a finite space $\mathcal{X}$, where $x \in \mathcal{X}$ has the probability $p(x) \in [0, 1]$ such that

$$\sum_{x \in \mathcal{X}} p(x) \;=\; 1.$$

Bernoulli distributions are the special case of categorical distributions where $|\mathcal{X}| = 2$. Since Bernoulli distributions are important in themselves and illustrate the general case quite well we briefly discuss the model selection problem for Bernoulli distributions in the following.

Assume that $\mathcal{X} = \{0, 1\}$ and that we have observed $x^{(1)}, \ldots, x^{(m)}$, where the $x^{(i)} \in \mathcal{X}$ have been drawn independently from a Bernoulli distribution with parameter $p \in [0, 1]$, that is, $p(x = 1) = p$. We summarize the observed data points in the sequence X.
Show that

1. The parameter p can be estimated as

   $$p_{ML} \;=\; \mathsf{argmax}_p \, L(p)$$

   from the likelihood function

   $$L(p) \;=\; p(X|p) \;=\; \binom{m}{n} p^n (1-p)^{m-n},$$

where n is the number of observations with value 1. Derive an explicit formula for p_{ML}.

Hint: Again, it is easier to work with the log-likelihood function instead of the likelihood function.

2. Treating p also as a random variable we can use the *Bayes theorem* and write

$$p(p|X) = \frac{p(p,X)}{p(X)} = \frac{p(X|p)\,p(p)}{p(X)}.$$

A particularly interesting choice is to model $p(p)$ as beta distributed with real parameters $\alpha > 0$ and $\beta > 0$, that is,

$$p_{\alpha,\beta}(p) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1} \sim \text{Beta}(\alpha,\beta),$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and $\Gamma(\cdot)$ is the Gamma function.

Show that the *posterior distribution* $p(p|X)$ is again a beta distribution, that is, the beta distribution a *conjugate prior* distribution for the Bernoulli distribution, and derive an explicit formula for p_{MAP}, that is, the maximum of the posterior distribution.

Chapter 7

Exercise 1. Going back to the example from the introduction. Assume we have observed data that are stored in the following table:

ID	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	0	1	0	1	1	0	0	0	1	0	1	1
B	0	1	1	0	0	1	1	0	1	1	0	1	0
C	0	1	1	0	1	1	0	0	0	0	0	1	1

1. Use the data to compute the maximum likelihood estimate of a trivariate categorical $\big(p(x)\big)_{x\in\{0,1\}^3}$.

2. Does the maximum likelihood estimate respect the interaction graph of the model?

3. Calculate the first iteration of the iterative proportional scaling algorithm on the thirteen data points.

Chapter 8

Exercise 1. Given a *linear exponential family*

$$p(x) \,=\, \exp\left(\sum_{i=1}^{k} \theta_i \varphi_i(x) - a(\boldsymbol{\theta})\right) \,=\, \exp\left(\boldsymbol{\theta}^\top \boldsymbol{\varphi}(x) - a(\boldsymbol{\theta})\right)$$

with canonical parameters $\boldsymbol{theta} \in \mathbb{R}^k$.
Compute and interpret the gradient and the Hessian of the *log-partition function* $a(\boldsymbol{\theta})$.

Exercise 2. Show that Dirichlet distributions

$$p(\mathbf{p}) \,=\, \frac{1}{D(\boldsymbol{\alpha})} \prod_{i=1}^{n} p_i^{\alpha_i - 1} \,\sim\, \mathsf{Dir}(\boldsymbol{\alpha}),$$

form a linear exponential family on the $(n-1)$-dimensional unit simplex Δ_{n-1}. Here,

$$\mathbf{p} \,=\, \sum_{i=1}^{n} p_i e_i, \quad \text{and} \quad \sum_{i=1}^{n} p_i \,=\, 1 \;\text{ and }\; p_i \,\geq\, 0,\, i \in [n],$$

where e_i, $i \in [n]$ is the i-th standard basis vector of $\mathbb{R}^n$, is an extreme point (vertex) of Δ_{n-1}.

Part III

Inference

Chapter 12

Complexity of Inference Queries

Here, inference means answering queries on the models, multivariate categoricals, that we have learned before. Serving queries on a model is typically considered in the area of *expert systems* that are concerned with the combination of *knowledge bases* and *inference engines*. In probabilistic expert systems the knowledge base is a probability distribution p on a multidimensional sample space $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$, typically a graphical model, and the inference engine supports at least the following queries:

1. *Prior and posterior marginals:* For the prior marginal, a subset of indices $I \subset [n] = \{1, \ldots, n\}$ is given and the marginal distribution $p(x_I)$ has to be computed. Here, x_I is the projection of $x \in \mathcal{X}$ onto the indices in I, that is, $x_I \in \mathcal{X}_I = \prod_{i \in I} \mathcal{X}_i$. For the posterior marginal another subset of indices $J \subseteq [n]$ is given together with fixed values $e_J \in \mathcal{X}_J$, called *evidence*, for the corresponding variables, and the distribution $p(x_I \mid e_J)$ on $\mathcal{X}_I$ has to be computed. The subset J does not have to be disjoint from I. If the subsets are not disjoint, then $p(x_I \mid e_J) = 0$ for all $x_I \in \mathcal{X}_I$ with $(x_I)_{I \cap J} \neq (e_J)_{I \cap J}$.

2. *Probability of evidence:* For subsets of indices $I, J \subseteq [n]$ and evidence $e_I \in \mathcal{X}_I$ and $e_J \in \mathcal{X}_J$ compute the posterior marginal probability $p(e_I \mid e_J)$. In the special case $J = \emptyset$, compute the prior marginal probability $p(e_I)$.

3. *Most probable explanation (MPE):* Given evidence $e_J \in \mathcal{X}_J$ for a subset of indices $J \subseteq [n]$, compute

$$\mathrm{argmax}_{x \in \mathcal{X}} \; p(x \mid e_J) \; = \; \mathrm{argmax}_{x \in \mathcal{X}} \; p(x, e_J).$$

4. *Maximum a posteriori (MAP) hypothesis:* Given the subsets $I, J \subseteq [n]$ and evidence $e_J \in \mathcal{X}_J$, similarly as in the definition of the marginal queries. Compute

$$\text{argmax}_{x_I \in \mathcal{X}_I} \; p(x_I \mid e_J) \; = \; \text{argmax}_{x_I \in \mathcal{X}_I} \; p(x_I, e_J).$$

   Note that the MPE query is the special case of a MAP query, where $I = [n]$.

Technically, computing the MAP estimate for model selection is very similar to computing the MAP hypothesis. The latter, though, does not need to choose a prior distribution.

The complexity of inference queries is measured in terms of the size of the input, that is, in the size of the representation of the model. The size of n-variate categoricals without additional structure is $n_1 \cdot \ldots \cdot n_n - 1$, that is, the total number of model parameters. The inference queries can be served in polynomial time in the size of the input, however, the large input size renders these models and inference on them infeasible in practice, already for moderately high-dimensional models, that is, for n in the range of a few dozens. In the following we assume a graphical model structure in the form of a Bayesian network that can dramatically reduce the size of the input (model size) while keeping the size of the sample space on which the inference queries have to be served. Since a Bayesian network is a special case of a Markov random field, serving the queries on Markov random fields is at least as hard as serving the queries on Bayesian networks.

Complexity of Inference Queries

For discussing the complexity of inference queries we consider their decision versions.

1. *Probability of evidence (D-PoE):* For a subsets of indices $I, J \subseteq [n]$, evidence $e_I \in \mathcal{X}_I$ and $e_J \in \mathcal{X}_J$, and $q \geq 0$ decide if $p(e_I \mid e_J) > q$.

2. *Most probable explanation (D-MPE):* Given evidence $e_I \in \mathcal{X}_I$ for a subset of indices $I \subseteq [n]$ and $q \geq 0$, decide if there exists $x \in \mathcal{X}$ such that $p(x, e_I) > q$.

3. *Maximum a posteriori hypothesis (D-MAP):* Given two subsets $I, J \subseteq [n]$, evidence $e_J \in \mathcal{X}_J$ and $q \geq 0$, decide if there exists $x_I \in \mathcal{X}_I$ such that $p(x_I, e_J) > q$.

It turns out that D-MPE is NP-complete, D-PoE is PP-complete, and D-MAP is NP^{PP}-complete.

Let us briefly recap the different complexity classes. The class NP contains all problems that have short proofs, that is, the correctness of a solution can be verified in polynomial time. The class PP contains all problems that can be decided by a randomized algorithm in polynomial time with an error probability less than $\frac{1}{2}$, that is, the algorithm will answer "yes" with probability more than $\frac{1}{2}$, if and only if "yes" is the correct answer. The class PP contains the class NP and thus any polynomial time algorithm for a complete problem in PP would lead to polynomial time algorithms for all problems in NP. Finally, the class NP^{PP} contains all problems whose solutions can be verified in polynomial time, where the verifier can use an oracle that solves problems in PP.

Proving completeness of a problem for a complexity class requires to show (1) that the problem is at least as *hard* as any problem in the class, and (2) that the problem is a *member* of the class. Hardness is typically shown by providing a reduction from a problem that is known to be complete for the class to the given problem, that is, any algorithm for solving the given problem can also be used to solve the problem that is known to be complete for the class. Hence, we need to find suitable problems in the complexity classes NP, PP, and NP^{PP} that can be reduced to D-MPE, D-PoE, and D-MAP, respectively. Here, we choose the problems SAT, MAJSAT (majority SAT), and E-MAJSAT (exists majority SAT) that are known to be complete for the classes NP, PP, and NP^{PP}, respectively.

Let F be a propositional logic formula (Boolean expression) that is build from Boolean variables $X_1, \ldots, X_n$, conjunctions $\wedge$, disjunctions $\vee$, and negations $\neg$. We have the following decision problems:

1. *Satisfiability (SAT):* Decide if there exists an instantiation of the variables that satisfy F.
2. *Majority Satisfiability (MAJSAT):* Decide if the majority of the variable instantiations satisfy f.
3. *Exists Majority Satisfiability (E-MAJSAT):* Given $1 \leq k \leq n$. Decide if there exists an instantiation $X_1^\star, \ldots, X_k^\star$ of $X_1, \ldots, X_k$ for which the majority of instantiations $X_1 = X_1^\star, \ldots, X_k = X_k^\star, X_{k+1}, \ldots, X_n$ satisfy F.

For proving hardness we show how to construct a Bayesian network for a given

propositional logic formula. Let F be be a propositional logic formula, the Bayesian network for F is constructed inductively from valid subexpressions of F. The Bayesian network is defined on the sample space of $\mathcal{X} = \{0,1\}^n$. A valid subexpression of F is itself a propositional logic formula. The inductive construction starts at the variables and keeps the invariant that any constructed Bayesian network for a subexpression has a unique leaf that represents the subexpression. The inductive construction is as follows:

1. If F is a single variable x, then the Bayesian network p_F has a single node f and the CPT associated with this node $p(x = true) = p(x = false) = \frac{1}{2}$.

2. If F is of the form $\neg g$, then the network for F is derived from the network p_G for G by adding a node f for F and a directed edge from the unique leaf of p_g to the new node f. That is, the leaf node that represents G becomes a parent node of the node f. The reduced CPT that is associated with the node f is given by $p(f = true \mid g = true) = 0$ and $p(f = true \mid g = false) = 1$. Note that the CPT is reduced since we do not explicitly state the implied probabilities for $p(f = false \mid g = true)$ and $p(f = false \mid g = false)$.

3. If F is of the form $G \vee H$, then p_F is obtained from the networks p_G and p_H by adding a new node f for F and directed edges from the unique leafs of p_G and p_H, respectively, to the new node. The reduced CPT associated with the new node reads as $p(f = true \mid g = true, h = true) = 1$, $p(f = true \mid g = true, h = false) = 1$, $p(f = true \mid g = false, h = true) = 1$ and $p(f = true \mid g = false, h = false) = 0$.

4. If F is of the form $G \wedge H$, then p_F is obtained from the networks p_G and p_H by adding a new node f for F and directed edges from the unique leafs of p_G and p_H, respectively, to the new node. The reduced CPT associated with the new node reads as: $p(f = true \mid g = true, h = true) = 1$, $p(f = true \mid g = true, h = false) = 0$, $p(f = true \mid g = false, h = true) = 0$ and $p(f = true \mid g = false, h = false) = 0$.

The time needed to construct the Bayesian network p_F from F is linear in the size of F. The three reductions that we are aiming at follow from the following lemma.

Lemma 10. *Let F be a propositional logic formula on the variables $X_1, \ldots, X_n$, where $n \geq 2$ and let $x_1, \ldots, x_n$ denote the corresponding variables of the Bayesian network p_F. Moreover, let f be the variable of p_F that corresponds the unique leaf of the DAG that represents the whole expression F. Then we*

have for the marginal distribution, where all variables different from $x_1, \ldots, x_n$ *and* f *have been marginalized out, that*

$$p_F(x_1, \ldots, x_n, f = true) = \begin{cases} 0 & : X_1, \ldots, X_n \text{ satisfies } \neg F \\ 2^{-n} & : X_1, \ldots, X_n \text{ satisfies } F \end{cases}$$

Proof. The claim follows, because the probability to observe $f = true$ for a non-satisfying instantiation of $X_1, \ldots, X_n$ is always 0 by construction, and the probability to observe any fixed instantiation $x_1, \ldots, x_n$ is 2^{-n}. □

Theorem 9. *Let* F *be a propositional logic formula on the variables* $X_1, \ldots, X_n$*, where* $n \geq 2$*, and let* $x_1, \ldots, x_n, g_1, \ldots, g_m$*, and* f *denote random variables of the Bayesian network* p_F *for* F*, where* $g_1, \ldots, g_m$ *are the variables for the subexpresssions at the inner nodes of the Bayesian network DAG. Then the following holds true:*

1. (Reducing SAT to D-MPE): *There exists an instantiation of the variables* $X_1, \ldots, X_n$ *that satisfies* F*, if and only if there exists an instantiation of the variables* $x_1, \ldots, x_n$ *and the variable* $g_1, \ldots, g_m$ *such that*

$$p_F(x_1, \ldots, x_n, g_1, \ldots, g_m, f = true) > 0.$$

2. (Reducing MAJSAT to D-PoE): *The majority of the instantiations of* $X_1, \ldots, X_n$ *satisfy* F*, if and only if*

$$p_F(f = true) > \frac{1}{2}.$$

   Here, $f = true$ *is the evidence.*

3. (Reducing E-MAJSAT to D-MAP): *For* $0 \leq k \leq n$*, there exists an instantiation* $X_1^\star, \ldots, X_k^\star$ *of the variables* $X_1, \ldots, X_k$ *for which the majority of the instantiations* $X_1 = X_1^\star, \ldots, X_k = X_k^\star, X_{k+1}, \ldots, X_n$ *satisfy* F*, if and only if there exists an instantiation of the variables* $x_1, \ldots, x_k$ *of* p_F *such that*

$$p_F(x_1 = x_1^\star, \ldots, x_k = x_k^\star, f = true) > 2^{-(k+1)}.$$

Proof. For (1), note that the variables $g_1, \ldots, g_m$ are deterministic functions of the variables $x_1, \ldots, x_n$, Therefore, there exists an instantiation of the variables $x_1, \ldots, x_n$ and the variable $g_1, \ldots, g_m$ such that

$$p_F(x_1, \ldots, x_n, g_1, \ldots, g_m, f = true) > 0.$$

if and only if there exists an instantiation of the variables $x_1, \ldots, x_n$ such that

$$p_F(x_1, \ldots, x_n, f = true) > 0.$$

The claim then follows directly from Lemma 10.

For (2), observe that there are 2^n instantiations of $X_1, \ldots, X_n$. Therefore, by Lemma 10,

$$\begin{aligned} &p(f = true) \\ &\quad = \sum_{x \in \{0,1\}^n} p_F(x_1, \ldots, x_n, f = true) = 2^{-n} \big|\big\{x \in \{0,1\}^n : x \text{ satisfies } f\big\}\big|, \end{aligned}$$

which means that $p_F(f = true) > \frac{1}{2}$ is equivalent to

$$\big|\big\{X \in \{true, false\}^n : X \text{ satisfies } F\big\}\big| > \frac{1}{2}|\{true, false\}^n| = 2^{n-1}.$$

Similarly for (3), there are are 2^{n-k} instantiations of $X_{k+1}, \ldots, X_n$. Therefore,

$$\begin{aligned} &p_F(x_1 = x_1^\star, \ldots, x_k = x_k^\star, f = true) \\ &\qquad = \sum_{x \in \{0,1\}^{n-k}} p_F(x_1 = x_1^\star, \ldots, x_k = x_k^\star, x, f = true) \\ &\qquad = 2^{-n} \big|\big\{X \in \{true, false\}^{n-k} : (X_1^\star, \ldots, X_k^\star, X) \text{ satisfies } F\big\}\big|, \end{aligned}$$

which means that $p_F(x_1 = x_1^\star, \ldots, x_k = x_k^\star, f = true) > 2^{-(k+1)}$ is equivalent to

$$\begin{aligned} &\big|\big\{\hat{X} \in \{true, false\}^{n-k} : (X_1^\star, \ldots, X_k^\star, X) \text{ satisfies } F\big\}\big| \\ &\qquad\qquad > \frac{1}{2}\big|\{true, false\}^{n-k}\big| = 2^{n-k-1}. \end{aligned}$$

□

Theorem 9 establishes the hardness of the different inference queries for their respective complexity class. To finish the completeness proof we still need to show that the problems are members of the complexity classes.

Lemma 11. *D-MPE is in NP.*

Proof. For the proof we need to show that we can validate a solution $x \in \mathcal{X}$ to the D-MPE problem in polynomial time. That is, we need to decide in polynomial time whether $p(x, e_I) > q$ is true or false. For that it is enough to compute $p(x, e_I)$ in polynomial time. If x is inconsistent with e_I, which can be decided in linear time, then $p(x, e_I) = 0$. Otherwise, $p(x, e_I) = p(x)$ can be computed using the chain rule. Computing the probability using the chain rule takes polynomial time in the size of the input, that is, the CPTs of the Bayesian network, because we need to access and multiply n entries of the CPTs, that is, one entry per CPT. □

Lemma 12. *D-PoE is in PP.*

Proof. We have to design a randomized algorithm that computes a solution to the D-PoE problem in polynomial time, while guaranteeing that the solution is correct with probability larger than $\frac{1}{2}$. The following algorithm does the job:

1. Sample a variable instantiation x from the Bayesian network, which can be done in polynomial time (here, without proof).
2. Return that "$p(e_I \mid e_J) > q$ is true" with the following probabilities:
   A. $\min\left\{1, \frac{1}{2q}\right\}$, if x is compatible with e_I and e_J,
   B. $\max\left\{0, \frac{1-2q}{2-2q}\right\}$, if x is compatible with e_J but not with e_I, and
   C. $\frac{1}{2}$, if x is not compatible with e_J.

Observe that the probability that the sampled x falls under Case A is $p(e_I, e_J)$, that it falls under Case B is

$$\begin{aligned}\sum_{x \in \{0,1\}^{|I|}:\, x \neq e_I} p(x_I = x, e_J) &= \sum_{x \in \{0,1\}^{|I|}: x \neq e_I} p(x_I = x | e_J) p(e_J) \\ &= \big(1 - p(e_I \mid e_J)\big) p(e_J),\end{aligned}$$

and that it falls under Case C is $1 - p(e_J)$. Hence, the algorithm returns "$p(e_I \mid e_J) > q$ is true" with probability

$$\begin{aligned} r \;=\; & \min\left\{1, \frac{1}{2q}\right\} p(e_I, e_J) \\ & + \max\left\{0, \frac{1-2q}{2-2q}\right\} \big(1 - p(e_I \mid e_J)\big) p(e_J) \\ & + \frac{1}{2}(1 - p(e_J)). \end{aligned}$$

It follows that $r > \frac{1}{2}$, if and only if

$$\min\left\{1, \frac{1}{2q}\right\} p(e_I, e_J) + \max\left\{0, \frac{1-2q}{2-2q}\right\} \big(1 - p(e_I \mid e_J)\big)p(e_J) \;>\; \frac{p(e_J)}{2},$$

which is equivalent to

$$\min\left\{1, \frac{1}{2q}\right\} p(e_I \mid e_J) + \max\left\{0, \frac{1-2q}{2-2q}\right\} \big(1 - p(e_I \mid e_J)\big) \;>\; \frac{1}{2}.$$

We distinguish the two cases $q < \frac{1}{2}$ and $q \geq \frac{1}{2}$. In the first case the condition for $r > \frac{1}{2}$ reduces to

$$p(e_I \mid e_J) + \frac{1-2q}{2-2q}\big(1 - p(e_I \mid e_J)\big) \;>\; \frac{1}{2},$$

which itself reduces to $p(e_I \mid e_J) > q$. In the second case, that is, $q \geq \frac{1}{2}$, the condition for $r > \frac{1}{2}$ reduces to

$$\frac{1}{2q} p(e_I \mid e_J) \;>\; \frac{1}{2},$$

which again reduces to $p(e_I \mid e_J) > q$. That is, with probability strictly larger than $\frac{1}{2}$ the algorithm returns "$p(e_I \mid e_J) > q$ is true", if and only if it indeed holds true that $p(e_I \mid e_J) > q$. □

Lemma 13. *D-MAP is in NP^{PP}.*

Proof. We have to validate any solution $x_I \in \mathcal{X}_I$ to the D-MAP problem in polynomial time, assuming that we have access to a PP-oracle. This amounts to deciding if $p(x_I, e_J) > q$, which is a PP-complete problem by Lemma 12 if we instantiate I and J in Lemma 12 with $I = I \cup J$ and $J = \emptyset$. □

Theorem 10. *D-MPE is NP-complete, D-PoE is PP-complete, and D-MAP is NP^{PP}-complete.*

Proof. Follows directly from Theorem 9 (hardness) and Lemmas 11, 12 and 13 (membership). □

Chapter 13

Knowledge Compilation

At this point we have almost forgotten that a Bayesian network is still a multivariate categorical and thus a function $p : \mathcal{X} = \{0,1\}^n \to [0,1]$. This function can be broken down into components that are easier to evaluate, more specifically, the function can be written as an arithmetic expression with two types of primitive expressions and two types of operators:

1. For $i \in [n]$ the primitive expressions are either indicator functions of the form $\overline{x}_{i:0}$ and $\overline{x}_{i:1}$, short for $\mathbf{1}[x_i = 0]$ and $\mathbf{1}[x_i = 1]$, respectively, or parameters $p_{i,pa(i):val(i,pa(i))}$ that are derived from the conditional probability tables (CPTs) that store probabilities of the form $p(x_i \mid x_{pa(i)})$. Here, $pa(i) \subseteq [i-1]$ denotes the indices of the parents of x_i in the Bayesian network, and $val(i, pa(i))$ denotes a value in $\mathcal{X}_{i,pa(i)} = \{0,1\}^{1+|pa(i)|}$.

2. The operators are just binary addition and multiplication.

The network polynomial associated with Bayesian network is the following multilinear polynomial

$$\sum_{z \in \mathcal{X}} \prod_{i=1}^{n} p_{i,pa(i):z_i,z_{pa(i)}} \overline{x}_{i:z_i} = \sum_{z \in \mathcal{X}} \left(\prod_{i=1}^{n} p_{i,pa(i):z_i,z_{pa(i)}} \right) \left(\prod_{i=1}^{n} \overline{x}_{i:z_i} \right),$$

where $\prod_{i=1}^{n} \overline{x}_{i:z_i}$ are monomials that are multilinear in the indicator functions, and $\prod_{i=1}^{n} p_{i,pa(i):z_i,z_{pa(i)}}$ are their coefficients. For a given $x \in \mathcal{X}$, evaluating the network polynomial boils down to evaluating the indicator functions which results in selecting the coefficient of exactly one monomial, namely the monomial that corresponds to x. The value of the selected coefficient is exactly $p(x)$. The idea of the knowledge compilation approach is to find a representation of the network polynomial that supports fast inference.

The knowledge compilation approach was pioneered by Adnan Darwiche. It comprises three steps: First, encoding the network polynomial as propositional logic formula in conjunctive normal form (CNF), second, transforming the propositional formula into an equivalent normal form (sd-DNNF), and third, decoding the transformed formula into an alternative but semantically equivalent representation of the network polynomial that supports (fast) algorithms for inference queries.

CNF encoding For a structural encoding of the network polynomial, we use the following logical variables: First, for $i \in [n]$ we let $\theta_{i:0}$ be the logical variable that corresponds to the indicator function $\overline{x}_{i:0}$ and let $\theta_{i:1}$ correspond to $\overline{x}_{i:1}$. Second, the logical variables that correspond to parameters $p_{i,pa(i):x_i,x_{pa(i)}}$ are denoted as $\varphi_{i,pa(i):x_i,x_{pa(i)}}$.
The encoding of the network polynomial into a CNF formula is divided into two parts. The first part encodes the exclusiveness of states, that is, for each variable $x_i \in [n]$ it is ensured that exactly one of the two logical variables $\theta_{i:0}$ and $\theta_{i:1}$ is true. Therefore, for every variable x_i the CNF contains clauses of the form

$$\theta_{i:0} \vee \theta_{i:1} \quad \text{and} \quad \neg(\theta_{i:0} \wedge \theta_{i:1}) \equiv \neg\theta_{i:0} \vee \neg\theta_{i:1}.$$

In terms of the network polynomial these clauses encode the monomials of the polynomial. The second part encodes the coefficients of the monomials. It turns out that the coefficients share common factors. For each monomial, the encoding

$$\begin{aligned}\varphi_{i,pa(i):z_i,z_{pa(i)}} \leftrightarrow & \left(\theta_{i:z_i} \wedge \bigwedge_{j\in pa(i)} \theta_{j:z_j}\right) \\ \equiv & \left(\left(\neg\varphi_{i,pa(i):z_i,z_{pa(i)}} \vee \theta_{i:z_i}\right) \wedge \bigwedge_{j\in pa(i)} \left(\neg\varphi_{i,pa(i):z_i,z_{pa(i)}} \vee \theta_{j:z_j}\right)\right) \\ & \wedge \left(\varphi_{i,pa(i):z_i,z_{pa(i)}} \vee \neg\theta_{i:z_i} \vee \bigvee_{j\in pa(i)} \neg\theta_{j:z_j}\right)\end{aligned}$$

specifies which factors are present in its coefficients. Here we have used

$$a \leftrightarrow b \equiv (a \rightarrow b) \wedge (b \rightarrow a) \equiv (\neg a \vee b) \wedge (\neg b \vee a)$$

and the distributive law $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$.

sd-DNNF The CNF formula that encodes the network polynomial is compiled into an equivalent formula in negation normal form (NNF), that is, the formula contains only the Boolean operators $\wedge$ (conjunction) and $\vee$ (disjunction), and the negation operator $\neg$ is applied only to variables. Additionally, the compiled NNF satisfies the following three properties

- *Smoothness*: For each disjunction operator, both operands operate on the same variables (that can be negated or not).
- *Determinism:* For each disjunction operator, its two operands represent mutually exclusive propositions, that is, the propositions cannot be satisfied simultaneously.
- *Decomposability*: For each conjunction operator, its two operands have no variables in common.

We call a NNF that has these three properties a sd-DNNF (smooth deterministic decomposable negation normal form). Darwiche has designed and implemented a state-of-the-art compiler (c2d compiler) that translates a CNF formula into an equivalent small sd-DNNF formula, actually into an equivalent sd-DNNF circuit (expression tree). Here, the size of the formula is the number of Boolean operators.

Decoding sd-DNNFs The encoded network polynomial can be reconstructed from the CNF encoding by considering all *interpretations* of the CNF, that is, all truth assignments for the logical variables that satisfy the CNF. Each interpretation provides exactly one monomial of the network polynomial together with its coefficient. Therefore, the whole polynomial is reconstructed by a sum over all interpretations. This characterization suggests a different encoding of the network polynomial, namely, for each of its monomials, a conjunction of all variables whose truth assignment is *true* for the monomial and of all negated variables whose truth assignment is *false* for the monomial. The alternative encoding is then a disjunction over co-clauses, namely, one co-clause for each monomial. That is, a disjunction of the following conjunctions over all $z \in \mathcal{X}$

$$\bigwedge_{i=1}^{n} \Big(\varphi_{i,pa(i):z_i,z_{pa(i)}} \wedge \bigwedge_{\bar{z} \in \mathcal{X}_{i,pa(i)}:\bar{z} \neq z_{i,pa(i)}} \neg \varphi_{i,pa(i):\bar{z}_i,\bar{z}_{pa(i)}} \Big) \wedge \bigwedge_{i=1}^{n} \big(\theta_{i:z_i} \wedge \neg \theta_{i:1-z_i} \big)$$

is an alternative encoding in disjunctive normal form (DNF) and thus also in NNF. Furthermore, it is smooth, deterministic, and decomposable. It is smooth because each co-clause contains each variable either directly or negated, it is

deterministic since the co-clauses are mutually exclusive, and it is decomposable since each co-clause contains each variable exactly once. However, the size of the *DNF encoding* is exponential in the number n of variables, in contrast to the CNF encoding.

Observation 1. *The network polynomial is decoded* symbolically *from its DNF encoding by the following substitutions*

$$\begin{aligned}
&\wedge \longrightarrow *,\ \vee \longrightarrow + \\
&\theta_{i:0} \longrightarrow \overline{x}_{i:0},\ \theta_{i:1} \longrightarrow \overline{x}_{i:1},\ \varphi_{i,pa(i):z_i,z_{pa(i)}} \longrightarrow p_{i,pa(i):z_i,z_{pa(i)}} \\
&\neg\theta_{i:0},\ \neg\theta_{i:1},\ \neg\varphi_{i,pa(i):z_i,z_{pa(i)}} \longrightarrow 1.
\end{aligned}$$

In the following we prove that the DNF encoding is unique up to symmetries. The semantics of the decoded network polynomial is not affected by these symmetries.

Lemma 14. *The smooth, decomposable, and deterministic DNF encoding is unique up to reordering of the co-clauses within the disjunction and the literals within the co-clauses.*

Proof. Let d_1 be the DNF encoding from above and assume that there is another equivalent DNF d_2 with these properties. Then d_2 cannot have more co-clauses than d_1, because in this case there either is a co-clause c_1 in d_1 such that there are two or more co-clauses in d_2 that are satisfied by the unique satisfying assignment of truth values to the variable for c_1, contradicting determinism of d_2, or there is a co-clause in d_2 that cannot be satisfied and thus needs to have at least one variable that appears negated and non-negated, contradicting decomposability of d_2.
Furthermore, d_2 cannot have fewer co-clauses than d_1, because otherwise there must be a co-clause c_2 ind d_2 that is satisfied by at least two truth value assignments to the variables. The latter truth value assignments have corresponding co-clauses in d_1. Let c_1 and $\hat{c}_1$ be two such co-clauses and let L be the set of literals that are not shared by c_1 and $\hat{c}_1$. Let c_1 correspond to $z \in \mathcal{X}$ and $\hat{c}_1$ to $\hat{z} \in \mathcal{X}$. Here, z and $\hat{z}$ must differ in at least on bit, because $c_1 \neq \hat{c}_1$. Without loss of generality, assume that $z_i = 1$ and $\hat{z}_i = 0$ for some $i \in [n]$, then

$$\varphi_{i,pa(i):1,z_{pa(i)}},\ \varphi_{i,pa(i):0,\hat{z}_{pa(i)}} \in L.$$

The co-clause c_2 cannot contain variables that correspond to literals in L, because otherwise it cannot be satisfied simultaneously by the truth value assignments that correspond to c_1 and $\hat{c}_1$, respectively. But then there exists a truth value

assignment c_2 that simultaneously sets $\varphi_{i,pa(i):1,z_{pa(i)}}$ and $\varphi_{i,pa(i):0,\hat{z}_{pa(i)}}$ to *true* and satisfies c_2. However, there is no co-clause in d_1 that is satisfied by this truth value assignment, a contradiction.
Therefore, there is exactly one co-clause c_2 in d_2 for every co-clause in c_1 that is satisfied by the assignment of truth values that corresponds to c_1. The co-clause c_2 in d_2 must contain the same variables (negated or not) as c_1: It cannot have more variables than c_1, because c_1 has already one literal for every variable. Thus, if c_2 has more variables, there is one variable for which c_2 contains two literals (actually the same literal more than once, because otherwise the co-clause would be unsatisfiable). But in this case, d_2 is no longer decomposable. Also, the clause c_2 has to contain all the variables in c_1. Assume for the contrary that a variable is missing from c_2. Then c_2 (and thus also d_2) can be satisfied by the truth value assignment that is the interpretation encoded by c_1 as well as by switching the truth value of the missing variable in this assignment. The latter assignment must also be a satisfying assignment for d_1, because otherwise d_1 and d_2 cannot be equivalent. Therefore, there must be a co-clause $\hat{c}_1 \neq c_1$ of d_1 that encodes the latter assignment. But then d_2 cannot be deterministic, because there is also a co-clause $\hat{c}_2 \neq c_2$ in d_2 that corresponds to $\hat{c}_1$ that is also satisfied by the latter assignment of truth values.
Therefore, up to reordering of the literals, every co-clause in d_1 is also a co-clause d_2 and vice versa. □

Lemma 14 together with Observation 1 can be used to show that the encoded network polynomial can be decoded symbolically from its sd-DNNF encoding.

Theorem 11. *Applying substitutions from Observation 1 to the sd-DNNF circuit computed by a knowledgecompiler from the CNF encoding gives an arithmetic circuit that computes the network polynomial.*

Proof. We show that the sd-DNNF encoding can be transformed into the encoding DNF by repeated applications of the distributive law, namely, replacing subformulas of the form $a \wedge (b \vee c)$ by $(a \wedge b) \vee (a \wedge c)$ as long as such subformulas exist. The decoding is invariant under applications of the distributive law because the distributive law $a \cdot (b + c) = a \cdot b + b \cdot c$ applies analogously to arithmetic formulas.
The transformed propositional formula is a DNF that is equivalent to the sd-DNNF and thus also to the original CNF. Furthermore, the distributive law maintains the properties of smoothness, determinism, and decomposability: (1) Assume that $a \wedge (b \vee c)$ is smooth, deterministic, and decomposable. Then $(a \wedge b) \vee (a \wedge c)$ is smooth, because a and $b \vee c$ have no variables in common (decomposibility) and b and c operate on the same variables (smoothness) implies

that also $a \wedge b$ and $a \wedge c$ operate on the same variables. It is deterministic, because when b and c cannot be satisfied simultaneously (determinism), then also $a \wedge b$ and $a \wedge c$ cannot be satisfied simultaneously. Finally, it is also decomposable, because if a does not share variables with $b \vee c$ (decomposibility), then a cannot share variables with b and c. (2) Assume that $(a \wedge b) \vee (a \wedge c)$ is smooth, deterministic, and decomposable. Then $a \wedge (b \vee c)$ is smooth, because that $a \wedge b$ and $a \wedge c$ operate on the same variables and a has no variables in common with neither b nor c implies that b and c need to operate on the same variables. It is deterministic, because if $a \wedge b$ and $a \wedge c$ cannot be satisfied simultaneously, then also b and c that do not share variables with a cannot be satisfied simultaneously. Finally, it is decomposable, because if a does neither share variables with b nor c, then it also cannot share variables with $b \vee c$. Therefore, the resulting DNF is also smooth, deterministic, and decomposable.
Hence, by Lemma 14 and up to reordering of the co-clauses within the disjunction and the variables within the co-clauses, the resulting DNF is the DNF encoding. Thus, applying the substitution scheme directly to the sd-DNNF circuit gives an arithmetic circuit that computes the network polynomial of the Bayesian network. □

Serving Inference Queries on Compiled Circuits

Let Δ be the the arithmetic circuit that has been compiled from the Bayesian network. In the following we describe how to efficiently serve PoE and MPE queries on Δ. It turns out that both queries can be served in time linear in the size of the circuit Δ.

PoE queries A general PoE query on a Bayesian network asks to compute the probability $p(e_I \mid e_J)$, where $e_I \in \mathcal{X}_I$ is evidence for a subset $I \subseteq [n]$ and additional evidence e_J is given for another subset $J \subseteq [n]$.
By the definition of conditional probabilities, the computation of

$$p(e_I \mid e_j) = \frac{p(e_I, e_J)}{p(e_J)}$$

boils down to computing the two marginal probabilities $p(e_I, e_J)$ and $p(e_J)$. Note that in case that the evidence e_I and e_j is conflicting on the shared variables with indices in $I \cap J$, then $p(e_I, e_J) = 0$, otherwise $p(e_I, e_J) = p(e_{I \cup J})$.
Marginal probabilities $p(e_J)$ are computed by Algorithm 2.
Values are propagated from the leaves to the root in Step 4 of Algorithm 2 as follows: Any node that is labeled by an indicator function propagates its value,

Algorithm 2 Marginal probability

input circuit Δ and evidence e_I
output $p(e_I)$
1: for $i \in I, i' \in \{0,1\}$: replace $\overline{x}_{i:i'}$ by the value $\mathbf{1}[e_i = i']$
2: for $j \in [n] \setminus I, j' \in \{0,1\}$: replace $\overline{x}_{j:j'}$ by the value 1
3: propagate values from leaves to root

that is, either 0 or 1, any parameter node its value, any product node the product of its childrens' values, and any summation node the sum of its children's values. The correctness of Algorithm 2 follows from the transformation rule, namely, replacing subexpressions of the form $a \wedge (b \vee c)$ by $(a \wedge b) \vee (a \wedge c)$, that we have used the proof of Theorem 11. In the corresponding arithmetic circuit the transformation rule means replacing subexpressions of the form $a \cdot (b + c)$ by $(a \cdot b) + (a \cdot c)$. Whenever a subexpression a, b or c is not compatible with the evidence, then its value is 0. Therefore, the compound expressions $(a \cdot b), (a \cdot c)$, and $(a \cdot b) + (a \cdot c)$, respectively, are zero if they include an incompatible subexpression. That is, summands in a subexpressions are 0 when at least one of their factors is 0. The elementary factors are indicator and parameter variables. Hence, any elementary factor with value 0 eliminates any summand in which it is contained. The expression that is represented by the root of the arithmetic circuit is (semantically equivalent to) the network polynomial. By replacing indicator nodes $\overline{x}_{i:i'}$ for $i \in I$, $i' \in \{0,1\}$ by the values $\mathbf{1}[e_i = i']$ and the indicator nodes $j \in [n] \setminus I, j' \in \{0,1\}$ by the value 1, we select exactly the summands in the network polynomial that are compatible with the evidence e_I and thus an expression for the marginal probability $p(e_I)$.

MPE queries A general MPE query on a Bayesian network means solving the problem

$$\mathsf{argmax}_{x_I}\ p(x_I \mid e_J),$$

for a subset $J \subseteq [n]$, evidence $e_J \in \mathcal{X}_J$, and $I = [n] \setminus J$. An alternative way to look at MPE queries is selecting the monomials from the network polynomial that have maximal coefficients.
The proof of Theorem 11 shows that the sd-DNNF circuit computes shared factors of the monomials' coefficients at once, which is computationally efficient. However, the sd-DNNF can also be specialized to compute individual monomials. Note that the monomials are in a one-to-one correspondence with the elements of the sample space $\mathcal{X}$. Hence, from the monomials with maximal coefficients we immediately get the maximizing states in $\mathcal{X}$. Algorithmically, we want to

exploit another one-to-one correspondence, namely, a one-to-one correspondence between the monomials and complete subcircuits of the sd-DNNF encoding. A subcircuit of the sd-DNNF circuit is called *complete* if it computes one monomial of the network polynomial. In the proof of Theorem 11 we transform the sd-DNNF by repeated applications of the distributive law. We obtain complete subcircuits from the sd-DNNF circuit by modifying the transformation rule. Instead of replacing subexpressions of the form $a \wedge (b \vee c)$ by $(a \wedge b) \vee (a \wedge c)$, we replace them either by $(a \wedge b)$ or by $(a \wedge c)$. Hence, any replacement removes one disjunction operator. Removing all disjunctions in this way results in a complete subcircuit. We now prove the following correspondence.

Lemma 15. *There is a one-to-one correspondence between the monomials of the network polynomial and the complete subcircuits of its sd-DNNF encoding.*

Proof. In the proof of Theorem 11, transformations by the distributive law maintain the number of disjunctions. Thus, the DNF encoding and the sd-DNNF have the same number of disjunctions, which is one less than the number of monomials of the network polynomial. Hence, removing one disjunction, that is, replacing subexpressions of the form $a \wedge (b \vee c)$ by either $(a \wedge b)$ or by $(a \wedge c)$, removes one monomial. The property of smoothness makes sure that the remaining monomials are proper, that is, of the form $\prod_{i=1}^{n} \overline{x}_{i:z_i}$ for some $z = (z_1, \ldots, z_n) \in \mathcal{X}$. This establishes the one-to-one correspondence. □

For serving MPE queries, it is sufficient to compute the complete subcircuits of the sd-DNNF with maximal value, which is done by Algorithm 3. The algorithm uses a specific rule to choose among the two possible replacements of subexpressions of the form $a \wedge (b \vee c)$ in Lemma 15 for serving MPE queries.

Algorithm 3 Most probable explanations (MPE)

input circuit Δ and evidence e_J
output $\text{argmax}_{x_I} p(x_I \mid e_J)$

1: for $j \in J, j' \in \{0,1\}$: replace $\overline{x}_{j:j'}$ by the value $\mathbf{1}[e_j = j']$
2: for $i \in I = [n] \setminus J, i' \in \{0,1\}$: replace $\overline{x}_{i:i'}$ by the value 1
3: replace sum nodes by maximization nodes
4: upward pass: calculate values of sub-circuits
5: downward pass: collect MPEs

The main work of Algorithm 3 takes place in Steps 4 and 5, after replacing the indicator nodes $\overline{x}_{j:j'}$ for $j \in J$, $j' \in \{0,1\}$ by the values $\mathbf{1}[e_j = j']$ and the indicator nodes $i \in [n] \setminus J, i' \in \{0,1\}$ by the value 1, and replacing the sum nodes of Δ by maximization nodes.

In the upward pass (Step 4 of Algorithm 3), starting from leaf nodes, the values of sub-circuits are propagated to the root node. Any node that is labeled by an indicator variable propagates its value, that is, either 0 or 1, any parameter node its value, any product node the product of its childrens' values, and any maximization node the maximum value of its children.
After reaching the root, MPE states are collected in a downward pass (Step 5 of Algorithm 3). At any a maximization node, all children with the highest value are followed, at a product node all children are followed. Whenever a node that is labeled by an indicator variable is reached, the variable is collected. Parameter nodes are ignored in the downward pass since we are only interested in collecting MPE states. Note that MPE states are not necessarily unique. Here, we have described Algorithm 3 such that it returns all MPE states.

Theorem 12. *Algorithm 3 computes all MPE states in time that is linear in the size of the compiled sd-DNNF circuit.*

Proof. The running time of the algorithm is determined by Steps 4 and 5 that are linear in the size of the circuit.
For the correctness of the algorithm, first note that all nodes of the circuit that are labeled by parameters $p_{i,pa(i):x_i,x_{pa(i)}}$ are instantiated with positive values (zeroes can be eliminated directly). Also, the nodes labeled by indicator variables $\overline{x}_{i:i'}$ are instantiated with non-negative values, namely either 0 or 1.
Step 5 of the algorithm computes a set of complete subcircuits of the maximization-circuit. Hence, it follows from Lemma 15 that Step 5 selects a set of monomials of the network polynomial, or more specifically the corresponding states in $\mathcal{X}$. It remains to show that the selected states are indeed maximizers.
By construction the complete subcircuits that are selected in Step 5 all have the same value. Here, the values are the coefficients of the corresponding monomials, except when the value of the monomial itself is zero, because it is not compatible with the evidence e_J. The coefficients factorize into non-negative factors that are shared between different coefficients. The factors, that is, values of sub-circuits, are propagated through the maximization circuit starting from the leaves (Step 4). At multiplication nodes, the factors are simply multiplied. At maximization nodes the maxima among alternative factors are selected.
Let s_1 be a non-selected complete subcircuit. The subcircuit s_1 differs from all selected subcircuits by following a non-maximal child at at least one maximization node in the downward pass (Step 5). Among all these maximization nodes, consider one that is closest to the root of s_1 and let s_2 be a selected subcircuit that differs from s_1 by following a different child at the considered maximization node. The coefficients for the monomials that correspond to s_1 and s_2, respectively, split into two factors. The first factor is computed in the upward pass (Step 4) by

the different children of the maximization node for s_1 and s_2, respectively. The second factor is computed also in the upward pass but after the maximization node. The circuits share the second factor and the first factor must be larger for s_2 than for s_1 since otherwise the child that corresponds to s_1 would have (also) been selected at the considered maximization node. Thus, since all factors are non-negative, the value of s_2 must be larger than the value of s_1. It follows that the values of all the selected complete subcircuits and thus the selected states are maximal. □

CNF Encoding of Pairwise Categoricals

Finally, we want to come back to pairwise models p on the sample space $\mathcal{X} = \prod_{i=1}^{n} \mathcal{X}_i$ with $\mathcal{X}_i = [n_i]$. In unnormalized, functional form these models are given as $\exp\left(\overline{x}^\top Q\, \overline{x}\right)$, where Q is a symmetric block matrix and $\overline{x}$ for $x \in \mathcal{X}$ are vectors of indicator functions

$$\overline{x} = (\overline{x}_1, \ldots, \overline{x}_n) \in \{0,1\}^{\sum_{i=1}^{n} n_i}$$

with $\overline{x}_i = (\overline{x}_{i:1}, \ldots, \overline{x}_{i:n_i}) \in \{0,1\}^{n_i}, i \in [n]$, where $\overline{x}_{i:i'} = \mathbf{1}[x_i = i']$ for $i' \in [n_i]$.

Pairwise interactions of discrete variables are always determined by a group of parameters, because interactions are defined on their indicator variables. Therefore, the parameter matrix Q exhibits the following group structure

$$Q = \left(\begin{array}{c|c|c} Q_{11} & \cdots & Q_{1n} \\ \hline \vdots & \ddots & \vdots \\ \hline Q_{n1} & \cdots & Q_{nn} \end{array}\right),$$

where $Q_{ij} = Q_{ji}^\top \in \mathbb{R}^{n_i \times n_j}$ describes the interaction between the variables x_i and x_j. As before, we constrain the interaction parameters that are associated with the indicator functions $\overline{x}_{i:1}, i \in [n]$ to zero for identifiability reasons.

In the following we express the function $\exp\left(\overline{x}^\top Q\, \overline{x}\right)$ by a multilinear polynomial (network polynomial) in the indicator functions, that is,

$$\sum_{z \in \mathcal{X}} c_z \cdot \prod_{i=1}^{n} \overline{x}_{i:z_i},$$

where c_Z is the coefficient of the monomial $\prod_{i=1}^n \overline{x}_{i:z_i}$. We get

$$\begin{aligned}
\exp\left(\overline{x}^\top Q\,\overline{x}\right) &= \exp\left(\sum_{i,j=1}^n \overline{x}_i^\top Q_{ij}\,\overline{x}_j\right) = \prod_{i,j=1}^n \exp\left(\overline{x}_i^\top Q_{ij}\,\overline{x}_j\right) \\
&= \prod_{i,j=1}^n \exp\left(\sum_{i'=1}^{n_i}\sum_{j'=1}^{n_j} \overline{x}_{i:i'} \cdot \overline{x}_{j:j'} \cdot q_{ij:i'j'}\right) \\
&= \prod_{i,j=1}^n \sum_{i'=1}^{n_i}\sum_{j'=1}^{n_j} \overline{x}_{i:i'} \cdot \overline{x}_{j:j'} \cdot \exp\left(q_{ij:i'j'}\right) \\
&= \prod_{i,j=1}^n \sum_{i'=1}^{n_i}\sum_{j'=1}^{n_j} \overline{x}_{i:i'} \cdot \overline{x}_{j:j'} \cdot \hat{q}_{ij:i'j'},
\end{aligned}$$

where we have used the shorthand notation $\hat{q}_{ij:i'j'} = \exp\left(q_{ij:i'j'}\right)$ and exploited that there is exactly one pair $(i', j') \in [n_i] \times [n_j]$ at which the product of indicator functions $\overline{x}_{i:i'} \cdot \overline{x}_{j:j'}$ can take the value 1, at all other pairs it is 0. By evaluating $\exp\left(\overline{x}^\top Q\,\overline{x}\right)$ at $z = (z_1, \ldots, z_n) \in \mathcal{X} = \prod_{i=1}^n [n_i]$, we get that the coefficient of the monomial $\prod_{i=1}^n \overline{x}_{i:z_i}$ is given as $\prod_{i,j=1}^n \hat{q}_{ij:z_i z_j}$. Therefore, we get

$$\exp\left(\overline{x}^\top Q\,\overline{x}\right) = \sum_{z\in\mathcal{X}} \left(\prod_{i,j=1}^n \hat{q}_{ij:z_i z_j}\right)\left(\prod_{i=1}^n \overline{x}_{i:z_i}\right).$$

CNF encoding of the unnormalized distribution For the encoding, we use the following logical variables: First, for $i \in [n]$, $i' \in [n_i]$ we let $\theta_{i:i'}$ be the variable that corresponds to the indicator variable $\overline{x}_{i:i'}$. Second, for $i, j \in [n]$, $i' \in [n_i]$, $j' \in [n_j]$ we let $\varphi_{ij:i'j'}$ be the variable that corresponds to the interaction parameter $\hat{q}_{ij:i'j'}$.

The encoding can be derived from the representation the multilinear polynomial representation of $\exp\left(\overline{x}^\top Q\,\overline{x}\right)$. The encoding is divided into two parts. The first part

$$\bigwedge_{i=1}^n \left(\left(\bigvee_{i'=1}^{n_i} \theta_{i:i'}\right) \wedge \left(\bigwedge_{i'<i''\in[n_i]} (\neg\theta_{i:i'} \vee \neg\theta_{i:i''})\right)\right)$$

encodes the exclusiveness of states for the monomials $\prod_{i=1}^n \overline{x}_{i:z_i}$, that is, for each variable $i \in [n]$ it is ensured that exactly one of the indicator variables $\theta_{i:i'}$, $i' \in [n_i]$ is *true*. The second part

$$\bigwedge_{i,j=1}^n \bigwedge_{i'=1}^{n_i} \bigwedge_{j'=1}^{n_j} \left((\theta_{i:i'} \wedge \theta_{j:j'}) \leftrightarrow \varphi_{ij':i'j'}\right),$$

where $a \leftrightarrow b$ is shorthand for $(\neg a \vee b) \wedge (a \vee \neg b)$, and $(a \wedge b) \leftrightarrow c$ for $(\neg a \vee \neg b \vee c) \wedge (a \vee \neg c) \wedge (b \vee \neg c)$, encodes the coefficients $\prod_{i,j=1}^{n} \hat{q}_{ij:z_i z_j}$ of the monomials. Clauses from the second part must only be included in the CNF if $\hat{q}_{ij:i'j'} \neq 1$, or equivalently, if the corresponding entry $q_{ij:i'j'}$ of the matrix block Q_{ij} is non-zero. Thus, sparsity in the pairwise model directly translates into a smaller CNF.

Serving inference queries As in the case of binary Bayesian networks, the CNF encoding can be compiled into an sd-DNNF circuit. The Algorithm 2 that is used for serving PoE queries is easily adapted for use with sd-DNNF circuits Δ for pairwise models. The only change is that i' in Line 1 and j' in Line 2 may run over more than two values.

Algorithm 4 Marginal probability

input circuit Δ and evidence e_I
output $p(e_I)$
1: for $i \in I, i' \in [n_i]$: replace $x_{i:i'}$ by the value $\mathbf{1}[e_i = i']$
2: for $j \in [n] \setminus I, j' \in [n_j]$: replace $x_{j:j'}$ by the value 1
3: propagate values from leaves to root

Note that calling the adapted Algorithm 2 without evidence, that is, $I = \emptyset$, returns the normalization constant for the unnormalized circuit Δ. However, for serving PoE queries that compute $p(e_I \mid e_J)$ it is enough to compute the unnormalized marginals $p(e_I, e_J)$ and $p(e_J)$. The circuit also does not need to be normalized for serving MPE queries by adapting Algorithm 3. Again the only change is that i' in Line 1 and j' in Line 2 now may run over more than two values.

Algorithm 5 Most probable explanations (MPE)

input circuit Δ and evidence e_J
output $\text{argmax}_{x_I} p(x_I \mid e_J)$
1: for $j \in J, j' \in [n_j]$: replace $x_{j:j'}$ by the value $\mathbf{1}[e_j = j']$
2: for $i \in I = [n] \setminus J, i' \in [n_i]$: replace $x_{i:i'}$ by the value 1
3: replace sum nodes by maximization nodes
4: upward pass: calculate values of sub-circuits
5: downward pass: collect MPEs

Chapter 14

Variational Inference

In variational inference, hard inference queries on a target distribution are delegated to a member from a parametrized family, called variational family, of distributions that support efficient inference. The distribution to which the inference is delegated is chosen to be close to the target distribution. A standard measure of closeness of probability distributions that are defined on the same sample space is the *Kullback-Leibler divergence*.

Kullback-Leibler Divergence and Evidence Lower Bound (ELBO)

Let p and $\hat{p}$ be probability distributions defined on the same sample space $\mathcal{X}$. For strictly positive $\hat{p}$, the *Kullback-Leibler divergence* from $\hat{p}$ to p is defined as

$$d(p\|\hat{p}) \;=\; -\sum_{x\in\mathcal{X}} p(x)\log\frac{\hat{p}(x)}{p(x)} \;=\; -\sum_{x\in\mathcal{X}} p(x)\log\hat{p}(x) \;-\; \mathsf{H}(p).$$

The Kullback-Leibler divergence is minimized at p, which can be seen from the Lagrange multiplier theorem that respects the constraint $\sum_{x\in\mathcal{X}}\hat{p}(x)=1$. We get

$$\frac{\partial d(p\|\hat{p})}{\partial\hat{p}(x)} \;=\; -\frac{p(x)}{\hat{p}(x)} \;=\; \lambda, \text{ with Lagrange multiplier } \lambda,$$

and thus $p(x)=-\lambda\hat{p}(x)$, and

$$1 \;=\; \sum_{x\in\mathcal{X}} p(x) \;=\; -\lambda\sum_{x\in\mathcal{X}}\hat{p}(x) \;=\; -\lambda.$$

Since the Hessian of $d(p\|\hat{p})$ is the diagonal matrix $\mathrm{diag}\big(\mathbf{p}\oslash(\hat{\mathbf{p}}\otimes\hat{\mathbf{p}})\big)$, which is positive semi-definite, the Kullback-Leibler divergence is convex and thus minimized at $\hat{p}=p$. Furthermore, from $d(p\|p)=\mathsf{H}(p)-\mathsf{H}(p)=0$ we get that

always $d(p\|\hat{p}) \geq 0$. However, the Kullback-Leibler divergence is, in general, not symmetric.

Given the target distribution

$$p(x) \;=\; \exp\Big(\sum_{I\in\mathcal{I}} q_I(x_I) \;-\; a(\mathbf{q})\Big),$$

a multivariate categorical on the sample space $\mathcal{X}$, where $\mathcal{I} \subseteq 2^{[n]}$ is a set of multi-index sets, and a variational family of distributions $p_\theta, \theta \in \Theta$ on $\mathcal{X}$. We have

$$\begin{aligned} 0 \;\leq\; d(p_\theta\|p) \;&=\; -\sum_{x\in\mathcal{X}} p_\theta(x)\log p(x) \;-\; \mathsf{H}(p_\theta) \\ &=\; -\sum_{x\in\mathcal{X}} p_\theta(x)\Big(\sum_{I\in\mathcal{I}} q_I(x_I) \;-\; a(\mathbf{q})\Big) \;-\; \mathsf{H}(p_\theta) \\ &=\; a(\mathbf{q}) \;-\; \sum_{x\in\mathcal{X}}\sum_{I\in\mathcal{I}} p_\theta(x) q_I(x_I) \;-\; \mathsf{H}(p_\theta) \\ &=\; a(\mathbf{q}) \;-\; \mathsf{E}_{p_\theta}\Big[\sum_{I\in\mathcal{I}} q_I\Big] \;-\; \mathsf{H}(p_\theta), \end{aligned}$$

or equivalently,

$$\sum_{I\in\mathcal{I}} \mathsf{E}_{p_\theta}[q_I] \;+\; \mathsf{H}(p_\theta) \;\leq\; a(\mathbf{q}),$$

where

$$\sum_{I\in\mathcal{I}} \mathsf{E}_{p_\theta}[q_I] \;+\; \mathsf{H}(p_\theta)$$

is called the *evidence lower bound (ELBO)*. One advantage of maximizing the ELBO instead of minimizing the Kullback-Leibler divergence is that for the ELBO the distribution p does not need to be normalized, that is, we do not need to know the often costly to compute log-normalization constant $a(\mathbf{q})$.

In variational inference, we approximate the distribution p by a distribution $\hat{p} \in \{p_\theta \;:\; \theta \in \Theta\}$ that maximizes the ELBO. All inference queries are then served on $\hat{p}$ instead of p.
The choice to minimize $d(p_\theta\|p)$ instead of $d(p\|p_\theta)$ means that we choose p_θ to which p has the smallest KL distance (Kullback-Leibler divergence), but not a p_θ that has the smallest KL distance to p. A justification for this choice is computational, namely, minimizing $d(p_\theta\|p)$ instead of $d(p\|p_\theta)$, or maxizing the corresponding ELBOs, requires computing expectations with respect to

p_θ instead of expectations with respect to p. By our assumption, however, computations with respect to p are practically infeasible.
The difference between minimizing $d(p_\theta\|p)$ and minimizing $d(p\|p_\theta)$ is that, in the first case, $p(x) = 0$ implies $p_\theta(x) = 0$, because otherwise $d(p_\theta\|p)$ would be infinite, whereas, in the second case, $p(x) > 0$ implies $p_\theta(x) > 0$, because $d(p\|p_\theta)$ would be infinite for $p_\theta(x) = 0$. That is, in the first case the support of the optimal p_θ is a subset of the support of p, whereas, in the second case, it is a superset of the support of p. Therefore, we tend to overestimate the support of p in the first case, and tend to underestimate its support in the second case.

In the following we assume for the target distribution that

$$q_I \;=\; \sum_{z\in\mathcal{X}_I} \hat{q}_z \prod_{i\in I} \overline{x}_{i:z_i} \text{ for } I \in \mathcal{I},$$

with indicator functions $\overline{x}_{i:i'}$.

In the following we discuss two variational families of distributions $p_\theta, \theta \in \Theta$ on $\mathcal{X}$, namely, the mean field family and families that are derived from sd-DNNF circuits. For maximizing the ELBO we use some numerical optimization algorithm that iteratively needs to evaluate the ELBO and its gradient at different parameter values θ. Therefore, we need to discuss how to compute the ELBO and its gradient for the two families.

Mean-field Variational Families

The probably simplest variational family is given by the family of full factored distribution on $\mathcal{X}$, that is, $p_\theta(x) = \prod_{i=1}^n p_i(x_i)$, where the parameters θ are the probabilities $\big(p_i(x_i)\big)_{x_i\in\mathcal{X}_i}$ for $i \in [n]$.

The expectation term of the ELBO for the mean-field family is given as

$$\begin{aligned}
\sum_{I\in\mathcal{I}} \mathsf{E}_{p_\theta}[q_I] \;&=\; \sum_{I\in\mathcal{I}} \sum_{z\in\mathcal{X}_I} \hat{q}_z \mathsf{E}_{p_\theta}\Big[\prod_{i\in I} \overline{x}_{i:z_i}\Big] \\
&= \sum_{I\in\mathcal{I}} \sum_{z\in\mathcal{X}_I} \hat{q}_z \sum_{x\in\mathcal{X}_I} p_\theta(x) \prod_{i\in I} \overline{x}_{i:z_i}(x) \\
&= \sum_{I\in\mathcal{I}} \sum_{z\in\mathcal{X}_I} \hat{q}_z \sum_{x\in\mathcal{X}_I} \prod_{j=1}^{n} p_j(x_j) \prod_{i\in I} \overline{x}_{i:z_i}(x).
\end{aligned}$$

The derivative with respect to the parameter $p_k(\hat{x}_k)$ is then given as

$$\sum_{I \in \mathcal{I}} \sum_{z \in \mathcal{X}_I} \hat{q}_z \mathsf{E}_{p_{-k}} \Big[\prod_{i \in I} \overline{x}_{i:z_i} \Big]_{x_k = \hat{x}_k},$$

where $p_{-k}(x) = \prod_{j=1\,:\,j \neq k}^{n} p_j(x_j)$ and

$$\mathsf{E}_{p_{-k}} \Big[\prod_{i \in I} \overline{x}_{i:z_i} \Big]_{x_k = \hat{x}_k} = \sum_{x \in \mathcal{X}_{I \setminus \{k\}}} \prod_{j=1\,:\,j \neq k}^{n} p_j(x_j) \prod_{i \in I} \overline{x}_{i:z_i}(x_{I \setminus \{k\}}, \hat{x}_k).$$

The entropy term of the ELBO for the mean-field family is given as

$$\begin{aligned} \mathsf{H}(p_\theta) \;&=\; -\mathsf{E}_{p_\theta}\big[\log p_\theta\big] \;=\; -\sum_{x \in \mathcal{X}} p_\theta(x) \log p_\theta(x) \\ &=\; -\sum_{x \in \mathcal{X}} \prod_{j=1}^{n} p_j(x_j) \log \prod_{j=1}^{n} p_j(x_j) \;=\; -\sum_{x \in \mathcal{X}} \prod_{j=1}^{n} p_j(x_j) \sum_{j=1}^{n} \log p_j(x_j). \end{aligned}$$

By using the product rule, the derivative with respect to the parameter $p_k(\hat{x}_k)$ is then given as

$$\sum_{j=1}^{n} \mathsf{E}_{p_{-k}}\big[\log p_j\big] \;-\; \sum_{x \in \mathcal{X}_{I \setminus \{k\}}} \prod_{j=1\,:\,j \neq k}^{n} p_j(x_j) \;=\; \sum_{j=1}^{n} \mathsf{E}_{p_{-k}}\big[\log p_j\big] \;-\; 1.$$

Remark: With the mean field approximation we can efficiently but approximately answer inference queries, such as, PoE queries of the form compute $p(e_I)$ for $I \subseteq [n]$ and MAP queries of the form $\mathsf{argmax}_{x_I \in \mathcal{X}_I} p(x_I)$ for $I \subseteq [n]$.

sd-DNNF Variational Families

We assume that the variational distributions $p_\theta, \theta \in \Theta$ are represented by a smooth, deterministic, and decomposable arithmetic circuit Δ, where the parameters θ are the parameter nodes of the circuit Δ. That is, we fix the structure of the arithmetic circuit and only optimize the parameters. The properties of smoothness, determinism, and decomposibility can be translated directly from NNF circuits to arithmetic circuits.

- *Smoothness*: For every addition operator, both operands operate on the same variables.
- *Determinism*: For every addition operator, at most one of its arguments is non-zero.

- *Decomposibility*: For every multiplication operator, its two operands have no variables in common.

We show that the terms of the ELBO can be computed recursively over the structure of the arithmetic circuit Δ that has sums and products as inner nodes and parameters and indicator functions at its leaves. That is, we present a recursive algorithm for computing the ELBO. From this algorithm we can get an algorithm for computing the gradient by automatic differentiation.

We compute the terms of the ELBO, that is, the entropy $\mathsf{H}(p_\theta)$ and the expectations $\mathsf{E}_{p_\theta}[q_I]$, individually, and start with a discussion of the expectation terms.

Expectation terms We show how to compute the contribution of the expectation terms

$$\mathsf{E}_{p_\theta}[q_I] = \sum_{z \in \mathcal{X}_I} \hat{q}_z \mathsf{E}_{p_\theta}\left[\prod_{i \in I} \overline{x}_{i:z_i}\right]$$

to the ELBO. Actually, we show how to recursively compute $\mathsf{E}_{p_\theta}\left[\prod_{i \in I} \overline{x}_{i:i'}\right]$. The recursive strategy will make sure that at each node of the arithmetic circuit Δ the expectation of a product of indicator functions will be computed, where all indices of the indicators are in the scope of the node at which the recursion will continue.

We begin our discussion by considering a sum node v of the circuit Δ. For the node v, let $I(v) \subseteq [n]$ be the set of indices of the variables on which the subcircuit that is rooted at v operates. The subcircuit rooted at v computes p_v, which is a not necessarily normalized probability distribution on $I(v)$. Therefore, we slightly abuse the notation here and in the following, and compute expectations with respect to $p(v)$ as well as its entropy.

$$\begin{aligned}
\mathsf{E}_{p_v}\left[\prod_{i \in I(v)} \overline{x}_{i:i'}\right] &= \sum_{x \in \mathcal{X}_{I(v)}} p_v(x) \prod_{i \in I(v)} \overline{x}_{i:i'}(x) \\
&= \sum_{x \in \mathcal{X}_{I(v)}} \sum_{u \in ch(v)} p_u(x) \prod_{i \in I(v)} \overline{x}_{i:i'}(x) \\
&= \sum_{u \in ch(v)} \sum_{x \in \mathcal{X}_{I(v)}} p_u(x) \prod_{i \in I(v)} \overline{x}_{i:i'}(x) \\
&= \sum_{u \in ch(v)} \sum_{x \in \mathcal{X}_{I(u)}} p_u(x) \prod_{i \in I(u)} \overline{x}_{i:i'}(x) = \sum_{u \in ch(v)} \mathsf{E}_{p_u}\left[\prod_{i \in I(u)} \overline{x}_{i:i'}\right],
\end{aligned}$$

where we have used for the fourth equality that Δ is smooth, that is, $I(v) = I(u)$ for all children u of v.

Next we consider a multiplication node v, which can be written as

$$\begin{aligned}
\mathsf{E}_{p_v}\Big[\prod_{i\in I(v)} \overline{x}_{i:i'}\Big] &= \sum_{x\in\mathcal{X}_{I(v)}} p_v(x) \prod_{i\in I(v)} \overline{x}_{i:i'}(x) \\
&= \sum_{x\in\mathcal{X}_{I(v)}} \Big(\prod_{u\in ch(v)} p_u(x)\Big)\Big(\prod_{i\in I(v)} \overline{x}_{i:i'}(x)\Big) \\
&= \sum_{x\in\mathcal{X}_{I(v)}} \Big(\prod_{u\in ch(v)} p_u(x_{I(u)})\Big)\Big(\prod_{u\in ch(v)}\prod_{i\in I(u)} \overline{x}_{i:i'}(x)\Big) \\
&= \sum_{x\in\mathcal{X}_{I(v)}} \prod_{u\in ch(v)} \Big(p_u(x_{I(u)}) \prod_{i\in I(u)} \overline{x}_{i:i'}(x)\Big) \\
&= \prod_{u\in ch(v)} \sum_{x\in\mathcal{X}_{I(u)}} p_u(x_{I(u)}) \prod_{i\in I(u)} \overline{x}_{i:i'}(x) \\
&= \prod_{u\in ch(v)} \mathsf{E}_{p_u}\Big[\prod_{i\in I(u)} \overline{x}_{i:i'}\Big],
\end{aligned}$$

where the third and fifth equality follows from the decomposibility of Δ, that is, $I(v)$ is the union of the pairwise disjoint $I(u)$ of the children u of v. For the fifth equality, observe that we have in general that

$$\begin{aligned}
\sum_x \sum_y \sum_z a(x)b(y)c(z) &= \sum_x a(x) \sum_y b(y) \sum_z c(z) \\
&= \Big(\sum_x a(x)\Big)\Big(\sum_y b(y)\Big)\Big(\sum_z c(z)\Big).
\end{aligned}$$

We need to take care of the case that u is a parameter node for the parameter θ_u, because in this case we have $I(u) = \emptyset$. To take care of this case, we define $\mathsf{E}_{p_u}\Big[\prod_{i\in I(u)=\emptyset} \overline{x}_{i:i'}\Big] = \theta_u$.

Finally, the expectation $\mathsf{E}_u\big[\overline{x}_{i:i'}\big]$ of an indicator function $\overline{x}_{i:i'}$ at an indicator node $u = \overline{x}_{i:i''}$ is $\mathbf{1}[i' = i'']$. Note that the index i is in the scope of u by the invariant of the recursion. This concludes the recursive computation of the expectation terms that start at the root of the circuit Δ.

Entropy term We show that the contribution of the entropy term to the ELBO can also be computed recursively and start by considering a sum node v of the circuit Δ. The entropy at v is given as $\mathsf{H}(p_v) = -\sum_{x \in \mathcal{X}_{I(v)}} p_v(x) \log p_v(x)$, where $I(v)$ are again the indices in the scope of p_v, that is, the indices of the variables that p_v depends on. The entropy at v can be written as

$$\begin{aligned}
\mathsf{H}(p_v) &= -\sum_{x \in \mathcal{X}_{I(v)}} p_v(x) \log p_v(x) = -\sum_{x \in \mathcal{X}_{I(v)}} \sum_{u \in ch(v)} p_u(x) \log \Big(\sum_{u \in ch(v)} p_u(x) \Big) \\
&= -\sum_{x \in \mathcal{X}_{I(v)}} \sum_{u \in ch(v)} p_u(x) \log p_u(x) = -\sum_{u \in ch(v)} \sum_{x \in \mathcal{X}_{I(v)}} p_u(x) \log p_u(x) \\
&= -\sum_{u \in ch(v)} \sum_{x \in \mathcal{X}_{I(u)}} p_u(x) \log p_u(x) = \sum_{u \in ch(v)} \mathsf{H}(p_u),
\end{aligned}$$

where we have used for the third equality that Δ is deterministic and for the fifth equality that Δ is smooth.

Next, we consider the entropy at a multiplication node v, which can be written as follows

$$\begin{aligned}
\mathsf{H}(p_v) &= -\sum_{x \in \mathcal{X}_{I(v)}} p_v(x) \log p_v(x) \\
&= -\sum_{x \in \mathcal{X}_{I(v)}} \Big(\prod_{u \in ch(v)} p_u(x) \Big) \log \Big(\prod_{u \in ch(v)} p_u(x) \Big) \\
&= -\sum_{x \in \mathcal{X}_{I(v)}} \Big(\Big(\prod_{u \in ch(v)} p_u(x) \Big) \sum_{u \in ch(v)} \log p_u(x) \Big) \\
&= -\sum_{x \in \mathcal{X}_{I(v)}} \Big(\sum_{u \in ch(v)} \Big(\big(p_u(x) \log p_u(x) \big) \prod_{w \in ch(v) \setminus \{u\}} p_w(x) \Big) \Big) \\
&= -\sum_{u \in ch(v)} \sum_{x \in \mathcal{X}_{I(v)}} \Big(p_u(x) \log p_u(x) \prod_{w \in ch(v) \setminus \{u\}} p_w(x) \Big) \\
&= -\sum_{u \in ch(v)} \sum_{x \in \mathcal{X}_{I(v)}} \Big(\big(p_u(x_{I(u)}) \log p_u(x_{I(u)}) \big) \prod_{w \in ch(v) \setminus \{u\}} p_w(x_{I(w)}) \Big) \\
&= \sum_{u \in ch(v)} \mathsf{H}(p_u) \prod_{w \in ch(v) \setminus \{u\}} \sum_{x \in \mathcal{X}_{I(w)}} p_w(x) \\
&= \sum_{u \in ch(v)} \mathsf{H}(p_u) \prod_{w \in ch(v) \setminus \{u\}} a(p_w),
\end{aligned}$$

where the function $a(p_w)$ denotes the normalizer at node w of Δ and the sixth equality follows from the decomposibility of Δ, that is, the scopes $I(u)$ of the children of product nodes are disjoint and thus the sum over all $x \in \mathcal{X}_{I(v)}$ can be split into sums over $x_{I(u)} \in \mathcal{X}_{I(u)}$ for the children u of v. That is, also at multiplication nodes the entropy can be computed from the entropy of its children, however, the computations are not straightforward and also involve the computations of normalizers for the children. Note, again, that we have $I(u) = \emptyset$ if u is a parameter node for the parameter θ_u. Therefore, we define $\mathsf{H}(p_u) = -\theta \log \theta$.

Finally, the entropy $\mathsf{H}(p_u)$ of an indicator node $u = \overline{x}_{i:i'}$ is

$$-\sum_{i'' \in \mathcal{X}_i} \overline{x}_{i:i'}(i'') \log \overline{x}_{i:i'}(i'') \;=\; 0,$$

because $1 \cdot \log(1) = 0$ and $0 \cdot \log(0) := 0$. This concludes the recursive computation of the expectation terms that start at the root of the circuit Δ.

Chapter 15

Gibbs Sampling

The three operations *marginalization*, *conditioning*, and *maximization* are at the core of most probabilistic inferences. All three operations have well known counterparts on data base tables (data frames), namely, *projection* for marginalization, *selection* for conditioning, and *aggregation* for maximization. In SQL (structured query language) projections are done by `SELECT` statement, where the variables that are *not* marginalized out have to be specified. Selections are done by `WHERE` statement, in which in the condition is directly expressed.

Data points $x^{(1)}, \ldots, x^{(m)}$ that are (independently) drawn from a joint probability distribution p on $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ can be stored in a database table

X_1	$\ldots$	X_n
$x_1^{(1)}$	$\ldots$	$x_n^{(1)}$
$\vdots$		$\vdots$
$x_1^{(m)}$	$\ldots$	$x_n^{(m)}$

If we denote the database table by T, then our most complex inference query type, MAP queries such as $\operatorname{armin}_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p\big(x_1, x_2 \mid e_3, e_4\big)$, can be served on T by SQL queries. For instance, the example MAP query is served by the following SQL query,

```
SELECT MOST_FREQUENT(X1,X2)
FROM T
WHERE X3 = e3 AND X4 =e4
```

Here, the `SELECT` statement selects the specified columns, the `WHERE` statement selects the rows that satisfy the condition, and the `MOST_FREQUENT` function

that returns the most frequent item in the table that results after the column and row selections.
This is remarkable, knowing the probability distribution p only indirectly through a number of samples drawn from p allows us to approximately answer probabilistic inference queries using standard database technology. However, sampling from the distribution poses an interesting conceptual and algorithmic challenge in itself that we address by Gibbs sampling.

Before we discuss Gibbs sampling from a multivariate categorical, we first describe the simpler inverse transform sampling method, which is also used as a subroutine in Gibbs sampling. In contrast to inverse transform sampling, Gibbs sampling can benefit algorithmically from a Markov random field structure.

Inverse transform sampling

As we have discussed before, well established deterministic methods, namely, pseudo random number generators, exist for generating samples from the unit interval $[0, 1]$, that approximate samples drawn independently from the uniform distribution, that is, the probability to observe a sample point in any closed subinterval $[a, b] \subseteq [0, 1]$ is just $b - a$. The idea behind *inverse transform sampling* for multivariate categoricals is to sample a random number r from $[0, 1]$ and to map it to a uniquely defined point

$$t(r) \in \mathcal{X} \;=\; \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$$

such that

$$p\big(t(r)\big) \;=\; \mathsf{P}\big[t^{-1}\big(t(r)\big)\big],$$

where p on the left hand side denotes a multivariate categorical distribution and P on the right hand side denotes the uniform distribution on $[0, 1]$.
An admissible mapping t can, for example, be constructed as follows: Let $x^{(1)}, \ldots, x^{(|\mathcal{X}|)}$ be the elements of $\mathcal{X}$ enumerated in some arbitrary order and define

$$t_k \;=\; \sum_{j=1}^{k} p(x^{(j)}) \quad \text{for} \quad k \in \big[|\mathcal{X}|\big].$$

Then

$$t : [0, 1] \to \mathcal{X},\; r \mapsto x^{(\,\min\{k \in [|\mathcal{X}|] \,:\, r \leq t_k\}\,)}$$

is an admissible mapping that can be used for sampling from a multivariate categorical distribution, because

$$k \;=\; \min\big\{k' \in [|\mathcal{X}|] \;:\; r \leq t_{k'}\big\}$$

implies $r \in (t_{k-1}, t_k]$ and thus

$$\mathsf{P}\big[t^{-1}(x^k)\big] \;=\; \mathsf{P}\big[(t_{k-1}, t_k]\big] \;=\; \mathsf{P}\Big[\Big(\sum_{j=1}^{k-1} p\big(x^{(j)}\big), \sum_{j=1}^{k} p\big(x^{(j)}\big)\Big]\Big]$$
$$= \sum_{j=1}^{k} p\big(x^{(j)}\big) - \sum_{j=1}^{k-1} p\big(x^{(j)}\big) \;=\; p\big(x^{(k)}\big).$$

Note that the size of a data structure for storing the mapping t is proportional to the size of the sample space $\mathcal{X}$. The size of the model p can be much smaller than the size of $\mathcal{X}$, for instance for Markov random fields. Gibbs sampling, in contrast to inverse transform sampling, can benefit from a Markov random field structure.

Gibbs sampling algorithm

Gibbs sampling is general technique for sampling from a multivariate distribution. In the general setting of multivariate categoricals it is less efficient than inverse transform sampling, but becomes more efficient for Markov random fields.
Let p be a multivariate categorical on the sample space $\mathcal{X}$ such that the conditional distributions $p(x_i \mid x_{-i})$ are known and accessible for all $i \in [n]$, that is, we have an algorithm, for instance, the inverse transform sampling algorithm, to draw independent samples from the conditional distributions. This, however, requires us to maintain $|\mathcal{X}_{-i}|$ tables of size $|\mathcal{X}_i|$ for every $i \in [n]$. For Markov random fields with $p(x_i \mid x_{-i}) = p(x_i \mid x_{N(i)})$ the number of tables reduces to $|\mathcal{X}_{N(i)}|$ for $i \in [n]$, which can be much smaller if the numbers of neighbors $N(i)$ are small. Let $N^-(i) = \big\{j \in N(i) \mid j < i\big\}$ and $N^+(i) = \big\{j \in N(i) \mid j > i\big\}$.
The Gibbs sampling algorithm, Algorithm 15, implements a *random walk* on the sample space $\mathcal{X}$. At this point it is not obvious how to use Gibbs sampling for generating independent samples from the probability density function p. Obviously, $x^{(1)}$ is not drawn from p but just an arbitrary point in $\mathcal{X}$. Furthermore, points $x^{(j)}$ and $x^{(j+l)}$, for a small value of l, that are generated shortly after each other are correlated. Thus, we cannot take the points $x^{(j)}$ that are generated by the random walk as independent samples from the multivariate categorical p, even late in the walk. The following theoretical analysis suggests two practical approaches for using the Gibbs sampler to generate approximately independent sample from p: (1) We start the random walk for every sample point that we want to generate, run it for a "large" number k of steps, and only keep the last point of each walk as a sample point. (2) We start the random walk only once, but take only every l-th point of the walk, for a "large" number l, as a sample point. Stop the walk once enough sample points have been generated.

Algorithm 6 Gibbs sampling

input conditional probability tables for the multivariate categorical p,
number of iterations $k \in \mathbb{N}$, and initialization $x^{(1)} = \left(x_1^{(1)}, \ldots, x_n^{(1)}\right) \in \mathcal{X}$

output $x^{(k)}$ is drawn approximately from p

1: **for** $j = 2$ to k **do**
2: **for** $i = 1$ to n **do**
3: set $x_i^{(j)} = x_i$ with probability $p\left(x_i \,|\, x_{N^-(i)}^{(j)}, x_{N^+(i)}^{(j-1)}\right)$
4: **end for**
5: **end for**
6: **return** $x^{(k)}$

Theoretical analysis

For a theoretical analysis of the Gibbs sampling random walk, we consider the *Markov chain* that corresponds to the random walk. The Markov chain acts on the space of multivariate categorical distributions, that is, on the $(|\mathcal{X}| - 1)$-dimensional unit simplex $\Delta_{\mathcal{X}}$, and is characterized by a $|\mathcal{X}| \times |\mathcal{X}|$ *stochastic* transition matrix M, that is, the entries in each row of M add up to 1. The transition matrix is the product of n matrices M_i, $i \in [n]$ that correspond to the random draws of $x_i^{(j)}$, $i \in [n]$ in Line 3 of the Gibbs sampling algorithm. That is,

$$M \;=\; M_1 \cdot \ldots \cdot M_n.$$

For determining the M_i, $i \in [n]$ we decompose every step of the random walk on $\mathcal{X}$ (outer loop) into n sub-steps corresponding to the inner loop in the Gibbs sampling algorithm. At some vector $x \in \mathcal{X}$, the sub-step that corresponds to the matrix M_i changes only the i-th component x_i of x. This change depends only on the current position x of the random walk, or more precisely on the entries of the Markov blanket $x_{N(i)}$. Notably, the change is independent of the value of x_i and also independent of the history of the random walk. The probability for the i-th component to take the value x_i given the components $x_{N(i)}$ is $p(x_i \mid x_{N(i)}) = p(x_i \mid x_{-i})$. Therefore,

$$M_i(x', x) \;=\; \begin{cases} p(x_i \mid x_{N(i)}) : & x'_{-i} = x_{-i} \\ 0 \qquad\quad : & \text{otherwise} \end{cases},$$

where $M_i(x', x)$ is the entry of M_i at the position $(x', x) \in \mathcal{X} \times \mathcal{X}$. The following calculation shows that M_i is indeed a stochastic matrix,

$$\sum_{x \in \mathcal{X}} M_i(x', x) \;=\; \sum_{x \in \mathcal{X}} p(x_i \mid x_{N(i)}) \mathbf{1}[x_{-1} = x'_{-i}] \;=\; \sum_{x_i \in \mathcal{X}_i} p(x_i \mid x'_{N(i)}) \;=\; 1.$$

If we apply the matrix M_i to any $q \in \Delta_{\mathcal{X}}$, then we obtain

$$\begin{aligned}
\big(q^\top M_i\big)(x) \;&=\; \sum_{x' \in \mathcal{X}} M_i(x', x) q(x') \\
&=\; \sum_{x' \in \mathcal{X}} p(x_i \mid x_{N(i)}) \mathbf{1}[x_{-i} = x'_{-i}] q(x') \\
&=\; \sum_{x'_i \in \mathcal{X}_i} p(x_i \mid x_{N(i)}) q(x'_i, x_{-i}) \\
&=\; p(x_i \mid x_{N(i)}) \sum_{x'_i \in \mathcal{X}_i} q(x'_i, x_{-i}) \\
&=\; p(x_i \mid x_{-i}) \sum_{x'_i \in \mathcal{X}_i} q(x'_i, x_{-i}) \;=\; p(x_i \mid x_{-i}) q(x_{-i}).
\end{aligned}$$

Plugging in p for q thus gives

$$\big(p^\top M_i\big)(x_1, \ldots, x_n) \;=\; p(x_i \mid x_{-i}) p(x_{-i}) \;=\; p(x_1, \ldots, x_n)$$

or shorter $p^\top M_i = p^\top$. Since this holds for all $i \in [n]$ we have

$$p^\top M \;=\; p^\top M_1 \cdot \ldots \cdot M_n \;=\; p^\top M_2 \cdot \ldots \cdot M_n \;=\; \ldots \;=\; p^\top M_n \;=\; p^\top$$

and thus p is an *invariant distribution* of the Markov chain. This is remarkable, because if $\bar{p}^\top = \lim_{k \to \infty} q^\top M^k$ exists, then $\bar{p}$ also satisfies

$$\begin{aligned}
\bar{p}^\top M \;&=\; \big(\lim_{k \to \infty} q^\top M^k\big) M \;=\; q^\top \big(\lim_{k \to \infty} M^k\big) M \\
&=\; q^\top \lim_{k \to \infty} M^{k+1} \;=\; q^\top \lim_{k \to \infty} M^k \;=\; \lim_{k \to \infty} q^\top M^k \;=\; \bar{p}^\top.
\end{aligned}$$

Therefore, if the multiplicity of the eigenvalue 1 of the matrix M is one, then $\bar{p} = p$ and thus

$$\lim_{k \to \infty} q^\top M^k \;=\; p^\top.$$

Actually, if the multiplicity of the eigenvalue 1 of the matrix M is one, then $\lim_{k \to \infty} M^k$ is the matrix whose rows are all equal to $\bar{p}^\top = p^\top$.

Coming back to the random walk that is realized by the Gibbs sampler. We have that $x^{(1)}$ is arbitrary. In terms of a probability distribution q on $\mathcal{X}$, the whole probability mass is initially concentrated on $x^{(1)}$, that is,

$$q(x) \;=\; \mathbf{1}\big[x = x^{(1)}\big].$$

The second vector $x^{(2)}$ is the result of a random experiment and the distribution of $x^{(2)}$ is given as

$$x^{(2)} \sim q^\top M,$$

where M is the transition matrix of the corresponding Markov chain. Accordingly we have

$$x^{(k)} \sim q^\top M^{k-1}.$$

If $\lim_{k\to\infty} q^\top M^k$ exists and the multiplicity of the eigenvalue 1 of M is one, then $q^\top M^{k-1} \approx p^\top$ for large k. That is, for large k the vector $x^{(k)}$ is sampled approximately from p.

Chapter 16

Exercises

Chapter 8

Exercise 1. The *Burglary* Bayes net has the binary variables (A)larm, (B)urglary, (C)all, (E)earthquake, and (R)adio. The DAG of the Burglary net is defined by the following parent sets: $pa(A) = \{B, E\}, pa(B) = \emptyset, pa(C) = \{A\}, pa(E) = \emptyset$, and $pa(R) = \{E\}$.

1. Draw the Burglary Bayes net.

2. List all the conditional independencies implied by the DAG.

3. Provide additional independencies not directly implied by the DAG.

Exercise 2. Given a mutlivariate categorical on n variables $x_i, i \in [n]$. For disjoint subsets $A, B, C \subset [n]$ we write $I(A, B, C)$ if the variables with indices in A and the variables with indices in C are independent given the values of the variables in B. The independencies implied by the DAG of a Bayes net can thus be written as $I(x_i, x_{pa(i)}, x_{nd(i)})$.

1. Show that $I(A, B, C)$ if and only if $I(C, B, A)$.

2. Show that $I(A, B, C \cup D)$ implies $I(A, B, C)$ and $I(A, B, D)$.

3. Argue that $I(A, B, C)$ and $I(A, B, D)$ does not imply $I(A, B, C \cup D)$.

Chapter 9

Exercise 1. Given a Bayesian network $pa(x_2) = \{x_1\}$ on two variables $x_1, x_2 \in \mathcal{X} = \{1, 2, 3\}$, and conditional probability tables (CPTs):

x_1	$p(x_1)$
1	0.3
2	0.5
3	0.2

x_1	x_2	$p(x_2\|x_1)$
1	1	0.2
1	2	0.8
2	1	1.0
2	2	0.0
3	1	0.6
3	2	0.4

1. Write down the network polynomial for the Bayesian network.
2. Provide a CNF encoding of the Bayesian network.

Exercise 2. Provide an alternative CNF encoding scheme for Bayesian networks and apply your scheme to the Bayesian network from Exercise 1.

Chapter 11

Exercise 1. Write down the transition matrix of the Gibbs sampler for the Markov random field from the introduction.