The 32nd Finnish Summer School on Probability Theory, 2010

Lectures on Statistical Learning Theory for Chow-Liu Trees

Timo Koski
Institutionen för matematik
Kungliga tekniska högskolan (KTH) , Stockholm

Chapter 1

Introduction

1.1 Product Approximation & Chow-Liu Trees

The topic of storing a high dimensional discrete probability distribution (in a digital medium) appeared very likely for the first time in the journal literature in [35, 50]. If it should not be possible to store the whole distribution, the idea suggested and analysed in loc.cit. was a product approximation of the discrete probability distribution. This special genré of approximation is the point of departure of these lectures.

The problem of storing probability distributions is another expression of the 'curse of dimensionality', and it emerges nowadays, e.g., in data mining, as expounded by H. Mannila et.al. in [34, pp.188−189].

For an intuitive statement of the issues involved we quote P.M. Lewis II in [50, p.220]:

> A product approximation is defined to be an approximation to a higher order distribution made up of a product of several of its lower order component distributions such that the product is an extension of the lower order distributions.

By extension Lewis means that the lower order component distributions can be obtained by marginalization from the product, and that the product is a probability distribution. A product of an arbitrary set of lower dimensional probability distributions will not satisfy these requirements.

The same approach appears in a probabilistic version of *knowledge integration*. The fancy catch-phrase 'knowledge integration' refers, according to Wikipedia[1], to the process of synthesizing multiple knowledge models (or representations) into a common model (representation). Probabilists think that multiple knowledge is represented by a set of probability distributions handling the uncertain information about a partial domain that, e.g., an agent in artificial intelligence possesses. Then knowledge integration is understood as the process

[1] `http://en.wikipedia.org/wiki/Knowledge_integration`

of constructing a joint probability distribution from a set of lower dimensional distributions. This requires a condition of acyclicity [10] on the index sets of the lower dimensional distributions. This kind of condition will also be used in the product approximation and leads to the definition of dependence structures (or conditional independencies). A dependence structure saves space when storing probability tables and saves time when computing and up-dating the probability of an event.

An influential and effective solution to the product approximation problem was some ten years after Lewis' efforts given by C.K. Chow and C.N. Liu [16]. Chow and Liu were associated with Thomas J. Watson Research Center (IBM). The approximation was restricted to a product of second order marginal distributions.

Chow and Liu gave an algorithm for how to select second-order factors for the product approximation so that among all such second-order approximations, the constructed approximation has the minimum Kullback- distance to the actual distribution to be stored. The approximation produces a tree in the sense of graph theory [1, chapter 3]. If every edge in the tree is given a weight corresponding to the mutual information between the variables at its nodes, then the tree which provides the optimal second-order approximation to the target distribution is nothing else but the maximum-weight spanning tree [1, chapter 7.2].

Following the achievement of Chow and Liu there have been a number of extensions of the algorithm, see, e.g., [63, 57] for the extension to *polytrees* and [26] and [72]. It is, however, clear that the complexity of an extended algorithm increases, as the parent set of a variable has more than one member, see [15]. We shall start with a general theory of product approximation that yields the Chow-Liu algorithm and a few other algorithms as special cases.

The goals of any product approximation of a high dimensional probability distribution may be either data compression (source coding) or inference (= computation of probabilities of events given instantiations of other events). Chow-Liu Trees have good properties of probabilistic inference, as the tree is equal to its junction tree, see, e.g., [45, ch.10] for definition of junction tree in the theory of decomposable graphs.

1.2 Statistical Learning Theory

Statistical learning theory applies techniques and ideas of statistics, probability (concentration inequalities), information theory and theoretical computer science to questions of model choice (estimation of structures) and classification and prediction, see, e.g., [14]. Within the framework of these lectures the statistical learning task can be in general terms expressed as follows, c.f. [39, 72].

We have a target class of probabilities $\mathcal{P}(\mathcal{X})$ on a finite but high dimensional discrete set $\mathcal{X}$ and concept class $\mathcal{P}(\mathcal{X};\mathcal{S})$ of probabilities on $\mathcal{X}$ depending on some approximating product structures $\mathcal{S}$. Then one fixes $\mathbf{p} \in \mathcal{P}(\mathcal{X})$ and gets a sample of N independent configurations in $\mathcal{X}$. The learning task is to find a

distribution $\mathbf{p}_{\mathcal{S}}$ in $\mathcal{P}(\mathcal{X};\mathcal{S})$ such that the Kullback distance between the empirical distribution of the N samples and $\mathbf{p}$ is minimized. We are concerned with finding product distributions to represent high dimensional probability distributions. The question is not to find the true structure of $\mathbf{p}$, but to find the best distribution in $\mathcal{P}(\mathcal{X};\mathcal{S})$ that permits tractable inference and source coding.

Chapter 2

Product Approximations of Discrete Probability Distributions

2.1 Multivariate Probability

We introduce a version of an operational calculus for discrete multivariate probability due to [68], c.f. [31], [74], and [88], too.

Let X_i denote a generic random variable defined in a probability space and assuming values $x_i \in \mathcal{X}_i$, an alphabet of discrete symbols with the cardinality $\mid \mathcal{X}_i \mid < \infty$. Symbols may be of any kind, not only numerals. Let $\boldsymbol{X} = (X_i)_{i=1}^d$, i.e., $\boldsymbol{X}$ is d-dimensional ($d < \infty$) block of random variables and thus assumes values in the alphabet $\mathcal{X} = \times_{i=1}^d \mathcal{X}_i$. A *configuration* is $\boldsymbol{x} = (x_i)_{i=1}^d \in \mathcal{X}$. The probability of $\boldsymbol{X}$ in the configuration $\boldsymbol{x}$ is

$$p(\boldsymbol{x}) = P(\boldsymbol{X} = \boldsymbol{x}).$$

Then

$$\mathbf{p} = \{p(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathcal{X}}$$

designates the probability distribution on $\mathcal{X}$ given by the probabilities $p(\boldsymbol{x})$.

The distribution $\mathbf{p}$ is in the first place to be seen, in the terminology of [68, p. 13], as a *tabular distribution*, i.e., $\mathbf{p}$ is a look-up table that gives the probability of each configuration according to $\mathbf{p}$.

Example 2.1.1 Let $\mathcal{X}_1 = \{x_1^1, x_2^1, x_3^1, x_4^1\}$ and $\mathcal{X}_2 = \{x_1^2, x_2^2, x_3^2\}$. A probability table $\mathbf{p}$ in the sense above on $\mathcal{X}_1 \times \mathcal{X}_2$ is given by

$\mathbf{p}$	x_2^1	x_2^2	x_2^3
x_1^1	0.05	0.10	0.05
x_1^2	0.15	0.00	0.25
x_1^3	0.00	0.20	0.05
x_1^4	0.10	0.00	0.05

E.g., if $\boldsymbol{x} = (x_1 x_2) = \left(x_1^2 x_2^3\right)$, then $p(\boldsymbol{x}) = 0.25$. Thus, be quite precise, one should in fact write a configuration as

$$\boldsymbol{x} = \left(x_i^{j_i}\right)_{i=1}^{d} \in \mathcal{X},$$

where

$$x_i^{j_i} \in \mathcal{X}_i = \{x_i^1, \ldots x_i^{|\mathcal{X}_i|}\}.$$

This can be kept in mind, but we do not burden the notation for a configuration by such finer detail.

■

An *algorithmic distribution* consists of an algorithm that, possibly with some numerical information, enables us to compute the probability of a configuration. An example is the forward-backward algorithm for Hidden Markov Chains [62]. Sometimes the algorithm is parametric in the sense that it consists of a formula, i.e., a simple algorithm, like e.g. the binomial probability with two parameters.

The representation and storing of a discrete distribution $\mathbf{p}$ requires then in general exponential size tables, as the list of numbers to be stored is in general of the order $\mid \mathcal{X}_1 - 1 \mid\mid \mathcal{X}_2 - 1 \mid \cdots \mid \mathcal{X}_d - 1 \mid = O\left(2^d\right)$. The intended storing is impossible already for $d > 30$. Therefore the plan is to represent/approximate $\mathbf{p}$ by some dependence structure (that might be) suitable for storing.

Let next $s(\mathcal{X})$ denote a Borel field[1] of subsets of $\mathcal{X}$. A Borel field $s(\mathcal{X})$ has a finite cardinality given in [77]. Then any probability table $\mathbf{p}$ defines a probability measure Pr on $(\mathcal{X}, s(\mathcal{X}))$ by

$$Pr(\mathbf{A}) = \sum_{\boldsymbol{x} \in \mathbf{A}} p(\boldsymbol{x}), \quad \mathbf{A} \in s(\mathcal{X}). \tag{2.1}$$

This is a finitely additive probability measure on $(\mathcal{X}, s(\mathcal{X}))$. Knowledge about dependence structures (or of conditional independences) saves space when storing probability tables and saves time when computing and up-dating the probability $Pr(\mathbf{A})$ of an event $\mathbf{A}$.

Here we have $Pr(\emptyset) = 0$, as $Pr(\mathcal{X}) = 1$ (by convention). Thus we can regard $\mathbf{p}$ as a density of Pr, e.g., with with respect to $\mu(A) =\mid A \mid$, but we shall not make real use (like a change of density) of this property. If Q is another probability measure on $(\mathcal{X}, s(\mathcal{X}))$ with density $\mathbf{q}$, we write

$$Pr \ll Q \quad \text{or} \quad \mathbf{p} \ll \mathbf{q},$$

[1] If $\mathbf{A} \in s(\mathcal{X})$, $\mathbf{B} \in s(\mathcal{X})$, then $\mathbf{A} \cup \mathbf{B} \in s(\mathcal{X})$, $\mathbf{A} \cap \mathbf{B} \in s(\mathcal{X})$, $\mathcal{X} \setminus \mathbf{A} \in s(\mathcal{X})$, $\mathcal{X} \setminus \mathbf{A}$ is the set of configuations outside A, $\setminus$ is the setminus.

if it holds for any $\mathbf{A} \in s(\mathcal{X})$ that

$$Q(\mathbf{A}) = 0 \Rightarrow Pr(\mathbf{A}) = 0. \tag{2.2}$$

We set

$$\mathbf{l} = \{1, 2, \ldots, d\}.$$

For a subset $A \subseteq \mathbf{l}$ we take

$$\boldsymbol{X}_A = (X_i)_{i \in A}$$

and with $\mathcal{X}_A = \times_{i \in A} \mathcal{X}_i$ the configurations

$$\boldsymbol{x}_A = (x_i)_{i \in A} \in \mathcal{X}_A.$$

If $A \subseteq \mathbf{l}$ and $B \subseteq \mathbf{l}$ with $A \cap B = \emptyset$, we understand by *concatenation*

$$\boldsymbol{x}_{A \cup B} = (x_i)_{i \in A \cup B}$$

of $\boldsymbol{x}_A = ((\boldsymbol{x}_A)_i)_{i \in A}$ and $\boldsymbol{x}_B = ((\boldsymbol{x}_B)_i)_{i \in B}$ the following:

$$x_i = \begin{cases} (\boldsymbol{x}_A)_i & \text{if } i \in A \\ (\boldsymbol{x}_B)_i & \text{if } i \in B. \end{cases}$$

This can be written as

$$\boldsymbol{x}_{A \cup B}^{\downarrow A} = \boldsymbol{x}_A, \boldsymbol{x}_{A \cup B}^{\downarrow B} = \boldsymbol{x}_B.$$

In this sense we shall take $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ projections of $\boldsymbol{x}_{A \cup B}$ onto $\mathcal{X}_A$ and $\mathcal{X}_B$ and write

$$\boldsymbol{x}_{A \cup B} = \boldsymbol{x}_A \cdot \boldsymbol{x}_B. \tag{2.3}$$

2.1.1 Marginal Probability

Let A^c designate the set theoretic complement of $A \subseteq \mathbf{l}$. Then we write

$$\boldsymbol{x} = \boldsymbol{x}_A \cdot \boldsymbol{x}_{A^c},$$

where $\boldsymbol{x}_A$ is the projection of $\boldsymbol{x}$ onto $\mathcal{X}_A$ and $\boldsymbol{x}_{A^c}$ is the projection of $\boldsymbol{x}$ onto $\mathcal{X}_{A^c}$. Then we can define the marginal distribution at A as

$$p_A(\boldsymbol{x}_A) = P(\boldsymbol{X}_A = \boldsymbol{x}_A) = \sum_{\boldsymbol{x}_{A^c} \in \mathcal{X}_{A^c}} p(\boldsymbol{x}_A \cdot \boldsymbol{x}_{A^c}). \tag{2.4}$$

In other words, we sum out the variables outside A. In view of the convention in Example 2.1.1 the summation above is a useful short hand for a much more extensive expression.

Fact 1 If $A = \emptyset$, then $p_\emptyset(\boldsymbol{x}_\emptyset) = 1$.

■

For any $A \subseteq \mathbf{l}$ we have the probability table

$$\mathbf{p}_A = \{p_A(\boldsymbol{x}_A)\}_{\boldsymbol{x}_A \in \mathcal{X}_A}.$$

Let us introduce an additional notation that will be helpful in the sequel. We designate the marginalization in (2.4) by

$$\mathbf{p}_A = \mathbf{p}^{\downarrow A}. \tag{2.5}$$

Also, any $\mathbf{p}^{\downarrow A}$ might require a lengthy summation without some simplifying structure that supports tractable computation.

Fact 2 The preceding implies the *vanishing principle*: If $A \subset B \subseteq \mathbf{l}$ and $p_A(\boldsymbol{x}_A) = 0$, then $p_B(\boldsymbol{x}_B) = 0$, since

$$p_B(\boldsymbol{x}_B) = \sum_{\boldsymbol{x}_{B^c} \in \mathcal{X}_{B^c}} p(\boldsymbol{x}_B \cdot \boldsymbol{x}_{B^c}) = \sum_{\boldsymbol{x}_{B^c} \in \mathcal{X}_{B^c}} p\left(\boldsymbol{x}_A \cdot \boldsymbol{x}_{B\setminus A} \cdot \boldsymbol{x}_{B^c}\right)$$

$$\leq \sum_{\boldsymbol{x}_{B\setminus A} \in \mathcal{X}_{B\setminus A}} \sum_{\boldsymbol{x}_{B^c} \in \mathcal{X}_{B^c}} p\left(\boldsymbol{x}_A \cdot \boldsymbol{x}_{B\setminus A} \cdot \boldsymbol{x}_{B^c}\right) = p_A(\boldsymbol{x}_A) = 0.$$

■

The fact and/or the proof above are perhaps more comprehensible, if re-done in a simple example.

Example 2.1.2 Take $\mathcal{X} = \{0,1\}^4$, $\mathbf{l} = \{1,2,3,4\}$. Suppose that $A = \{1,2\}$ and $B = \{1,2,3\}$, so that $B^c = \{4\}$. Suppose that $p_A(00) = 0$. Then

$$p_B(001) = p(0010) + p(0011)$$

$$\leq (p(0010) + p(0000)) + (p(0001) + p(0011))$$

$$= p_{A\cup B^c}(000) + p_{A\cup B^c}(001) = p_A(00) = 0.$$

■

Fact 3 We put to record another property of the entities introduced above, c.f., [68, p.4]. Suppose that $W \subseteq A \subset \mathbf{l}$. We can then think of finding $p_W(\boldsymbol{x}_W)$ either by direct marginalization, or by first finding $p_A(\boldsymbol{x}_A)$ and obtaining the desired margin from this. We take by (2.5)

$$\mathbf{p}_A = \mathbf{p}^{\downarrow A}, \quad \mathbf{p}_W = \mathbf{p}^{\downarrow W}.$$

Then we claim that

$$\mathbf{p}_W = \left(\mathbf{p}^{\downarrow A}\right)^{\downarrow W}. \tag{2.6}$$

Let us start with computation of $\left(\mathbf{p}^{\downarrow A}\right)^{\downarrow W}$ by summing out variables outside W in $\mathbf{p}_A$. This is

$$\sum_{\boldsymbol{x}_{A\cap W^c} \in \mathcal{X}_{A\cap W^c}} p_A(\boldsymbol{x}_W \cdot \boldsymbol{x}_{A\cap W^c})$$

and by (2.4)

$$= \sum_{\boldsymbol{x}_{A\cap W^c}\in\mathcal{X}_{A\cap W^c}} \sum_{\boldsymbol{x}_{A^c}\in\mathcal{X}_{A^c}} p\left(\boldsymbol{x}_W \cdot \boldsymbol{x}_{A\cap W^c} \cdot \boldsymbol{x}_{A^c}\right).$$

But the configuration $\boldsymbol{x}_{A\cap W^c} \cdot \boldsymbol{x}_{A^c}$ is equal to $\boldsymbol{x}_{W^c}$ and thus we get

$$= \sum_{\boldsymbol{x}_{W^c}\in\mathcal{X}_{W^c}} p\left(\boldsymbol{x}_W \cdot \boldsymbol{x}_{A\cap W^c} \cdot \boldsymbol{x}_{A^c}\right) = \sum_{\boldsymbol{x}_{W^c}\in\mathcal{X}_{W^c}} p\left(\boldsymbol{x}_W \cdot \boldsymbol{x}_{W^c}\right)$$

$$= p_W\left(\boldsymbol{x}_W\right),$$

as was desired.

■

2.1.2 Conditional Probability

Let $A \subseteq \mathbf{l}$ and $B \subseteq \mathbf{l}$ with $A \cap B = \emptyset$, and use the concatenation to define

$$p_{A\cup B}\left(\boldsymbol{x}_A, \boldsymbol{x}_B\right) \stackrel{\text{def}}{=} p_{A\cup B}\left(\boldsymbol{x}_{A\cup B}\right) \cdot$$

i.e.,

$$p_{A\cup B}\left(\boldsymbol{x}_A, \boldsymbol{x}_B\right) = \sum_{\boldsymbol{x}_{(A\cup B)^c}\in\mathcal{X}_{(A\cup B)^c}} p\left(\boldsymbol{x}_{A\cup B} \cdot \boldsymbol{x}_{(A\cup B)^c}\right).$$

Thus we can define for $p_B\left(\boldsymbol{x}_B\right) > 0$ the conditional probability

$$p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right) = \frac{p_{A\cup B}\left(\boldsymbol{x}_A, \boldsymbol{x}_B\right)}{p_B\left(\boldsymbol{x}_B\right)}. \tag{2.7}$$

This interpreted as $P\left(\boldsymbol{X}_A = \boldsymbol{x}_A \mid \boldsymbol{X}_B = \boldsymbol{x}_B\right)$. By the vanishing principle in Fact 2 we have that $p_B\left(\boldsymbol{x}_B\right) = 0$ implies $p_{A\cup B}\left(\boldsymbol{x}_A, \boldsymbol{x}_B\right) = 0$. Then $p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right) = \frac{0}{0}$ is undefined. By taking the definition as

$$p_B\left(\boldsymbol{x}_B\right) \cdot p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right) = p_{A\cup B}\left(\boldsymbol{x}_A, \boldsymbol{x}_B\right),$$

we can by convention attach to $p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right)$ an arbitrary value in $[0, 1]$.

For the sake of practicing the calculus, let us check a few basic and elementary consequences of this definition.

Fact 4 We shall establish that

$$\sum_{\boldsymbol{x}_A\in\mathcal{X}_A} p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right) = 1. \tag{2.8}$$

This holds by the following computation.

$$\sum_{\boldsymbol{x}_A\in\mathcal{X}_A} p_{A\cup B}\left(\boldsymbol{x}_A \cdot \boldsymbol{x}_B\right) = \sum_{\boldsymbol{x}_A\in\mathcal{X}_A} \sum_{\boldsymbol{x}_{(A\cup B)^c}\in\mathcal{X}_{(A\cup B)^c}} p\left(\boldsymbol{x}_A \cdot \boldsymbol{x}_B \cdot \boldsymbol{x}_{(A\cup B)^c}\right)$$

and since $\boldsymbol{x} = \boldsymbol{x}_A \cdot \boldsymbol{x}_B \cdot \boldsymbol{x}_{(A\cup B)^c}$ and $\mathcal{X}_A \times \mathcal{X}_{(A\cup B)^c} = \mathcal{X}_{B^c}$, we have

$$= \sum_{\boldsymbol{x}_{B^c} \in \mathcal{X}_{B^c}} p(\boldsymbol{x}) = p_B(\boldsymbol{x}_B),$$

which gives (2.8).

■

Hence we can introduce on $\mathcal{X}_A$ the conditional density or the conditional probability table

$$\mathbf{p}_{A|\boldsymbol{x}_B} = \{p(\boldsymbol{x}_A \mid \boldsymbol{x}_B)\}_{\boldsymbol{x}_A \in \mathcal{X}_A}. \tag{2.9}$$

Fact 5 If $B = \emptyset$, then

$$p(\boldsymbol{x}_A \mid \boldsymbol{x}_\emptyset) = \frac{p_{A\cup\emptyset}(\boldsymbol{x}_{A\cup\emptyset})}{p_\emptyset(\boldsymbol{x}_\emptyset)}. \tag{2.10}$$

In view of fact 1 we have

$$p(\boldsymbol{x}_A \mid \boldsymbol{x}_\emptyset) = p_A(\boldsymbol{x}_A). \tag{2.11}$$

■

2.1.3 Conditional Independence, Independence and the Chain Rule

We consider three disjoint subsets A, B, C of $\mathbf{l}$. The we say that $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$ are *conditionally independent given* $\boldsymbol{X}_C$, if and only if it holds for all configurations $\boldsymbol{x}_A \in \mathcal{X}_A$, $\boldsymbol{x}_B \in \mathcal{X}_B$ and $\boldsymbol{x}_C \in \mathcal{X}_C$ that

$$p_{A\cup B\cup C}(\boldsymbol{x}_A \cdot \boldsymbol{x}_B \cdot \boldsymbol{x}_C) \cdot p_C(\boldsymbol{x}_C) = p_{A\cup C}(\boldsymbol{x}_A \cdot \boldsymbol{x}_C)\, p_{B\cup C}(\boldsymbol{x}_B \cdot \boldsymbol{x}_C). \tag{2.12}$$

We designate the conditional independence as defined above with

$$A \perp B \mid_{\mathbf{p}} C. \tag{2.13}$$

It is immediately clear that $A \perp B \mid_{\mathbf{p}} C$ is equivalent to

$$p(\boldsymbol{x}_A \cdot \boldsymbol{x}_B \mid \boldsymbol{x}_C) = p(\boldsymbol{x}_A \mid \boldsymbol{x}_C)\, p(\boldsymbol{x}_B \mid \boldsymbol{x}_C), \tag{2.14}$$

and to

$$p(\boldsymbol{x}_A \mid \boldsymbol{x}_B \cdot \boldsymbol{x}_C) = p(\boldsymbol{x}_A \mid \boldsymbol{x}_C). \tag{2.15}$$

In view of Fact 5 above we get that

$$A \perp B \mid_{\mathbf{p}} \emptyset \tag{2.16}$$

means that

$$p_{A\cup B}(\boldsymbol{x}_A \cdot \boldsymbol{x}_B) = p_A(\boldsymbol{x}_A)\, p_B(\boldsymbol{x}_B), \tag{2.17}$$

which defines *independence* of $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$. We add to this a more formal definition by a product of tables $\mathbf{p}_A \times \mathbf{p}_B$. We write (2.16) this simply as

$$A \perp_{\mathbf{p}} B. \tag{2.18}$$

By concatenation we can define the product of two functions with different domains, or of two marginal probabilities, $\mathbf{p}_A \times \mathbf{p}_B$, as

$$(p_A \times p_B)(\boldsymbol{x}_{A\cup B}) = p_A(\boldsymbol{x}_A)\, p_B(\boldsymbol{x}_B). \tag{2.19}$$

Suppose now that for $A \subseteq \mathbf{l}$ and $B \subseteq \mathbf{l}$ with $A \cap B = \emptyset$ and for all $\boldsymbol{x} \in \mathcal{X}$

$$p(\boldsymbol{x}) = (p_A \times p_B)(\boldsymbol{x}_{A\cup B}) \cdot p\left(\boldsymbol{x}_{(A\cup B)^c} \mid \boldsymbol{x}_{A\cup B}\right).$$

Then, by fact 4, $\mathbf{p}^{\downarrow A\cup B} = \mathbf{p}_A \times \mathbf{p}_B$ and

$$p(\boldsymbol{x}_A \mid \boldsymbol{x}_B) = p_A(\boldsymbol{x}_A).$$

Next, let us consider

$$\mathbf{p}_i = \mathbf{p}^{\downarrow i}, \quad i = 1, \ldots, d, \tag{2.20}$$

and $\mathbf{p}_i = \{p_i(x_i)\}_{x_i \in \mathcal{X}_i}$. Then we get by the preceding the product of tables $\times_{i=1}^d \mathbf{p}_i$ such that for all $\boldsymbol{x} \in \mathcal{X}$

$$(\times_{i=1}^d p_i)(\boldsymbol{x}) = \prod_{i=1}^d p_i(x_i).$$

Let L be the probability measure on $(\mathcal{X}, s(\mathcal{X}))$ corresponding to $\times_{i=1}^d \mathbf{p}_i$ by (2.1). Then for Pr with density $\mathbf{p}$ it holds in the sense of (2.2) that

$$Pr \ll L \tag{2.21}$$

by the vanishing principle in Fact 2. Sometimes (2.21) is called marginal absolute continuity. This kind of argument can be used, as the probabilities are on a finite space $(\mathcal{X}, s(\mathcal{X}))$, to show that various quantities of information theory used in the sequel will be finite.

Let $(A_i)_{i=1}^k$ be a partition of $\mathbf{l}$ with blocks A_i, i.e.,

$$A_i \cap A_j = \emptyset, j \neq i, \cup_{i=1}^k A_i = \mathbf{l}.$$

By successive applications of (2.7) we have the **chain rule**

$$p(\boldsymbol{x}) = p_{A_1}(\boldsymbol{x}_{A_1}) \prod_{l=2}^k p\left(\boldsymbol{x}_{A_l} \mid \boldsymbol{x}_{A_{l-1}}, \ldots, \boldsymbol{x}_{A_1}\right). \tag{2.22}$$

We assume $p(\boldsymbol{x}) > 0$ in order to avoid discussion of conventions, when there are blocks of zero probability. The expression in the right hand side is the intuitive starting point of the theory in the sequel. We shall be dealing with *dependence structure simplification*, see [59, 60]. This can also be seen as an introduction of a simplifying scheme of conditional independencies.

2.2 Dependence Structures and Product Approximations

2.2.1 Introduction

Intuitively speaking our aim can now be stated as finding a simple approximation of $\mathbf{p}$ by reduction of the number of variables in the conditioning sets in the conditional probability tables in the right hand side of the chain rule (2.22).

As far as probability tables and algorithms are concerned, we shall hereby be advancing an intermediate case. The product approximation is a table which is obtained from other tables involving only a few variables by the algorithm of multiplication.

There are $2^d - 2$ lower dimensional distributions. However, an arbitrary subfamily of lower dimensional distributions will not serve the desired purpose: a set of lower dimensional distributions $\{\mathbf{p}_{W_l}\}_{l=1}^s$, $s < 2^d - 2$, will have to satisfy probabilistic consistency conditions, which in fact require that the schema of sets $\{W_l\}_{l=1}^s$ will satisfy an acyclicity condition found in the theory of relational databases in [10].

We consider first the question of existence of extensions with a more general formulation.

2.2.2 On the Existence of Extensions with Given Margins

We indulge here upon a degree of mathematical generality that will not be sustained in the sequel. Let a) -c) be given, paraphrasing [42, 43]:

a) a family of non-empty subsets $\{W_l\}_{l=1}^s$ of $\mathbf{l}$. Without loss of generality we may assume that $\mathbf{l} = \bigcup_{l=1}^s W_l$.

b) the alphabets $\mathcal{X}_l$ (in the general case these are Hausdorff spaces). By a), $\mathcal{X} = \times_{l=1}^s \mathcal{X}_{W_l}$.

c) a finite Borel measure $\mathbf{p}_{W_l}$ for each W_l, $l = 1, \ldots, s$.

The *classical marginal problem* concerns the existence of a Borel measure (or its density) $\mathbf{p}^*$ on $\mathcal{X}$ such that for all W_l

$$\mathbf{p}_{W_l} = (\mathbf{p}^*)^{\downarrow W_l}. \tag{2.23}$$

Some fundamental contributions to this problem are due to H.G. Kellerer, E. Marczewski, V. Strassen et.al.. We are only interested in the modest special case of finite discrete spaces. In fact, the classical marginal problem has either no solution, one solution or an infinite number of solutions, which feels intuitively plausible.

In [52] the condition in (2.23) is called *collective compatibility*. If the sets W_l are disjoint and the Borel measures are probability measures, the problem has at least the trivial solution (provided the spaces have a countable base):

$$\mathbf{p}^* = \times_{l=1}^s \mathbf{p}_{W_l},$$

or (with the appropriate projections of $\boldsymbol{x}$)

$$p^*(\boldsymbol{x}) = p_{W_1}(\boldsymbol{x}_{W_1}) \cdots p_{W_s}(\boldsymbol{x}_{W_s}).$$

The densities $\mathbf{p}_{W_l}$ should also satisfy an additional consistency condition known as *pairwise compatibility*, see [52]. By pairwise compatibility one means that $C_{i,j} = W_i \bigcap W_j$, $i \neq j$ implies that the margin at $C_{i,j}$ is independent of W_i and W_j, i.e.,

$$\mathbf{p}_{C_{i,j}} = \mathbf{p}_{W_i}^{\downarrow C_{i,j}} = \mathbf{p}_{W_j}^{\downarrow C_{i,j}}. \tag{2.24}$$

In case $C_{i,j} = \emptyset$, we understand (2.24) by Fact 1.

Remark 2.2.1 The classical marginal problem has, as mentioned in the introduction, by a probabilistic interpretation bearing on *knowledge integration*. This is in the sense that pieces of partial knowledge, $\{\mathbf{p}_{W_l}\}_{l=1}^s$, are to be transformed into full knowledge in the form of $\mathbf{p}^*$. If infinitely many solutions exist, one might use some algorithm like maximum entropy or iterative proportional fitting [41, 82] to find a unique optimal solution.

■

The following example (due to [83]) shows that pairwise compatibility does not imply collective compatibility.

Example 2.2.1 Let $\mathbf{l} = \{1, 2, 3\}$ and $W_1 = \{1, 2\}$, $W_2 = \{2, 3\}$ and $W_3 = \{3, 1\}$ and let $\mathcal{X} = \{0, 1\}^3$. Three pairwise joint distributions are specified by

$p_{W_1}(x_1x_2)$	$x_2 = 0$	$x_2 = 1$
$x_1 = 0$	1/2	0
$x_1 = 1$	0	1/2

$p_{W_2}(x_2x_3)$	$x_3 = 0$	$x_3 = 1$
$x_2 = 0$	0	1/2
$x_2 = 1$	1/2	0

$p_{W_3}(x_1x_3)$	$x_3 = 0$	$x_3 = 1$
$x_1 = 0$	1/2	0
$x_1 = 0$	0	1/2

These are pairwise compatible, since $W_1 \bigcap W_2 = C_{12} = \{2\}$, $W_2 \bigcap W_3 = C_{23} = \{3\}$, $W_3 \bigcap W_1 = C_{31} = \{1\}$, and we get

$$\mathbf{p}_{C_{12}} = \mathbf{p}_{C_{23}} = \mathbf{p}_{C_{31}}$$

with

x	0	1
$p_{C_{ij}}(x)$	$\frac{1}{2}$	$\frac{1}{2}$

.

Thus, if a common extension, say $\mathbf{p}^*$, with probabilities $p^*(x_1x_2x_3)$, existed on $\{0,1\}^3$, we would have by marginalization

$$\frac{1}{2} = p_{W_1}(00) = p^*(000) + p^*(001) =$$

$$\leq (p^*(100) + p^*(000)) + (p^*(010) + p^*(001))$$

$$\leq p_{W_2}(00) + p_{W_3}(01) = 0 + 0 = 0.$$

Hence the given probability tables are not collectively compatible.

■

It turns out that the answer to the question, whether pairwise compatibility is sufficient for the existence of the probability $\mathbf{p}^*$ with given marginal distributions $\mathbf{p}_{W_l}$ is of purely combinatorial nature, as the answer is independent of the spaces $\mathcal{X}_{W_l}$ and the probabilities $\mathbf{p}_{W_l}$, see [42, 43].

We shall next describe a situation, where the extension is unique, as we constrain the extension to be of a certain product form in a situation, where pairwise compatibility implies collective compatibility. Again, without loss of generality we take $\mathbf{l} = \bigcup_{l=1}^{s} W_l$.

Definition 2.2.1 There is a total ordering of the sets $W_1, W_2, \ldots, W_s$ such that

$$\text{for every } j > 1 \text{ there is } l < j \text{ s.t. } F_j \equiv W_j \bigcap \left(\bigcup_{r=1}^{j-1} W_r \right) \subseteq W_l. \tag{2.25}$$

The property in (2.25) is known as the *running intersection property*. A set of subsets of $W_1, W_2, \ldots, W_s$ having the running intersection property, given some ordering, is called *acyclic*.

■

The acyclicity condition holds trivially, if $\{W_l\}_{l=1}^{s}$ is partition of $\mathbf{l}$. The acyclity condition was introduced for design of relational databases and analyzed in [10].

There are in fact several interesting links (or ways translating) between relational databases and product representations of probability [13, 87, 88] and Bayesian networks. This means that there exists the possibility of doing probabilistic inference with product approximations using algorithms of databases for which we refer to [44, 53].

Remark 2.2.2 In example 2.2.1 above we have

$$W_3 \cap (W_1 \cup W_2) = \{1, 3\},$$

but $\{1,3\}$ is not a subset of either W_1 or of W_2. Hence the (2.25) does not hold for this ordering. It is easy to check that there is no other ordering that gives (2.25). Hence acyclicity does not hold in the example 2.2.1.

■

It is shown by Malvestuto [52] that acyclicity is equivalent to that pairwise compatibility for all i, j implies collective compatibility. Then Malvestuto shows that the extension in the product form

$$p^*(\boldsymbol{x}) = \frac{\prod_{l=1}^{s} p_{W_l}(\boldsymbol{x}_{W_l})}{\prod_{h=1}^{s-1} p_{V_l}(\boldsymbol{x}_{V_h})} \tag{2.26}$$

exists and is unique. Here each V_l is a subset of two or more W_l's. The uniqueness requires, however, as pointed out in [60], e.g., the assumption that $p_{W_l}(\boldsymbol{x}_{W_l}) > 0$ for all l and all configurations (c.f. the vanishing principle in Fact 2).

The extension in (2.26) is the precise general expression for the product approximation envisaged by Lewis et.al. [50]. We are thus led by the result in (2.26) to introduce our dependence structure simplification.

2.2.3 Dependence Structures

We shall now study the extension of a family of lower dimensional probability distributions by the product representation in (2.26).

As in the discussion preceding the expression in (2.26) we suppose that we have $W_1, W_2, \ldots, W_k$ with $\cup_{l=1}^{s} W_l = \mathbf{l}$ and satisfying the running intersection property (2.25) with this ordering. Then we set $B_1 = \emptyset$ and

$$B_j = W_j \bigcap \left(\bigcup_{k=1}^{j-1} W_k \right), \quad j = 2, \ldots, k$$

and

$$A_j = W_j \setminus B_j \Leftrightarrow W_j = A_j \cup B_j, \quad j = 1, \ldots, k.$$

Now we can readily check that $(A_i)_{i=1}^{k}$ is a partition of $\mathbf{l}$ and that the B_i's satisfy

$$B_j \subset \cup_{i=1}^{j-1} A_i.$$

We use, for reasons of easy reference, $(A_i, B_i)_{i=1}^{k}$ thus obtained as our dependence structure. This hides the running intersection property and acyclicity in the background.

Definition 2.2.2 *[Dependence Structure]* Let $(A_i)_{i=1}^{k}$ be a partition of $\mathbf{l}$, and let $\mathcal{S}$ be a sequence of pairs of subsets of $\mathbf{l}$

$$\mathcal{S} = (A_i, B_i)_{i=1}^{k} \tag{2.27}$$

such that

$$B_1 = \emptyset, B_r \subset \cup_{i=1}^{r-1} A_i \subseteq \mathbf{l}, \quad r = 2, \ldots, k. \tag{2.28}$$

Then $\mathcal{S}$ is a *dependence structure.*

■

Now we can define in an exact manner the product approximation introduced Lewis.

Definition 2.2.3 Let $\mathcal{S}$ be a dependence structure. Then the probability distribution defined by

$$p(\boldsymbol{x} \mid \mathcal{S}) = p_{A_1}(\boldsymbol{x}_{A_1}) \prod_{i=2}^{k} p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}), \boldsymbol{x} \in \mathcal{X}, \tag{2.29}$$

is called a *product approximation* of the distribution $\mathbf{p}$.

■

The product (2.29) is identified as (2.26), if we take $V_\cdot \leftrightarrow B_\cdot$ and $W_\cdot \leftrightarrow A_\cdot \cup B_\cdot$ and $\mathbf{p}^* \leftrightarrow \mathbf{p}_{\mathcal{S}}$.

With a generic $\mathcal{S}$ as in (4.2), we introduce from (2.29)

$$\mathbf{p}_{\mathcal{S}} = \{p(\boldsymbol{x} \mid \mathcal{S})\}_{\boldsymbol{x} \in \mathcal{X}}. \tag{2.30}$$

In order to secure uniqueness of $\mathbf{p}_{\mathcal{S}}$ without arbitrary conventions, we should assume that and $p(\boldsymbol{x}) > 0$ for all configurations, where we recall the vanishing principle in Fact 2.

Due to (2.11) we have for any $B_i = \emptyset$, that $p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}) = p(\boldsymbol{x}_{A_i})$. This will be tacitly used a number of times in the examples below.

Example 2.2.2 Take $k = d$, $A_i = \{i\}$ for $i = 1, 2, \ldots, d$ and $B_i = \emptyset$ for $i = 1, 2, \ldots, d$. This is a dependence structure. Then we can write

$$p(\boldsymbol{x} \mid \mathcal{S}) = \prod_{i=1}^{d} p_i(x_i), \quad \boldsymbol{x} = (x_i)_{i=1}^{d} \in \mathcal{X}. \tag{2.31}$$

This corresponds to approximating (and storing) a probability table $\mathbf{p}$ by a product of its first order marginal distributions or by the independence of the sequence of random variables $X_1, \ldots, X_n$. This is sometimes known as the naíve approximation of $\mathbf{p}$.

■

Example 2.2.3 Take $k = d$, $A_i = \{i\}$ for $i = 1, 2, \ldots, d$ and $B_i = \{i-1\}$, $i = 2, \ldots, d$, $B_1 = \emptyset$. This is a valid dependence structure. Then we can write

$$p(\boldsymbol{x} \mid \mathcal{S}) = p_1(x_1) \prod_{i=2}^{d} p(x_i \mid x_{i-1}), \quad \boldsymbol{x} = (x_i)_{i=1}^{d} \in \mathcal{X}. \tag{2.32}$$

This is, of course, the joint distribution of the sequence $X_1, X_2, \ldots, X_d$, which satisfies the (standard) Markov property.

■

Example 2.2.4 Let r be an integer in $\mathbf{l}$. We take the blocks of the partition as

$$A_1 = \{1, 2, \ldots, r\}, A_i = \{r + i - 1\}, i = 2, \ldots, k = d - r + 1.$$

and define

$$B_1 = \emptyset, B_2 = A_1.$$

$$B_i = [\{r + i - 2\} \cup B_{i-1}] \setminus \{i - 2\}, \quad i = 3, \ldots, k = d - r + 1.$$

This is easier to glance as the product approximation

$$p(\boldsymbol{x} \mid \mathcal{S}) = p_{A_1}(x_1, \ldots, x_r)\, p(x_{r+1} \mid x_r, \ldots, x_1) \cdots p(x_d \mid x_{d-1}, \ldots, x_{d-r}) \tag{2.33}$$

for any $\boldsymbol{x} \in \mathcal{X}$. This is, for natural reasons, called *rth-order Markov property* for $X_1, X_2, \ldots, X_d$, but is, when dealing with binary alphabets [6], known as a *Chow expansion of order r.*

■

Example 2.2.5 Let $\mathcal{G} = (\mathbf{l}, E)$ be a directed and acyclic graph (this will be defined in the sequel). The set of *nodes* is $\mathbf{l} = \{1, \ldots, d\}$ and the *edges* are ordered pairs of $\mathbf{l} \times \mathbf{l}$, i.e.,

$$E = \{(j, k) | j \in \mathbf{l}, k \in \mathbf{l}, j \neq k\}.$$

If $(j, k) \in E$, the node j is said to be a parent of the node k. We can always (see Lemma 2.1 in [68]) enumerate the nodes of a directed and acyclic graph so that the parents of the node i are included in $\{1, 2, \ldots, i - 1\}$. The node 1 has no parents. This ordering of nodes is called *well ordering.* We let $B_i \stackrel{\text{def}}{=}$ parents of i. Then

$$\mathcal{S} = (\{i\}, B_i)_{i=1}^{k} \tag{2.34}$$

is a dependence structure in the sense of Definition 2.2.2. This implies

$$p(\boldsymbol{x} \mid \mathcal{S}) = p_1(x_1) \prod_{i=2}^{d} p\left(x_i \mid \boldsymbol{x}_{B_{i-1}}\right), \boldsymbol{x} \in \mathcal{X}. \tag{2.35}$$

The pair consisting of a directed and acyclic graph and a probability distribution factorized as in (2.35) is known as a *Bayesian network* for the random variables $X_1, \ldots, X_d$, [45, 57].

■

Example 2.2.6 Directed or causal polytrees, to be treated in section 4.4.1 below, are a special case of the preceding example. The algorithm of Chow and Liu, the topic of the next lectures, can be extended to polytrees with tractable computability.

■

2.2.4 Sum-Product Algorithm and Properties of the Product Approximation

We shall now endeavour to verify and/or derive a few basic properties of $\mathbf{p}_{\mathcal{S}}$. The fundamental computational tool in all of this and in much of everything that will follow is the **Sum-Product formula** or algorithm.

Let A and B be disjoint subsets of $\mathbf{l}$ and $\varphi(x_A)$ and $\varphi(x_B)$ be any two (nonnegative) functions (assuming values in a semi-ring) defined on $\mathcal{X}_A$ and $\mathcal{X}_B$, respectively. Then the simplest version of the Sum-Product formula is

$$\sum_{x_A \in \mathcal{X}_A} \varphi(x_A) \cdot \varphi(x_B) = \varphi(x_B) \cdot \sum_{x_A \in \mathcal{X}_A} \varphi(x_A). \tag{2.36}$$

The proof is by the distributive law $xz + yz = z(x + y)$ valid by assumption in all rings. The obvious application of this will be to various marginalizations of $\mathbf{p}_{\mathcal{S}}$, which are of the form

$$\sum \sum \ldots \sum p_{A_1}(\boldsymbol{x}_{A_1}) \prod_{i=2}^{k} p_{A_i}(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}).$$

The work [3] finds generalizations of the Sum-Product formula in new effective algorithms for marginalization, (2.4), and not only for probability distributions. The effectiveness of the algorithms is a consequence of product representations of a function $\beta(\boldsymbol{x})$ that is to be marginalized. It is assumed that $\beta(\boldsymbol{x})$ can be written as a finite product of functions $\alpha_i(\boldsymbol{x}_{S_i})$ with local domains S_i and values in a semi-ring. Clearly, this generalizes the idea of product approximation, where we have $\alpha_i(\boldsymbol{x}_{S_i}) = p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i})$, so that $\boldsymbol{x}_{S_i} = \boldsymbol{x}_{A_i} \cdot \boldsymbol{x}_{B_i}$.

Fact 6 We should check the coherence of the definition 2.2.2, i.e., whether it holds that

$$\sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x} \mid \mathcal{S}) = 1. \tag{2.37}$$

To establish this, we write

$$\sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x} \mid \mathcal{S}) = \sum_{\boldsymbol{x} \in \mathcal{X}} p_{A_1}(\boldsymbol{x}_{A_1}) \prod_{i=2}^{k} p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}).$$

As $(A_i)_{i=1}^{k}$ is a partition of $\mathbf{l}$, then $\mathcal{X} = \times_{i=1}^{k} \mathcal{X}_{A_i}$, and we get

$$= \sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \ldots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p_{A_1}(\boldsymbol{x}_{A_1}) \prod_{i=2}^{k} p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i})$$

and we use the Sum-Product formula (2.36) and $k - (k-1) - \ldots - 1$ as the *elimination order,* to get

$$= \sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} p_{A_1}(\boldsymbol{x}_{A_1}) \sum_{\boldsymbol{x}_{A_2} \in \mathcal{X}_{A_2}} p(\boldsymbol{x}_{A_2} \mid \boldsymbol{x}_{B_2}) \ldots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}_{A_k} \mid \boldsymbol{x}_{B_k}).$$

Recall that the indices in B_i are not involved in summation over $\mathcal{X}_{A_i}$ and therefore we have for every A_i by (2.8) that

$$\sum_{\boldsymbol{x}_{A_i} \in \mathcal{X}_{A_i}} p\left(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}\right) = 1,$$

and we can work from the innermost summation $\sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}}$ to the left, which establishes (2.37).

∎

Fact 7 We should, of course, check collective compatibility, i.e., (2.23), too. Thus we need to show that

$$\mathbf{p}_{\mathcal{S}}^{\downarrow A_r \cup B_r} = \left\{p\left(\boldsymbol{x}_{A_r} \mid \boldsymbol{x}_{B_r}\right) p_{B_r}\left(\boldsymbol{x}_{B_r}\right)\right\}_{\boldsymbol{x}_{A_r} \cdot \boldsymbol{x}_{B_r} \in \mathcal{X}_{A_r \cup B_r}}.$$

This can be written in a less compact notation as

$$p_{A_r \cup B_r}\left(\boldsymbol{x}_{A_r \cup B_r} \mid \mathcal{S}\right) = p\left(\boldsymbol{x}_{A_r} \mid \boldsymbol{x}_{B_r}\right) p_{B_r}\left(\boldsymbol{x}_{B_r}\right), \tag{2.38}$$

where $1 \leq r \leq k$. We start with

$$p\left(\boldsymbol{x} \mid \mathcal{S}\right) = p_{A_1}\left(\boldsymbol{x}_{A_1}\right) \prod_{i=2}^{k} p\left(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}\right), \boldsymbol{x} \in \mathcal{X},$$

where we can sum out $\boldsymbol{x}_{A_k} \ldots \boldsymbol{x}_{A_{r+1}}$ by the same marginalization strategy i.e., the Sum-Product formula (2.36) and the elimination order $k-(k-1)-\ldots-(r+1)$ as in the preceding Fact 6 to obtain

$$p_{\cup_{i=1}^{r} A_i}\left(\boldsymbol{x}_{\cup_{i=1}^{r} A_i} \mid \mathcal{S}\right) = \sum_{\boldsymbol{x}_{A_{r+1}} \in \mathcal{X}_{A_{r+1}}} \ldots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p\left(\boldsymbol{x} \mid \mathcal{S}\right), \tag{2.39}$$

which gives

$$p_{\cup_{i=1}^{r} A_i}\left(\boldsymbol{x}_{\cup_{i=1}^{r} A_i} \mid \mathcal{S}\right) = p\left(\boldsymbol{x}_{A_r} \mid \boldsymbol{x}_{B_r}\right) \left[p_{A_1}\left(\boldsymbol{x}_{A_1}\right) \prod_{i=2}^{r-1} p\left(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}\right)\right]. \tag{2.40}$$

But since $\{A_i\}_{i=1}^{r-1}$ is a partition of $\mathbf{1} \backslash (A_i)_{i=r}^{k}$, the second factor in the expression above is a probability density (this follows as in fact 6 above) on $\times_{i=1}^{r-1} \mathcal{X}_i$, i.e.,

$$p_{\cup_{i=1}^{r-1} A_i}\left(\boldsymbol{x}_{A_1}, \ldots, \boldsymbol{x}_{A_{r-1}}\right) = p_{A_1}\left(\boldsymbol{x}_{A_1}\right) \prod_{i=2}^{r-1} p\left(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}\right).$$

Thus we sum out all the variables outside B_r and get

$$p_{B_r}\left(\boldsymbol{x}_{B_r}\right) = \sum_{\boldsymbol{x}_{B_r^c} \in \left(\mathcal{X}_{\prod_{i=1}^{r-1} A_i}\right) \backslash \mathcal{X}_{B_r}} p_{\cup_{i=1}^{r-1} A_i}\left(\boldsymbol{x}_{A_1}, \ldots, \boldsymbol{x}_{A_{r-1}}\right),$$

and from (2.40)

$$\sum_{\boldsymbol{x}_{B_r^c}\in\left(\mathcal{X}_{\prod_{i=1}^{r-1}A_i}\right)\setminus\mathcal{X}_{B_r}} p_{\cup_{i=1}^{r}A_i}\left(\boldsymbol{x}_{\cup_{i=1}^{r}A_i}\mid\mathcal{S}\right)=p\left(\boldsymbol{x}_{A_r}\mid\boldsymbol{x}_{B_r}\right)p_{B_r}\left(\boldsymbol{x}_{B_r}\right),$$

which is the desired property as claimed in (2.38).

■

Fact 8 (The Causal Markov Property) An important feature of the product $\mathbf{p}_{\mathcal{S}}$ is the set of conditional independences inherent in it. We show next one of these properties, or that we have for any r with $1<r\leq k$

$$A_r\perp\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r\mid_{\mathbf{p}_{\mathcal{S}}}B_r. \tag{2.41}$$

The goal is to check the condition in (2.14) and we begin with

$$p_{A_r\cup\left[\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r\right]\cup B_r}\left(\boldsymbol{x}_{A_r}\cdot\boldsymbol{x}_{\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r}\cdot\boldsymbol{x}_{B_r}\right)=$$

$$=p_{A_r\cup\left(\cup_{i=1}^{r-1}A_i\right)}\left(\boldsymbol{x}_{A_r}\cdot\boldsymbol{x}_{\cup_{i=1}^{r-1}A_i}\right),$$

since $\{A\}_{i=1}^{k}$ is a partition of $\mathbf{l}$ and $B_r\subseteq\cup_{i=1}^{r-1}A_i$. Then we apply (2.40) above to get

$$=p\left(\boldsymbol{x}_{A_r}\mid\boldsymbol{x}_{B_r}\right)\left[p_{A_1}\left(\boldsymbol{x}_{A_1}\right)\prod_{i=2}^{r-1}p\left(\boldsymbol{x}_{A_i}\mid\boldsymbol{x}_{B_i}\right)\right]$$

$$=p\left(\boldsymbol{x}_{A_r}\mid\boldsymbol{x}_{B_r}\right)\cdot p_{\cup_{i=1}^{r-1}A_i}\left(\boldsymbol{x}_{\cup_{i=1}^{r-1}A_i}\right)$$

$$=p\left(\boldsymbol{x}_{A_r}\mid\boldsymbol{x}_{B_r}\right)\cdot p_{\left[\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r\right]\cup B_r}\left(\boldsymbol{x}_{\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r}\cdot\boldsymbol{x}_{B_r}\right)$$

$$=p\left(\boldsymbol{x}_{A_r}\mid\boldsymbol{x}_{B_r}\right)\cdot p\left(\boldsymbol{x}_{\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r}\mid\boldsymbol{x}_{B_r}\right)p_{B_r}\left(\boldsymbol{x}_{B_r}\right).$$

Collecting from the above we have shown that

$$p_{A_r\cup\left[\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r\right]\cup B_r}\left(\boldsymbol{x}_{A_r}\cdot\boldsymbol{x}_{\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r}\cdot\boldsymbol{x}_{B_r}\right)$$

$$=p\left(\boldsymbol{x}_{A_r}\mid\boldsymbol{x}_{B_r}\right)p\left(\boldsymbol{x}_{\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r}\mid\boldsymbol{x}_{B_r}\right)p_{B_r}\left(\boldsymbol{x}_{B_r}\right).$$

By division of both sides in this equality by $p_{B_r}\left(\boldsymbol{x}_{B_r}\right)$ we get

$$p\left(\boldsymbol{x}_{A_r}\cdot\boldsymbol{x}_{\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r}\mid\boldsymbol{x}_{B_r}\right)=p\left(\boldsymbol{x}_{A_r}\mid\boldsymbol{x}_{B_r}\right)p\left(\boldsymbol{x}_{\left(\cup_{i=1}^{r-1}A_i\right)\setminus B_r}\mid\boldsymbol{x}_{B_r}\right). \tag{2.42}$$

By (2.14) this establishes (2.41) as claimed. In view of the examples of dependence structures given above we should recognize (2.41) as a more general sort of Markov property. One reference for (2.41) is *causal Markov property* [71, pp.29−30]. There is a open discussion about the interpretation of the word 'causal' depending on the interpretation of probability, too, [86].

■

2.2.5 What is lost by a Product Approximation ?

A result in [58] shows that one can construct for any distribution a directed acyclic graph by means of the causal Markov condition given a total ordering of the variables.

MORE TO FOLLOW

2.3 Bayesian Update, Approximation of Bayesian Diagnosis and Bayesian Classification

Suppose that $\boldsymbol{x} = \boldsymbol{x}_S \cdot \boldsymbol{x}_D$, where $D = \mathbf{l} \setminus S$. We have *evidence* $\mathbf{e}$ in the form of *instantiations* on $\boldsymbol{X}_S$, which means (see [57, p.152]) that we have observed the values of the variables $\boldsymbol{X}_S$, $\mathbf{e} = \{\boldsymbol{X}_S = \boldsymbol{x}_S\}$. Then we are asked to *update* our probability table $\mathbf{p}^{\downarrow D}$ taking this evidence into account. A standard solution goes as follows.

Let $\mathbf{q}_D = \mathbf{p}^{\downarrow D}$ be the (prior) table on $\mathcal{X}_D$ before the update. Then $\mathbf{d} \in \mathcal{X}_D$ denotes a generic configuration. We apply (2.9) and get the conditional density

$$\mathbf{p}_{S|\mathbf{d}} = \{p(\boldsymbol{x}_S \mid \mathbf{d})\}_{\boldsymbol{x}_S \in \mathcal{X}_S}$$

on $\mathcal{X}_S$. Then the desired up-date of $\mathbf{q}_D$ is, of course, the *posterior probability* $\mathbf{p}_{D|\mathbf{e}}$ given by

$$\mathbf{q}_D \mapsto \mathbf{p}_{D|\mathbf{e}} = \{p(\mathbf{d} \mid \mathbf{e})\}_{\mathbf{d} \in \mathcal{X}_D},$$

where

$$p(\mathbf{d} \mid \mathbf{e}) = \frac{p(\mathbf{e} \mid \mathbf{d})\, q(\mathbf{d})}{p_S(\mathbf{e})}. \tag{2.43}$$

Here we have

$$p_S(\mathbf{e}) = \sum_{\mathbf{d} \in \mathcal{X}_D} p(\mathbf{e} \mid \mathbf{d})\, q(\mathbf{d}) = \sum_{\mathbf{d} \in \mathcal{X}_D} p(\mathbf{e}, \mathbf{d}).$$

Of course, since $\mathbf{e} = \{\boldsymbol{X}_S = \boldsymbol{x}_S\}$, we are inserting the configuration $\boldsymbol{x}_S$ in the formula above. The notation $\mathbf{e}$ points out the distinction of interpretation, i.e., that $\boldsymbol{x}_S$ is an instantiation in the sense of a passively observed sample of $\boldsymbol{X}_S$.

Many of the techniques of supervised and semisupervised learning, or classification, employ (2.43), i.e., Bayes' rule. This requires in many applications the estimation of probability distribution on a high number dimensions. Hence very large ensembles of sample data, c.f. [6], may be needed that in many real applications may be difficult to obtain. Thus a product approximation can be of merit.

Let us think of S as symptom variables, and D as diagnosis variables. We want to find the best estimate of the configuration $\boldsymbol{x}_D$ using the evidence $\mathbf{e}$. Let as above $\mathbf{d} \in \mathcal{X}_D$. Then we consider

$$\mathcal{P}(\mathbf{d}) = \left\{\mathbf{p}_{A|\mathbf{d}} \mid \mathbf{p}_{A|\mathbf{d}} = \mathbf{p}_{S|\mathbf{d}}^{\downarrow A} \quad A \subseteq S\right\}, \tag{2.44}$$

where we applied the notation in (2.9). In other words, $\mathcal{P}(\mathbf{d})$ is the set of lower dimensional marginals of $\mathbf{p}_{S|\mathbf{d}}$.

Hence, we are led to approximate $\mathbf{p}_{S|\mathbf{d}}$ in (2.43) by a product of marginals in $\mathcal{P}(\mathbf{d})$. We denote by $\mathbf{p}_{S|\mathbf{d};\mathcal{S}_D}$ the product satisfying the required conditions of consistency on $\mathcal{X}_S$. Then the approximate Bayesian maximum a posteriori diagnosis or the configuration in $\mathcal{X}_D$ with maximum probability given $\mathbf{e}$ under the product approximation is

$$\widehat{\mathbf{d}}(\mathbf{e}) = \arg\max_{\mathbf{d}\in\mathcal{X}_D} q_S(\mathbf{d}) \cdot p(\mathbf{e} \mid \mathbf{d}; \mathcal{S}_D).$$

For the applications to supervised learning or classification/identification one usually formulates the model a bit differently. Let $\mathcal{X}_D \mapsto C$ and $\mathcal{X}_D \mapsto \mathcal{X}$ in the preceding, and C is an alphabet of class indices c. Then $(\boldsymbol{x}, c) \in \mathcal{X} \times C$. Then we have

$$p(\boldsymbol{x}, c) = p(\boldsymbol{x} \mid c)\, q(c).$$

Let $\mathbf{e} = \{\boldsymbol{X} = \boldsymbol{x}\}$. Then the maximum a posterior estimate of c, or the supervised classification of $\mathbf{e}$, $\widehat{c}(\mathbf{e})$ is

$$\widehat{c}(\mathbf{e}) = \text{argmax}_{c\in C} p(c \mid \mathbf{e}) = \text{argmax}_{c\in C} p(\mathbf{e} \mid c)\, q(c).$$

Then we approximate each $p(\mathbf{e} \mid c)$ with a product approximation $p(\mathbf{e} \mid \mathcal{S}, c)$ and get

$$\widehat{c}(\mathbf{e}; \mathcal{S}) = \text{argmax}_{c\in C} p(\mathbf{e} \mid \mathcal{S}, c)\, q(c).$$

The impact of this product approximation on the probability of error in supervised classification is studied in [28]. In fact, the work by Chow and Liu in [16] seems to be motivated by a study of supervised learning with a product approximation. We shall discuss classification or unsupervised learning aided by the Chow-Liu theory of second order product approximations in more detail in section 7.3 below.

There existed a conjecture that the Chow and Liu algorithm (to be derived below) also minimizes the Bayesian probability of classification error, but this was shown to be a truth with an ample number of modifications, see [8].

2.3.1 Other Variants of Approximation

The final examples are intended to illustrate other strategies of approximating a probability table, where one truncates an additive expansion of a probability table. The example can be omitted without loss of continuity.

Example 2.3.1 Consider the alphabet $\mathcal{X} = \{0, 1\}^d$ known as the binary hypercube in d dimensions. For the binary hypercube we need in general to specify a probability table with $2^d - 1$ entries. Hence we may encounter a difficulty with storing. To recapitulate the *Bahadur-Lazarsfeld -Streitberg representation* of any positive ($p(\boldsymbol{x}) > 0$ for all $\boldsymbol{x}$) probability function on $\{0, 1\}^d$ we define the margins

$$\mathbf{p}_i = \mathbf{p}^{\downarrow i}, \quad i = 1, \ldots, d. \tag{2.45}$$

so that $\mathbf{p}_i = \{p_i(x_i)\}_{x_i \in \{0,1\}}$, where

$$p_i(x_i) = f_i^{x_i}(1-f_i)^{1-x_i}$$

and $0 < f_i < 1$, i.e., $f_i = P(X_i = 1)$. Then we set

$$(\times_{i=1}^{d} p_i)(\boldsymbol{x}) = \prod_{i=1}^{d} p_i(x_i).$$

Let us next define

$$y_i = y_i(\boldsymbol{x}) = \frac{x_i - f_i}{\sqrt{f_i(1-f_i)}}, \quad i = 1, \ldots, d. \tag{2.46}$$

We take $\mathbf{w} = (w_1, w_2, \ldots, w_d) \in \{0,1\}^d$, a binary vector and we denote by $U_{\mathbf{w}}(\boldsymbol{x})$ products of all subsets of $y_1, \ldots, y_d$

$$U_{\mathbf{w}}(\boldsymbol{x}) = \prod_{i=1}^{d} y_i(\boldsymbol{x})^{w_i}, \quad U_{\mathbf{0}}(\boldsymbol{x}) = 1.$$

Then we can find unique the coefficients $\beta_{\mathbf{w}}$ such that

$$f_{\text{interactions}}(\boldsymbol{x}) = \sum_{\mathbf{w} \in \{0,1\}^d} \beta_{\mathbf{w}} U_{\mathbf{w}}(\boldsymbol{x}). \tag{2.47}$$

This is to be checked against the corrections in [73]. We can think of the coefficients $\beta_{\mathbf{w}}$ as interactions of order $R(\mathbf{w})$ minus one, where the rank $R(\mathbf{w})$ of the polynomial $U_{\mathbf{w},c}$ is defined as

$$R(\mathbf{w}) = \sum_{i=1}^{d} w_i.$$

Here $\beta_{\mathbf{0}} = 1$, and if $R(\mathbf{w}) = 1$, then $\beta_{\mathbf{w}} = 0$. For $R(\mathbf{w}) = 2$ the coefficients $\{\beta_{\mathbf{w}}\}$ are correlations. Then the *Bahadur-Lazarsfeld -Streitberg representation* is

$$p(\boldsymbol{x}) = (\times_{i=1}^{d} p_i)(\boldsymbol{x}) f_{\text{interactions}}(\boldsymbol{x}). \tag{2.48}$$

The multivariate Bernoulli distribution $\times_{i=1}^{d} \mathbf{p}_i$ is the first order term. Since $Pr \ll L$ (see (2.21)) we could consider $f_{\text{interactions}}$ as the Radon-Nikodym derivative

$$\frac{dPr}{dL}(\boldsymbol{x}) = f_{\text{interactions}}(\boldsymbol{x}).$$

Clearly one can define an approximation a probability table $\mathbf{p}$ by truncation in $f_{\text{interactions}}(\boldsymbol{x})$ of kth order, i.e., by taking $\beta_{\mathbf{w}} = 0$ for $R(\mathbf{w}) > k$. This has a difficulty, as one does not automatically obtain a valid probability distribution. Hence other ways of approximating $p(\boldsymbol{x})$ on $\{0,1\}^d$ can be easier to use.

■

Example 2.3.2 D. Wedelin gives in [84] a version of product approximation based on representation of probability distributions on B^d in terms of positive potentials that are functions of Hadamard transforms of the original variables.

■

Chapter 3

Reverse I-Projection and the Optimal Product Approximation

3.1 Kullback-Leibler Distance

The *relative entropy* or the *information divergence* or the *I-divergence* or the *Kullback distance* $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}})$ between $\mathbf{p}$ and $\mathbf{p}_{\mathcal{S}}$ in (2.30) is defined by

Definition 3.1.1

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \ln \frac{p(\boldsymbol{x})}{p(\boldsymbol{x} \mid \mathcal{S})}. \tag{3.1}$$

■

By an application of Jensen's inequality we can show, see section A.1.4 in the Appendix, that

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) \geq 0. \tag{3.2}$$

Also, $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) = 0$ if and only if $\mathbf{p} = \mathbf{p}_{\mathcal{S}}$. If $\mathbf{p} \ll \mathbf{p}_{\mathcal{S}}$, in the sense of (2.2) (as probably should be the case), then $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) < \infty$, otherwise $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) = +\infty$. In fact, in finite spaces

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) < \infty \Leftrightarrow \mathbf{p} \ll \mathbf{p}_{\mathcal{S}}.$$

The logarithm is the natural logarithm unless otherwise stated, whereby relative entropy is measured in nats (per sample). The Kullback distance is not really a distance with the properties of a metric on the set of probabilities on $\mathcal{X}$.

Why should the Kullback distance be applied here? Let us keep in mind, as stated in the introduction, that one goal of storing is data compression, i.e., source coding. The Kullback distance is the expected length of redundancy, when coding a configuration $\boldsymbol{x}$, generated by the distribution $\mathbf{p}$, using $\lceil -\log_2 p_{\mathcal{S}}(\boldsymbol{x}) \rceil$ bits, see [20, chapter 5,thm.5.4.3] or [65, ch.3] (and below).

One should from this point of view add to $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}})$ the number of bits needed thereto for storing the probability tables in $\mathbf{p}_{\mathcal{S}}$. The expression $\mathrm{DL}(\mathbf{p}_{\mathcal{S}})$ in (B.2) of Appendix B gives this number. Thus we should be minimizing

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) + \ln 2 \cdot \mathrm{DL}(\mathbf{p}_{\mathcal{S}}).$$

This gives a hint at the minimum description length (MDL) criterion for selection of $\mathbf{p}_{\mathcal{S}}$, see, e.g., [49, 76]. The MDL criterion is, however, usually given in a different form.

The task of *Optimal Product Approximation of* $\mathbf{p}$ is thus to find a dependence structure $\mathcal{S}$ such that $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}})$ is minimized.

Definition 3.1.2 (Optimal Product Approximation) Find an $\mathcal{S}^*$ such that

$$\mathcal{S}^* \in \mathrm{argmin}_{\mathbf{p}_{\mathcal{S}}} D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}). \tag{3.3}$$

■

Following [24], we call $\mathbf{p}_{\mathcal{S}^*}$ a *Reverse I-Projection* of $\mathbf{p}$ onto the set of all probability measures with $\mathcal{S}$ as dependence structure. This is delibrately vague, as we should restrict the domain, where $\mathcal{S}$ is confined to lie, before it makes good sense to talk about the intended minimization. These matters will be made more precise later, but before that our goal will be to rewrite $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}})$ for a generic $\mathcal{S}$ in an instructive form found by Lewis, see [50].

Remark 3.1.1 A measure $\mathbf{q}^*$ in some convex set $\mathcal{E}$ of probability measures with $D(\mathbf{p} \parallel \mathbf{q}) < \infty$ satisfying

$$\mathbf{q}^* = \mathrm{argmin}_{\mathbf{p} \in \mathcal{E}} D(\mathbf{p} \parallel \mathbf{q}). \tag{3.4}$$

is the *I-Projection* of $\mathbf{q}$, see [21]. This defines the method of minimum discrimination information for approximation of distributions by lower order margins in [47]. I-Projections emerge in the Sanov theorem 6.1.1 to be used in the sequel for the large deviation theory of learning of (Chow-Liu) trees.

■

Fact 9 In the derivations to follow we need a computation that will, as a byproduct, define the Shannon entropy, $H(A)$, for $\mathbf{p}_A$. For any $A \subseteq \mathbf{l}$ we observe the following

$$-\sum_{\boldsymbol{x} \in \mathcal{X}} p_A(\boldsymbol{x}) \ln p(\boldsymbol{x}_A) = -\sum_{\boldsymbol{x}_A \in \mathcal{X}_A} p_A(\boldsymbol{x}_A) \ln p_A(\boldsymbol{x}_A) \sum_{\boldsymbol{x}_{A^c} \in \mathcal{X}_{A^c}} p(\boldsymbol{x}_{A^c} \mid \boldsymbol{x}_A) \tag{3.5}$$

and by (2.8)

$$= -\sum_{\boldsymbol{x}_A \in \mathcal{X}_A} p_A(\boldsymbol{x}_A) \ln p_A(\boldsymbol{x}_A).$$

We set

$$H(A) \stackrel{\text{def}}{=} -\sum_{\boldsymbol{x}_A \in \mathcal{X}_A} p_A(\boldsymbol{x}_A) \ln p_A(\boldsymbol{x}_A). \tag{3.6}$$

If $A = \emptyset$, we define, by the fact 1, $H(A) = 0$. Hence, $H(A)$ as given above is the *Shannon entropy* (in natural logarithms or 'nats') of the probability table $\mathbf{p}_A$ (or of the random variable $\boldsymbol{X}_A$).

When we apply the definition in (3.6) with $A = \mathbf{l}$ we get $H(\mathbf{l})$, which is

$$H(\mathbf{l}) = -\sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \ln p(\boldsymbol{x}).$$

Now we can state explicitly the result [20, chapter 5,thm.5.4.3] mentioned above. We recall, e.g., the Huffmann code [1, pp. 94−102] and let

$$E_{\mathbf{p}}\left[\lceil -\log_2 p_{\mathcal{S}}(\boldsymbol{X})\rceil\right] = \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \lceil -\log_2 p_{\mathcal{S}}(\boldsymbol{x})\rceil.$$

Then we clearly obtain the two inequalities

$$H(\mathbf{l}) + D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) \leq E_{\mathbf{p}}\left[\lceil -\log_2 p_{\mathcal{S}}(\boldsymbol{X})\rceil\right] \leq H(\mathbf{l}) + D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) + 1.$$

This shows the Kullback distance (in bits) as the expected source coding length redundancy, within one bit, when coding a configuration $\boldsymbol{x}$, generated by the distribution $\mathbf{p}$, using $\lceil -\log_2 p_{\mathcal{S}}(\boldsymbol{x})\rceil$ bits.

■

Now we can we can write (3.1) as

$$\begin{aligned} D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) &= \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \ln p(\boldsymbol{x}) - \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \ln p(\boldsymbol{x} \mid \mathcal{S}) \\ &= -H(\mathbf{l}) - \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \ln p(\boldsymbol{x} \mid \mathcal{S}). \end{aligned} \tag{3.7}$$

Then we can use the partition $(A_i)_{i=1}^k$ of $\mathbf{l}$ in the definition 2.2.3 to write the second term in the right hand side above as

$$-\sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \ln p(\boldsymbol{x} \mid \mathcal{S}) = -\sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \cdots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}) \ln p(\boldsymbol{x} \mid \mathcal{S}).$$

and use the expression in (2.29) to get

$$= -\sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \cdots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}) \ln \left[p_{A_1}(\boldsymbol{x}_{A_1}) \prod_{k=2}^{k} p(\boldsymbol{x}_{A_k} \mid \boldsymbol{x}_{B_k}) \right]$$

$$= -\sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \cdots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}) \left[\ln p_{A_1}(\boldsymbol{x}_{A_1}) + \sum_{i=2}^{k} \ln p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}) \right].$$

We apply the Sum-Product rule (2.36) rule to obtain

$$= - \sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \cdots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}) \ln p_{A_1}(\boldsymbol{x}_{A_1}) - \sum_{i=2}^{k} \left[\sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \cdots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}) \ln p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}) \right].$$

We use the argument in (3.5) to get

$$- \sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \cdots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}) \ln p_{A_1}(\boldsymbol{x}_{A_1}) = H(A_1) = H(A_1 \cup B_1) - H(B_1), \tag{3.8}$$

since $B_1 = \emptyset$, so that $H(B_1) = H(\emptyset) = 0$. As

$$p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}) = \frac{p_{A_i \cup B_i}(\boldsymbol{x}_{A_i}, \boldsymbol{x}_{B_i})}{p_{B_i}(\boldsymbol{x}_{B_i})}$$

we obtain again with the same steps of computation as in (3.5), in view of the fact that $A_k \cap B_i = \emptyset$ for $i < k$, that

$$\sum_{\boldsymbol{x}_{A_1} \in \mathcal{X}_{A_1}} \cdots \sum_{\boldsymbol{x}_{A_k} \in \mathcal{X}_{A_k}} p(\boldsymbol{x}) \ln p(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i})$$

$$= \sum_{\boldsymbol{x}_{A_i \cup B_i} \in \mathcal{X}_{A_i \cup B_i}} p_{A_i \cup B_i}(\boldsymbol{x}_{A_i \cup B_i}) \ln p_{A_i \cup B_i}(\boldsymbol{x}_{A_i \cup B_i}) - \sum_{\boldsymbol{x}_{B_i} \in \mathcal{X}_{B_i}} p_{B_i}(\boldsymbol{x}_{B_i}) \ln p_{B_i}(\boldsymbol{x}_{B_i}),$$

which is equal to

$$= -H(A_i \cup B_i) + H(B_i). \tag{3.9}$$

Collecting the results from (3.7) − (3.9) and taking into account the appropriate minus signs, we have obtained the following formula valid for any dependence structure $\mathcal{S}$,

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) = -H(\mathbf{l}) + \sum_{i=1}^{k} [H(A_i \cup B_i) - H(B_i)]. \tag{3.10}$$

We shall find a couple of additional representations of $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}})$ by rewriting (3.10) using *mutual (multi)information* to be defined next.

3.2 Mutual Information

Mutual information can be understood as a quantity indicating the degree of dependence between the random variables $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$ for arbitrary $A \subseteq \mathbf{l}$ and $B \subseteq \mathbf{l}$ with $A \cap B = \emptyset$. A more illuminative interpretation of mutual information and the other quantities introduced below is given in the diagram in Appendix A.2.

The formal definitions and formulas to be introduced below are standard identities in information theory, see, e.g., [20, chapter 2], adapted to the present notation.

Definition 3.2.1 The mutual information $\mathbf{I}(A,B)$ between the random variables $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$ with $A \cap B = \emptyset$ is defined as

$$\mathbf{I}(A,B) \stackrel{\text{def}}{=} H(A) + H(B) - H(A \cup B). \tag{3.11}$$

■

We have clearly the symmetry property

$$\mathbf{I}(A,B) = \mathbf{I}(B,A). \tag{3.12}$$

If we want to consider (3.11) in a less compact form, we can readily check that

$$\mathbf{I}(A,B) = \sum_{\boldsymbol{x}_{A\cup B} \in \mathcal{X}_{A\cup B}} p_{A\cup B}(\boldsymbol{x}_{A\cup B}) \ln \frac{p_{A\cup B}(\boldsymbol{x}_{A\cup B})}{p_A(\boldsymbol{x}_A)\, p_B(\boldsymbol{x}_B)}. \tag{3.13}$$

In view of (3.1) and (2.19) this means that

$$\mathbf{I}(A,B) = D(\mathbf{p}_{A\cup B} \parallel \mathbf{p}_A \times \mathbf{p}_B),$$

and therefore due to (3.2), that $\mathbf{I}(A,B) \geq 0$. Furthermore $\mathbf{I}(A,B) = 0$, if and only if $\mathbf{p}_{A\cup B} = \mathbf{p}_A \times \mathbf{p}_B$. Since $\mathbf{p}_{A\cup B} \ll \mathbf{p}_A \times \mathbf{p}_B$ (c.f., the argument yielding (2.21)) we have that $\mathbf{I}(A,B) < \infty$.

Let us also define

$$H(A \mid B) \stackrel{\text{def}}{=} H(A \cup B) - H(B). \tag{3.14}$$

This equals by the preceding

$$H(A \mid B) = \sum_{\boldsymbol{x}_B \in \mathcal{X}_B} p_B(\boldsymbol{x}_B) \sum_{\boldsymbol{x}_A \in \mathcal{X}_A} (-1) \cdot p(\boldsymbol{x}_A \mid \boldsymbol{x}_B) \ln p(\boldsymbol{x}_A \mid \boldsymbol{x}_B).$$

When we apply (3.14) to (3.10), we get

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) = -H(\mathbf{l}) + \sum_{i=1}^{k} [H(A_i \mid B_i)]. \tag{3.15}$$

Since we have by definitions (3.11) and (3.14) that

$$\mathbf{I}(A,B) = H(A) - H(A \mid B), \tag{3.16}$$

we obtain in (3.15) that

$$D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) = -H(\mathbf{l}) + \sum_{i=1}^{k} H(A_i) - \sum_{i=1}^{k} \mathbf{I}(A_i, B_i). \tag{3.17}$$

We observe here that $H(\mathbf{l})$ is independent of the product approximation. Hence we have re-formulated (3.3) as maximization of the objective function

$$Q(\mathcal{S}) = -\sum_{i=1}^{k} H(A_i) + \sum_{i=1}^{k} \mathbf{I}(A_i, B_i) \tag{3.18}$$

to find the Optimal Product Approximation. An important observation is that $Q(\mathcal{S})$ depends on the probability distribution $\mathbf{p}$ only through the lower dimensional marginal distributions at $(A_i, B_i)_{i=1}^k$, as pointed out by P.M. Lewis [50], who first discovered a version of (3.18). This formula has been rediscovered a number of times in the literature on Bayesian networks. K-U. Höffgen [39] and N. Srebro [72] have presented generalizations of the Lewin formula for pac-learning of concepts and for tree-width Markov networks, respectively.

With $Q(\mathcal{S})$ in (3.18) and a set of pairwise consistent lower dimensional marginal distributions with acyclic domains we can maximize $Q(\mathcal{S})$ by evaluation.

Alternatively, we might want to find $\mathcal{S}$ so that $D(\mathbf{p} \parallel \mathbf{p}_{\mathcal{S}}) < \epsilon$ for a given $\epsilon > 0$, see [59], in which case the set of available marginal distributions may be complete.

Remark 3.2.1 For an interpretation (in framework of the theory for digital communication) of the identities (3.14), (3.11) and (3.16) we refer to the diagram A.1 in the Appendix. It holds by the preceding development that we also have

$$Q(\mathcal{S}) = -\sum_{i=1}^{k} H(A_i \mid B_i). \tag{3.19}$$

Hence, the maximization $Q(\mathcal{S})$ can be, by reference to the diagram in Appendix A.2, seen as minimization of the sum of uncertainties of A_i's given B_i's.

■

Remark 3.2.2 Claude E. Shannon [67] observed that, with our notation,

$$d(A, B) = H(A \mid B) + H(B \mid A) \tag{3.20}$$

is a metric between $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$ (or between $\mathbf{p}_A$ and $\mathbf{p}_B$). This fact is used in [66] to find dependence trees with a technique different from that to be treated in the sequel.

■

3.3 Statistical Learning of Dependence Structure

The maximization of $Q(\mathcal{S})$ in (3.18) becomes a technique of *statistical learning of the dependence structure* $\mathcal{S}$ (in some restricted domain of such structures), if we replace the known density $\mathbf{p}$ with an empirical distribution $\widehat{\mathbf{p}}$ for a set of N of observations of $\boldsymbol{X}$ as data.

In the context of Example 2.2.5 this amounts to learning of the structure of a directed acyclic graph from data. In plain words, we want to estimate statistically which node is connected to which node. We refer to [15, 49] for surveys and methods based on mutual information.

For the purpose of statistical estimation of lower dimensional marginals we may want to constrain the general dependence structure in definition 2.2.2 by

$$|A_i \cup B_i| \leq l; 1 \leq l \leq d. \tag{3.21}$$

Here $|A_i \cup B_i|$ is the cardinality of the set.

Even with (3.21), the maximization of (3.18) may turn out to be difficult in a majority of cases.

There is a special scenario that permits an effective solution of the Optimal Product Approximation problem. This is the framework for the algorithm of C.K. Chow and C.N. Liu in [16]. Here one restricts the components of the product approximation to two dimensional distributions, or, requires in (3.21) that $l = 2$.

Chapter 4

The Chow-Liu Theory for the Optimal Product Approximation Problem

In this chapter we shall derive the algorithm of [16] known as the Chow-Liu Algorithm for maximization of $Q(\mathcal{S})$. This is valid for a dependence structure such that the cardinalities are constrained by

$$|A_i \cup B_i| \leq 2. \tag{4.1}$$

In other words we shall give $\mathbf{p}_{\mathcal{S}}$ as a product of second order marginal distributions.

We recapitulate first, for ease of exact reference, some standard definitions and facts of of graph theory, c.f., [1, 85].

4.1 Spanning Trees and Weighted Trees

Consider the undirected graph is $\mathcal{G} = (\mathbf{l}, E)$, i.e., the set of *nodes* is $\mathbf{l} = \{1, \ldots, d\}$ and the *edges* are unordered pairs of $\mathbf{l} \times \mathbf{l}$, i.e.,

$$E = \{(j,k) | j \in \mathbf{l}, k \in \mathbf{l}, j \neq k\}.$$

The graphs considered here are simple in the sense that there are no loops, i..e., edges of the form (j,j), and that there are no multiple edges between two nodes. The nodes j and k are said to be the ends of the edge (j,k).

A graph $\mathcal{G}$ is called *complete*, if there is an edge in E for every possible pair of nodes. The *degree of a node* is is defined as the number of edges incident[1] on it. A *path* of between two nodes is a sequence of distinct edges with an initial and final node. A graph is said to be *connected*, if there is a path between any

[1] the edge (j,k) is incident on j and k

pair of nodes in the graph. A *cycle* is a path, where the initial and final nodes coincide.

A *subgraph* $\mathcal{H}$ of $\mathcal{G}$ is a graph $\mathcal{H} = (\mathbf{l}_H, E_H)$ such that $\mathbf{l}_H \subseteq \mathbf{l}$ and $E_H \subseteq E$. A subgraph $\mathcal{H}$ is *induced* by $A \subseteq \mathbf{l}$, if every edge in E having both its ends in A is also in E_H. A subgraph $\mathcal{H}$ of $\mathcal{G}$ is *spanning subgraph* of $\mathcal{G}$, if $\mathbf{l}_H = \mathbf{l}$.

An (undirected) *tree* $\mathcal{T}$ is a connected undirected graph that has no cycles. It follows that there is a unique path between any two nodes. A *spanning tree of a graph* is a spanning graph of $\mathcal{G}$, which is itself a tree.

A *labelled tree* is a tree on d nodes is a tree, where each node is labelled by one of the integers in $\{1, 2, \ldots, d\}$. In the sequel we are mostly going to refer to labelled trees as trees.

A *weighted* undirected graph is

$$\mathcal{G} = ((\mathbf{l}, E)|\mathbf{w}),$$

where $\mathbf{w}$ is a map from E to the non-negative real numbers defined on the edges $(j, k) \in E$. The weight of a graph (a tree) is the sum of its edge weights. The weight to be used by the Chow-Liu Algorithm will be defined via the mutual information in (3.15):

$$\mathbf{w}(j, k) \stackrel{\text{def}}{=} \mathbf{I}(j, k) = H(j) + H(k) - H(j \cup k).$$

4.2 The Chow-Liu Tree with known p

4.2.1 Chow-Liu Dependence Structure and the Optimal Product

We start with the following special case of dependence structures (2.2.2).

Definition 4.2.1 *[Chow-Liu Dependence Structure]* Let $(i_r)_{r=1}^d$ be an arbitrary permutation of $\mathbf{l} = \{1, 2, \ldots, d\}$. The singleton sets $A_r = \{i_r\}$, $r = 1, \ldots, d$, are a partition of $\mathbf{l}$. Let σ be a sequence of pairs of singletons of $\mathbf{l}$

$$\sigma = (i_r, j_r)_{r=1}^d, \tag{4.2}$$

where

$$j_1 = \emptyset, j_r \in \{i_1, \ldots, i_{r-1}\} \subseteq \mathbf{l}, \quad r = 2, \ldots, d. \tag{4.3}$$

Then σ is a *Chow-Liu dependence structure.*

■

The name tree (or labelled tree) for a Chow-Liu dependence structure is intuitive and correct, since any Chow-Liu dependence structure corresponds to a spanning tree. In fact we may take $(i_r)_{r=1}^d$ as nodes of a graph and join i_d with j_d with an arrowhead pointing from j_r to i_r (thereby introducing direction). We continue with (i_{d-1}, j_{d-1}) and further downwards in the ordering. Since we are in fact dealing with a well ordering, i.e., $j_r \in \{i_1, \ldots, i_{r-1}\}$, we cannot create a cycle.

Hence the procedure will depict a tree. As the tree connects all the nodes, it is a spanning tree. If $j_r = \emptyset$, there is no arrow pointing to the node i_r. Any node in a directed tree with $j_r = \emptyset$ is called a root. By construction (or selection of permutation) i_1 is a root. If there is only one root, the (directed) tree is called proper.

By the causal Markov property (2.41)

$$i_r \perp \{i_1, \ldots, i_{r-1}\} \setminus j_r \mid_{\mathbf{p}_\mathcal{S}} j_r. \tag{4.4}$$

This implies that by conditioning on any j_r we can cut the tree into two conditionally independent trees. Hence a Chow-Liu tree is an example of *Markov trees*.

An example of a Chow-Liu dependence tree for the nodes $\mathbf{l} = \{1, 2, 3, 4, 5, 6, 7\}$ (in this order) is given by

$$(j_1, j_2, j_3, j_4, j_5, j_6, j_7) = (\emptyset, 1, 2, 1, 4, 5, 3).$$

This dependence structure, $(r, j_r)_{r=1}^7$, could be the outcome of step 2. of Algorithm 4 in the Appendix, i.e, before the step of random permutation in the algorithm.

As a tree this is depicted in Figure 4.1. In this figure each node is labelled with the variable associated with it.

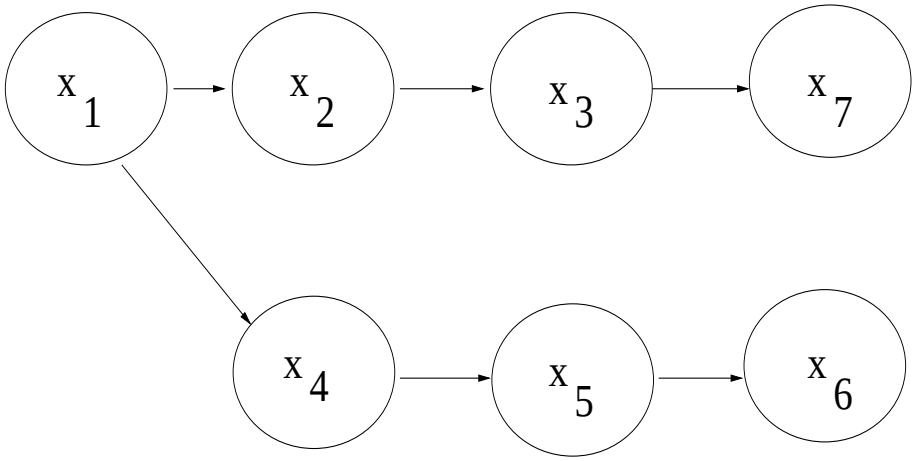

Figure 4.1: A Chow-Liu Dependence Tree

The Chow-Liu algorithm, as defined later, will always output a connected tree. We shall in section 7.2.4 below present a small modification, based on stochastic complexity, that may deliver a disconnected Chow-Liu dependence tree (also known as a forest). An illustration of this for seven nodes is given by

$$(j_1, j_2, j_3, j_4, j_5, j_6, j_7) = (\emptyset, 1, \emptyset, \emptyset, 4, 4, 5)$$

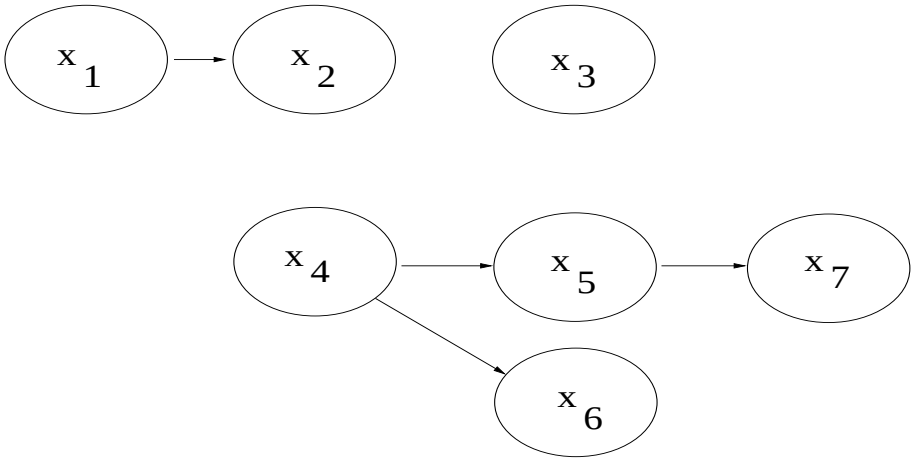

Figure 4.2: A Forest of Chow-Liu Dependence Trees

and the trees are depicted in Figure 4.2 below. For a σ as defined above we have in (3.18)

$$Q(\sigma) = -\sum_{r=1}^{d} H(i_r) + \sum_{r=1}^{d} \mathbf{I}(i_r, j_r).$$

In addition, a Chow-Liu dependence structure defines a product approximation of a known probability distribution $\mathbf{p}$ as

$$\begin{aligned} p(\boldsymbol{x} \mid \sigma) &= p_{i_1}(x_{i_1}) \prod_{r=2}^{d} p(x_{i_r} \mid x_{j_r}) \\ &= \prod_{r=1}^{d} p_{i_r}(x_{i_r}) \prod_{r=2}^{d} \frac{p_{i_r \cup j_r}(x_{i_r}, x_{j_r})}{p_{i_r}(x_{i_r})\, p_{j_r}(x_{j_r})}, \quad \boldsymbol{x} = (x_i)_{i=1}^{d} \in \mathcal{X}. \end{aligned} \tag{4.5}$$

For the trees in the figures 4.1 and 4.2 the joint distributions $p(x_1, \ldots, x_7)$ are

$$p(x_1)\, p(x_2|x_1)\, p(x_3|x_2)\, p(x_4|x_1)\, p(x_5|x_4)\, p(x_6|x_5)\, p(x_7|x_3)$$

and

$$p(x_1)\, p(x_2|x_1)\, p(x_3)\, p(x_4)\, p(x_6|x_4)\, p(x_7|x_5), \tag{4.6}$$

respectively. The following theorem recapitulates the first main result in [16].

Theorem 4.2.1 Let $\mathbf{p}$ be a probability distribution on $\mathcal{X}$. Let

$$\mathcal{G} = ((\mathbf{l}, E)|\mathbf{w}),$$

be a complete weighted graph with $\mathbf{w}$ given by

$$\mathbf{w}(j,k) = \mathbf{I}(j,k), \quad (j,k) \in E, \tag{4.7}$$

where $\mathbf{I}(j,k)$'s are computed using $\mathbf{p}^{\downarrow i,j}$'s respectively. Then the maximum weight spanning tree of $\mathcal{G}$ defines a Chow-Liu dependence structure σ, which maximizes

$$Q(\sigma) = -\sum_{r=1}^{d} H(i_r) + \sum_{r=1}^{d} \mathbf{I}(i_r, j_r) \tag{4.8}$$

Proof: C.f., [31]. Let us note that

$$\sum_{r=1}^{d} H(i_r) = \sum_{i=1}^{d} H(i)$$

and the first term in $Q(\sigma)$ is in fact independent of σ. Hence it suffices to maximize $\sum_{r=1}^{d} \mathbf{I}(i_r, j_r)$.

We show first that any (weighted) spanning tree $\mathcal{T}_d = ((\mathbf{l}, F)|\mathbf{w})$ of $\mathcal{G}$ defines a Chow-Liu dependence structure σ in the sense of definition 4.2.1. Since there are no cycles, there has to be at least one node $i_d \in \mathbf{l}$ with degree one. More precisely, take the node with degree one that has the highest index and renumber. If we delete i_d and the corresponding edge $(i_d, j_d) \in F$, we get a subgraph $\mathcal{T}_{d-1}$ (of $\mathcal{T}_d$)

$$\mathcal{T}_{d-1} = \{(\mathbf{l} \setminus i_d, F \setminus (i_d, j_d))|\mathbf{w}\},$$

which is connected and contains no cycles, i.e., $\mathcal{T}_{d-1}$ is a tree. We can again eliminate a node $i_{d-1} \in \mathbf{l} \setminus i_d$ with degree one in $\mathcal{T}_{d-1}$ and the corresponding edge $(i_{d-1}, j_{d-1}) \in F \setminus (i_d, j_d)$, $j_{d-1} \in \mathbf{l} \setminus (i_d, i_{d-1})$. If we proceed in this manner, we are going to exhaust all $d-1$ edges of the spanning tree $\mathcal{T}_d$ until there is only one isolated node with $j_1 = \emptyset$. The resulting sequence of pairs of nodes

$$\sigma = (i_r, j_r)_{r=1}^{d}$$

satisfies the requirement in definition 4.2.1.

Conversely, we have established above that any dependence structure σ as given in definition 4.2.1 uniquely defines a spanning tree of $\mathcal{G}$.

The weight of $\mathcal{T}_d$ is computed as

$$\mathbf{W}(\mathcal{T}_d) = \sum_{r=2}^{d} \mathbf{I}(i_r, j_r). \tag{4.9}$$

Let next $\mathcal{T}_d^*$ denote a maximum weight spanning tree of $\mathcal{G}$, and let σ^* be its Chow-Liu dependence structure. We want to prove that

$$Q(\sigma^*) \geq Q(\sigma),$$

where σ is the Chow-Liu dependence structure of any other spanning tree $\mathcal{T}$. Let us next suppose that there existed a σ (and therefore a spanning tree of $\mathcal{G}$) such that

$$Q(\sigma) > Q(\sigma^*).$$

Since $\mathcal{T}_d^*$ has maximum weight, we have

$$\mathbf{W}(\mathcal{T}_d^*) = \sum_{r=2}^{d} \mathbf{I}(i_r^*, j_r^*) \geq \mathbf{W}(\mathcal{T}_d) = \sum_{r=2}^{d} \mathbf{I}(i_r, j_r).$$

But in view of (4.8) the assumption $Q(\sigma) > Q(\sigma^*)$ implies that

$$\sum_{r=2}^{d} \mathbf{I}(i_r^*, j_r^*) < \sum_{r=2}^{d} \mathbf{I}(i_r, j_r).$$

■

An algorithm for finding the Chow-Liu dependence structure or the Chow-Liu tree for a known $\mathbf{p}$ is given in section 4.3.4 below.

4.3 The Chow-Liu Algorithm with unknown p: Maximum Likelihood Estimate of Chow-Liu Dependence Structure

In case $\mathbf{p}$ is unknown, we suppose that there is data (or a training set) $\mathbf{D}$, which consists a sample of N independent observations (passively observed instantiations of configurations in $\mathcal{X}$) of $\boldsymbol{X} = (X_i)_{i=1}^d$. We set

$$\mathbf{D} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\},$$

where

$$\boldsymbol{x}^{(n)} = \left(x_i^{(n)}\right)_{i=1}^{d} \in \mathcal{X}.$$

We shall hereby apply *maximum likelihood* estimation for finding the Chow-Liu dependence structure σ. It turns out that the maximum likelihood estimate is nothing else but the procedure obtained in the preceding section, if the probabilities $p_{i\cup j}(x_i, x_j)$, $p_i(x_i)$, $p_j(x_j)$, i.e., the mutual informations, are replaced by their relative frequency, also known as plug-in, estimates. This is the second main result in [16]. But first we present an observation that links maximum likelihood to the reverse I-projection (3.3).

4.3.1 Maximum Likelihood and the Reverse I-projection

First we need certain additional notations. Let $\mathcal{P}(\mathcal{X})$ be the set of all probability distributions (expressed as densities) on $\mathcal{X}$, i.e.,

$$\mathcal{P}(\mathcal{X}) = \left\{\mathbf{p} \mid \mathbf{p} = \{p(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{X}}\right\}. \tag{4.10}$$

Let $\mathcal{T}_d = (\mathbf{l}, \sigma)$ be a spanning tree on $\mathbf{l}$, where σ is a Chow-Liu dependence structure as given in definition 4.2.1. Let us designate by $\mathbf{T}_d$ the set of all spanning trees on $\mathbf{l}$. Then

$$\mathcal{P}(\mathcal{X}, \mathbf{T}_d) = \left\{ \mathbf{p}_\sigma \mid \mathbf{p}_\sigma = \{p(\boldsymbol{x} \mid \sigma)\}_{\boldsymbol{x} \in \mathcal{X}}, \mathcal{T}_d = (\mathbf{l}, \sigma) \in \mathbf{T}_d \right\} \tag{4.11}$$

is the set of all tree dependent probability distributions on $\mathcal{X}$ and $\mathcal{P}(\mathcal{X}, \mathbf{T}_d) \subset \mathcal{P}(\mathcal{X})$.

We set for any $\boldsymbol{x} \in \mathcal{X}$

$$I_{\boldsymbol{x}}\left(\boldsymbol{x}^{(n)}\right) = \begin{cases} 1 & \text{if } \boldsymbol{x} = \boldsymbol{x}^{(n)} \\ 0 & \text{otherwise.} \end{cases}$$

Then we define the empirical probability on $\mathcal{X}$ based on D as

$$\widehat{p}_N(\boldsymbol{x}) = \frac{1}{N} \sum_{n=1}^{N} I_{\boldsymbol{x}}\left(\boldsymbol{x}^{(n)}\right). \tag{4.12}$$

If we let the empirical probability table on $\mathcal{X}$ be given as

$$\widehat{\mathbf{p}}_N = \{\widehat{p}_N(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathcal{X}}, \tag{4.13}$$

then we can find the maximum likelihood estimate $\widehat{\mathbf{p}}_N^{\mathrm{ML}}$ as

$$\widehat{\mathbf{p}}_N^{\mathrm{ML}} = \arg \min_{\mathbf{p} \in \mathcal{P}(\mathcal{X}, \mathbf{T}_d)} D(\widehat{\mathbf{p}}_N \parallel \mathbf{p}). \tag{4.14}$$

To check this, we compute as in (3.7)

$$D(\widehat{p}_N \parallel \mathbf{p}) = -\widehat{H}_N(\mathbf{l}) - \sum_{\boldsymbol{x} \in \mathcal{X}} \widehat{p}_N(\boldsymbol{x}) \ln p(\boldsymbol{x}),$$

where $\widehat{H}_N(\mathbf{l}) = -\sum_{\boldsymbol{x} \in \mathcal{X}} \widehat{p}_N(\boldsymbol{x}) \ln \widehat{p}_N(\boldsymbol{x})$. Now we get from (4.12)

$$\begin{aligned} \sum_{\boldsymbol{x} \in \mathcal{X}} \widehat{p}_N(\boldsymbol{x}) \ln p(\boldsymbol{x}) &= \frac{1}{N} \sum_{n=1}^{N} \sum_{\boldsymbol{x} \in \mathcal{X}} I_{\boldsymbol{x}}\left(\boldsymbol{x}^{(n)}\right) \ln p(\boldsymbol{x}) \\ &= \frac{1}{N} \sum_{n=1}^{N} \ln p\left(\boldsymbol{x}^{(n)}\right), \end{aligned}$$

so that

$$D(\widehat{\mathbf{p}}_N \parallel \mathbf{p}) = -\widehat{H}_N(\mathbf{l}) - \frac{1}{N} \sum_{n=1}^{N} \ln p\left(\boldsymbol{x}^{(n)}\right). \tag{4.15}$$

Obviously, the first term $\widehat{H}_N(\mathbf{l})$ is independent of the tree dependence. Hence we verify the claim in (4.14), i.e, $D(\widehat{\mathbf{p}}_N \parallel \mathbf{p})$ is minimized by maximization of $\frac{1}{N} \sum_{n=1}^{N} \ln p\left(\boldsymbol{x}^{(n)}\right)$, which is the *loglikelihood function*. Hence, in the present situation, the maximum likelihood probability distribution is the *reverse I-projection* of $\widehat{\mathbf{p}}_N$ onto the set of tree dependent distributions $\mathcal{P}(\mathcal{X}, \mathbf{T}_d)$.

We shall next find the maximum likelihood tree-dependent distribution, but the first step is to write $\frac{1}{N} \sum_{n=1}^{N} \ln p\left(\boldsymbol{x}^{(n)}\right)$ as a loglikelihood in a more explicit manner.

4.3.2 The Loglikelihood Function

When $\sigma = (i_r, j_r)_{r=1}^d$ is a Chow-Liu dependence structure as in (4.2) and (4.3) above, we introduce the parameter $\mathcal{P}$ as the set of two dimensional distributions given by

$$\mathcal{P} = \{p_{i\cup j}(x_i, x_j); (i,j) \in \mathbf{l}\times\mathbf{l}, i \neq j\}.$$

Then we write from (4.5) $p(\boldsymbol{x} \mid \sigma, \mathcal{P})$ as the corresponding parametric probability is

$$\begin{aligned} p(\boldsymbol{x} \mid \sigma, \mathcal{P}) &= p_{i_1}(x_{i_1}) \prod_{r=2}^{d} p(x_{i_r} \mid x_{j_r}), \\ &= \prod_{r=1}^{d} p_{i_r}(x_{i_r}) \prod_{r=2}^{d} \frac{p_{i_r\cup j_r}(x_{i_r}, x_{j_r})}{p_{i_r}(x_{i_r})\, p_{j_r}(x_{j_r})} \quad \boldsymbol{x} = (x_i)_{i=1}^d \in \mathcal{X}. \end{aligned} \tag{4.16}$$

The likelihood function for the parameter $(\sigma, \mathcal{P})$ given $\mathbf{D}$ is

$$L(\sigma, \mathcal{P}) = \prod_{n=1}^{N} p\left(\boldsymbol{x}^{(n)} \mid \sigma, \mathcal{P}\right).$$

For any $i \in \mathbf{l}$ and $\xi \in \mathcal{X}_i$ we introduce

$$I_{\xi,i}\left(\boldsymbol{x}^{(n)}\right) = \begin{cases} 1 & \text{if } \xi = x_i^{(n)} \\ 0 & \text{otherwise.} \end{cases}$$

Thus the first line in (4.16) gives

$$L(\sigma, \mathcal{P}) = \prod_{n=1}^{N} \prod_{\xi\in\mathcal{X}_{i_1}} p_{i_1}\left(x_{i_1}^{(n)}\right)^{I_{\xi,i_1}\left(\boldsymbol{x}^{(n)}\right)} \prod_{r=2}^{d} \prod_{\xi\in\mathcal{X}_{i_r}} \prod_{\eta\in\mathcal{X}_{j_r}} \left[p\left(x_{i_r}^{(n)} \mid x_{j_r}^{(n)}\right)\right]^{I_{\xi,i_r}\left(\boldsymbol{x}^{(n)}\right) I_{\eta,j_r}\left(\boldsymbol{x}^{(n)}\right)}$$

$$= \prod_{\xi\in\mathcal{X}_{i_1}} \prod_{n=1}^{N} p_{i_1}\left(x_{i_1}^{(n)}\right)^{I_{\xi,i_1}\left(\boldsymbol{x}^{(n)}\right)} \prod_{r=2}^{d} \prod_{\xi\in\mathcal{X}_{i_r}} \prod_{\eta\in\mathcal{X}_{j_r}} \left[\prod_{n=1}^{N} p\left(x_{i_r}^{(n)} \mid x_{j_r}^{(n)}\right)\right]^{I_{\xi,i_r}\left(\boldsymbol{x}^{(n)}\right) I_{\eta,j_r}\left(\boldsymbol{x}^{(n)}\right)}. \tag{4.17}$$

Then we take the *loglikelihood* function as

$$l(\sigma, \mathcal{P}) = \ln L(\sigma, \mathcal{P}) = \frac{1}{N} \sum_{n=1}^{N} \ln p\left(\boldsymbol{x}^{(n)} \mid \sigma, \mathcal{P}\right).$$

In view of (4.17) this loglikelihood $l(\sigma, \mathcal{P})$ equals

$$= \sum_{\xi\in\mathcal{X}_{i_1}} \frac{1}{N} \sum_{n=1}^{N} I_{\xi,i_1}\left(\boldsymbol{x}^{(n)}\right) \ln p_{i_1}\left(x_{i_1}^{(n)}\right)$$

$$+\sum_{r=2}^{d}\sum_{\xi\in\mathcal{X}_{i_r}}\sum_{\eta\in\mathcal{X}_{j_r}}\frac{1}{N}\sum_{n=1}^{N}I_{\xi,i_r}\left(\boldsymbol{x}^{(n)}\right)I_{\eta,j_r}\left(\boldsymbol{x}^{(n)}\right)\ln p\left(x_{i_r}^{(n)}\mid x_{j_r}^{(n)}\right). \tag{4.18}$$

Let now for any $i\in\mathbf{l}$ and $\xi\in\mathcal{X}_i$

$$\widehat{p}_i\left(\xi\right)=\frac{1}{N}\sum_{n=1}^{N}I_{\xi,i}\left(\boldsymbol{x}^{(n)}\right), \tag{4.19}$$

which is the relative frequency of $\xi\in\mathcal{X}_i$ in our data set $\mathbf{D}$, or, the *plug-in estimate* of p_i. Thereto we have

$$\widehat{p}_{ij}\left(\xi,\eta\right)=\frac{1}{N}\sum_{n=1}^{N}I_{\xi,i}\left(\boldsymbol{x}^{(n)}\right)I_{\eta,j}\left(\boldsymbol{x}^{(n)}\right), \tag{4.20}$$

which is the relative frequency of the pair $(\xi,\eta)\in\mathcal{X}_i\times\mathcal{X}_j$ in $\mathbf{D}$. Here we may observe that in the first term of (4.18)

$$\frac{1}{N}\sum_{n=1}^{N}I_{\xi,i_1}\left(\boldsymbol{x}^{(n)}\right)\ln p_{i_1}\left(x_{i_1}^{(n)}\right)=\frac{1}{N}\sum_{n=1}^{N}I_{\xi,i_1}\left(\boldsymbol{x}^{(n)}\right)\ln p_{i_1}\left(\xi\right)$$

$$=\ln p_{i_1}\left(\xi\right)\frac{1}{N}\sum_{n=1}^{N}I_{\xi,i_1}\left(\boldsymbol{x}^{(n)}\right)=\widehat{p}_{i_1}\left(\xi\right)\ln p_{i_1}\left(\xi\right).$$

Similar straightforward rearrangements yield that

$$l\left(\sigma,\mathcal{P}\right)=\sum_{\xi\in\mathcal{X}_{i_1}}\widehat{p}_{i_1}\left(\xi\right)\ln p_{i_1}\left(\xi\right)+\sum_{r=2}^{d}\sum_{\eta\in\mathcal{X}_{j_r}}\widehat{p}_{j_r}\left(\eta\right)\sum_{\xi\in\mathcal{X}_{i_r}}\frac{\widehat{p}_{i_rj_r}\left(\xi,\eta\right)}{\widehat{p}_{j_r}\left(\eta\right)}\ln p_{i_r|j_r}\left(\xi\mid\eta\right), \tag{4.21}$$

where we have introduced the auxiliary notation $p_{i_r|j_r}\left(\xi\mid\eta\right)$ to designate the conditional probability of $\boldsymbol{X}_{i_r}=\xi$ given $\boldsymbol{X}_{j_r}=\eta$.

4.3.3 Maximum Likelihood

Now we maximize (4.21) as a function of $\sigma,\mathcal{P}$. Let us first fix the structure σ. Gibbs' inequality (A.10) (or the fact that the Kullback distance is nonnegative) yields that the maximum likelihood (ML) estimates are

$$p_{i_1}^{\mathrm{ML}}\left(\xi\right)=\widehat{p}_{i_1}\left(\xi\right),\quad\xi\in\mathcal{X}_{i_1},$$

$$p_{i_r|j_r}^{\mathrm{ML}}\left(\xi\mid\eta\right)=\frac{\widehat{p}_{i_rj_r}\left(\xi,\eta\right)}{\widehat{p}_{j_r}\left(\eta\right)},\quad\xi\in\mathcal{X}_{i_r},\eta\in\mathcal{X}_{j_r},r=2,\ldots,d.$$

If we insert these estimates back in (4.21) we get

$$l\left(\sigma,\mathcal{P}^{\mathrm{ML}}\right)=\sum_{\xi\in\mathcal{X}_{i_1}}\widehat{p}_{i_1}\left(\xi\right)\ln\widehat{p}_{i_1}\left(\xi\right)+\sum_{r=2}^{d}\sum_{\eta\in\mathcal{X}_{j_r}}\sum_{\xi\in\mathcal{X}_{i_r}}\widehat{p}_{i_rj_r}\left(\xi,\eta\right)\ln\frac{\widehat{p}_{i_rj_r}\left(\xi,\eta\right)}{\widehat{p}_{j_r}\left(\eta\right)}$$

$$= \sum_{r=1}^{d} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r}(\xi) \ln \widehat{p}_{i_r}(\xi) + \sum_{r=2}^{d} \sum_{\eta \in \mathcal{X}_{j_r}} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r j_r}(\xi, \eta) \ln \frac{\widehat{p}_{i_r j_r}(\xi, \eta)}{\widehat{p}_{i_r}(\xi) \widehat{p}_{j_r}(\eta)},$$

where we invoked the identities

$$\sum_{r=2}^{d} \sum_{\eta \in \mathcal{X}_{j_r}} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r j_r}(\xi, \eta) \ln \frac{1}{\widehat{p}_{i_r}(\xi)} = \sum_{r=2}^{d} \sum_{\eta \in \mathcal{X}_{j_r}} \sum_{\xi \in \mathcal{X}_{i_r}} (-1) \ln \widehat{p}_{i_r}(\xi) \widehat{p}_{i_r j_r}(\xi, \eta)$$

$$= \sum_{r=2}^{d} \sum_{\xi \in \mathcal{X}_{i_r}} (-1) \ln \widehat{p}_{i_r}(\xi) \sum_{\eta \in \mathcal{X}_{j_r}} \widehat{p}_{i_r j_r}(\xi, \eta) = -\sum_{r=2}^{d} \sum_{\xi \in \mathcal{X}_{i_r}} \ln \widehat{p}_{i_r}(\xi) \widehat{p}_{i_r}(\xi).$$

In other words, we have the *profile likelihood function*

$$l\left(\sigma, \mathcal{P}^{\mathrm{ML}}\right) = \sum_{r=1}^{d} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r}(\xi) \ln \widehat{p}_{i_r}(\xi) + \sum_{r=2}^{d} \widehat{\mathbf{I}}(i_r, j_r), \tag{4.22}$$

where we have introduced the plug-in estimate of mutual information

$$\widehat{\mathbf{I}}(i_r, j_r) = \sum_{\eta \in \mathcal{X}_{j_r}} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r j_r}(\xi, \eta) \ln \frac{\widehat{p}_{i_r j_r}(\xi, \eta)}{\widehat{p}_{i_r}(\xi) \widehat{p}_{j_r}(\eta)}. \tag{4.23}$$

Clearly, the loglikelihood function in (4.22) is the empirical version, or plug-in estimate, of (4.8). The first term $\sum_{r=1}^{d} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r}(\xi) \ln \widehat{p}_{i_r}(\xi)$ does not depend on σ. Hence, we find the maximum likelihood estimate σ^{ML} of the structure σ by

$$\sigma^{\mathrm{ML}} = \mathrm{argmax}_{\sigma} \left\{ \sum_{r=2}^{d} \widehat{\mathbf{I}}(i_r, j_r) \right\}. \tag{4.24}$$

The number of trees with d nodes is finite. Hence in principle we could find σ^{ML} by exhaustive search and evaluation of $l\left(\sigma, \mathcal{P}^{\mathrm{ML}}\right)$. Nevertheless, since the number of spanning trees with d nodes is d^{d-2} [85, Cayley$'$s formula p.82], exhaustive search is infeasible in practice. Hence the second main result of [16] is the observation that there exists a computationally effective way of finding σ^{ML}.

4.3.4 The Algorithm

The Chow-Liu Tree Algorithm is outlined below. The input is the data set

$$\mathbf{D} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$$

There are well known standard algorithms for finding the maximum weight spanning tree, e.g., the *Kruskal algorithm* or the *Prim algorithm* [1], independently discovered by several others, too, c.f., [48, 79]. The algorithm finds the maximum weight spanning tree in $\mathcal{O}\left(d^2 \ln d\right)$ time.

Algorithm 1 CHOW-LIU-TREE (**D**)

1: Determine the the marginal distributions $\widehat{p}_{ij}$ and $\widehat{p}_i$
2: Compute $\mathbf{w} \leftarrow \widehat{\mathbf{I}}(i,j)$
3: **do** MST-KRUSKAL $(\mathcal{G}, \mathbf{w})$
4: **return** σ^{ML}

In other words, in order to use the theorem above for product approximation, we must first compute the $d(d-1)/2$ (c.f., (3.12)) numbers (c.f., 4.23))

$$\widehat{\mathbf{I}}(i,j) = \sum_{x_i \in \mathcal{X}_i} \sum_{x_j \in \mathcal{X}_j} \widehat{p}_{i \cup j}\left(x_i, x_j\right) \ln \frac{\widehat{p}_{i \cup j}\left(x_i, x_j\right)}{\widehat{p}_i\left(x_i\right) \widehat{p}_j\left(x_j\right)}. \tag{4.25}$$

(see (3.10)) for all pairs of $i \in \mathbf{l}, j \in \mathbf{l}$. Then one finds the maximal weight spanning tree of the corresponding complete weighted graph by MST-KRUSKAL in **Algorithm 2** modified from [19, pp.504-509].

Algorithm 2 MST-KRUSKAL $(\mathcal{G}, \mathbf{w})$

1: A $\rightarrow \emptyset$
2: **for** each node $i \in \mathbf{l}$
3: **do** MAKE-SET(i)
4: sort the edges of $\mathbf{l}$ by nonincreasing weight $\mathbf{w}$
5: **for** each edge $(i,j) \in \mathbf{l}$, in order by nonincreasing $\mathbf{w}$
6: **do if** FIND-SET$(i) \neq$ FIND-SET(j)
7: **then** $A \leftarrow A \cup \{(i,j)\}$
8: UNION(i,j)
9: **return** A

FIND-SET(i) returns a representative element from the set that contains i. Lines 1-3 initialize the set A to the empty set and create $|V|$ trees, one containing each node. The edges in $\mathbf{l}$ are sorted into order by nonincreasing weight in line 4. The **for** loop in lines 5-8 checks, for each edge (i,j), whether the endpoints i and j belong to the same tree. If they do, then the edge (i,j) cannot be added to the first forest without creating a cycle, and the edge (i,j) is discarded. Otherwise, the two edges belong to different trees, and the edge (i,j) is added to A in line 7, and the edges of the two trees are merged in line 8.

Remark 4.3.1 Edmonds' algorithm or the *Chu-Liu-Edmonds algorithm* solves a directed maximum spanning tree problem, i.e., finds a maximum/minimum weight spanning tree from directed edges, see [30] for an implementation and for references.

■

4.4 Extensions and Applications of the Chow-Liu Algorithm

4.4.1 Causal Polytrees

Rebane and Pearl [63] introduced the *(causal) polytree*[2], which is a directed graph with at most one undirected path between any two nodes. A polytree has thus no undirected cycles either. Every directed tree is a polytree, but not every polytree is a directed tree. A polytree can be found using edge weights by the MST-KRUSKAL $(\mathcal{G}, \mathbf{w})$ in **Algorithm 2**, as will be shown below.

Let Pol$\mathcal{T}_d$ be a polytree with d nodes. σ is the dependence structure of Pol$\mathcal{T}_d$.

Since a polytree is a an acyclic directed graph, the dependence structure is found in Example 2.2.5 above. Thus $A_i = \{i\}$ and $B_i =$ the set of parents of the node i. We assume that the nodes are well ordered, as defined in Example 2.2.5.

There is, however, an important property not holding for Bayesian networks in general. Let $B_i = \{j_i(1), \ldots, j_i(m)\}$ be the set of parents of i. In a polytree it holds for every $i \in \mathbf{l}$ that

$$p_{B_i}\left(\boldsymbol{x}_{B_i} \mid \sigma\right) = \prod_{l=1}^{m} p_{j_i(l)}\left(x_{j_i(l)}\right). \tag{4.26}$$

We shall demonstrate this. An *ancestor* of the node r in Pol$\mathcal{T}_d$ is a node j $(< r)$ such that there is path (thus unique) from j to r. We set

$$\mathrm{AN}_r = \{ \text{ the ancestors of } r \text{ in Pol}\mathcal{T}_d\}. \tag{4.27}$$

Thus, for any $j_r(k)$ and $j_r(l)$, $l \neq k$, in B_r

$$j_r(k) \notin \mathrm{AN}_{j_r(l)}, \quad j_r(l) \notin \mathrm{AN}_{j_i(k)},$$

because otherwise there would be multiple paths to r. Furthermore

$$\mathrm{AN}_{j_r(k)} \cap \mathrm{AN}_{j_r(l)} = \emptyset,$$

as, if this was not true, there would again exist multiple paths. We have thus in (4.27) the union of disjoint sets

$$\mathrm{AN}_r = \cup_{k=1}^{m} \left[\{j_r(k)\} \cup \mathrm{AN}_{j_r(k)}\right]. \tag{4.28}$$

Then we can eliminate, i.e., sum out with the strategy generically expressed in (2.39), variables so that we get the joint density

$$p_{\mathrm{AN}_r}\left(\boldsymbol{x}_{\mathrm{AN}_r} \mid S\right).$$

[2]Polytree is a well defined notion of graph theory, but the corresponding article `http://en.wikipedia.org/wiki/Polytree` in Wikipedia is what is known as a stub.

By the partition in (4.28) we get

$$p_{\mathrm{AN}_r}\left(\boldsymbol{x}_{\mathrm{AN}_r} \mid S\right) =$$

$$= p_{\cup_{k=1}^{m}\left[\{j_r(k)\}\cup \mathrm{AN}_{j_r(k)}\right]}\left(\left(x_{j_r(1)} \cdot \boldsymbol{x}_{\mathrm{AN}_{j_r(1)}}\right) \cdot \left(x_{j_r(2)} \cdot \boldsymbol{x}_{\mathrm{AN}_{j_r(2)}}\right) \cdots \left(x_{j_r(m)} \cdot \boldsymbol{x}_{\mathrm{AN}_{j_r(m)}}\right) \mid S\right).$$

But, since $p\left(\boldsymbol{x} \mid \sigma\right)$ is factorized as in (2.35) we get the last expression as

$$= \prod_{l=1}^{m} p_{(l)}\left(x_{j_r(l)} \cdot \boldsymbol{x}_{\mathrm{AN}_{j_r(l)}}\right), \tag{4.29}$$

where we have pulled together the factors over the various subpolytrees as

$$p_{(l)}\left(x_{j_r(l)} \cdot \boldsymbol{x}_{\mathrm{AN}_{j_r(l)}}\right) = p\left(x_{j_r(l)} \mid B_{j_r(l)}\right) \cdot \prod_{i \in \mathrm{AN}_{j_r(l)}} p\left(x_i \mid B_i\right).$$

This is possible, as there are no common variables between the various blocks $j_r(l) \cup \mathrm{AN}_{j_r(l)}$. Each and every one of $p_{(l)}\left(x_{j_r(l)} \cdot \boldsymbol{x}_{\mathrm{AN}_{j_r(l)}}\right)$ is joint probability density. Thus we may sum out $\mathrm{AN}_{j_r(l)}$ and get the margin $p_{(l)}\left(x_{j_r(l)}\right)$ and do this separately for each l. Therefore we get from (4.29) the desired product in (4.26).

Example 4.4.1 An polytree is given in Figure 4.3. We shall exemplify the proof of (4.26). In the Figure 4.3 we have

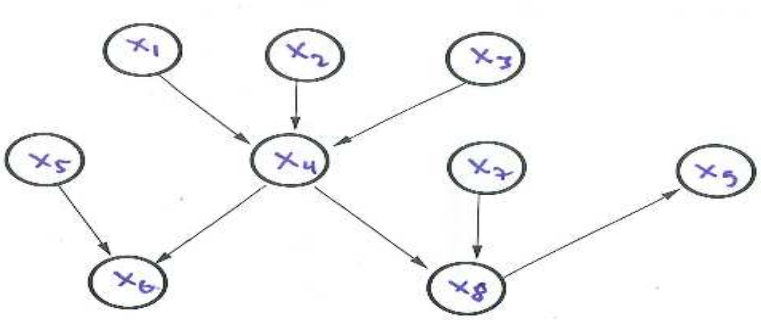

Figure 4.3: A polytree

$p\left(\boldsymbol{x} \mid \sigma\right) = p(x_9 \mid x_8)p(x_8 \mid x_4x_7)p(x_6 \mid x_4x_5)p(x_4 \mid x_1x_2x_3)p(x_1)p(x_2)p(x_3)p(x_5)p(x_7).$

We use the elimination order $9-8-7-6-5$ and the Sum-Product rule to get

$$p_{1\cup 2\cup 3\cup 4}\left(\boldsymbol{x}_{1\cup 2\cup 3\cup 4} \mid \sigma\right) = \sum_{x_5}\sum_{x_6}\sum_{x_7}\sum_{x_8}\sum_{x_9} p\left(\boldsymbol{x} \mid \sigma\right)$$

$$= p(x_4 \mid x_1x_2x_3)p(x_1)p(x_2)p(x_3)$$

When we as the final step sum out x_4 in the last expression, we obtain

$$p_{1\cup 2\cup 3}\left(\boldsymbol{x}_{1\cup 2\cup 3} \mid \sigma\right) = p(x_1)p(x_2)p(x_3). \tag{4.30}$$

This verifies the equality (4.26) with respect to B_4 in Figure 4.3.

Also, if we use the elimination order $9-6-5-8$ we obtain

$$p_{1\cup 2\cup 3\cup 4\cup 7}\left(\boldsymbol{x}_{1\cup 2\cup 3\cup 4\cup 7} \mid \sigma\right) = \sum_{x_8}\sum_{x_5}\sum_{x_6}\sum_{x_9} p\left(\boldsymbol{x}\right)$$

$$= p(x_4 \mid x_1x_2x_3)p(x_1)p(x_2)p(x_3)p(x_7)$$

and, when we use (4.30),

$$= p(x_1x_2x_3x_4 \mid \sigma)p(x_7).$$

Then

$$p_{4\cup 7}\left(\boldsymbol{x}_{4\cup 7} \mid \sigma\right) = \sum_{x_1}\sum_{x_2}\sum_{x_3} p(x_1,x_2,x_3,x_4)p(x_7) = p(x_4)p(x_7),$$

which verifies the claim with respect to B_8 in Figure 4.3.

■

Next we show, see [63], that the Chow-Liu algorithm can be extended to polytrees. This requires a lemma and an auxiliary quantity. For the auxiliary, we define in view of (3.13) a conditional mutual information as

$$\mathbf{I}(A,C|B) = \sum_{\boldsymbol{x}_{A\cup B\cup C}\in\mathcal{X}_{A\cup B\cup C}} p_{A\cup B\cup C}\left(\boldsymbol{x}_{A\cup B\cup C}\right)\ln\frac{p\left(\boldsymbol{x}_A\boldsymbol{x}_C \mid \boldsymbol{x}_B\right)}{p\left(\boldsymbol{x}_A|\boldsymbol{x}_B\right)p\left(\boldsymbol{x}_C|\boldsymbol{x}_B\right)}. \tag{4.31}$$

Lemma 4.4.2 Assume that $A \perp B \mid_{\mathbf{p}} C$. Then it holds for the mutual informations that

$$\min\left(\mathbf{I}(A,C),\mathbf{I}(B,C)\right) \geq \mathbf{I}(A,B). \tag{4.32}$$

Proof: We shall prove the following identities under $A \perp B \mid_{\mathbf{p}} C$.

$$\mathbf{I}(A,B) + \mathbf{I}(A,C|B) = \mathbf{I}(A,C), \tag{4.33}$$

$$\mathbf{I}(A,B) + \mathbf{I}(C,B|A) = \mathbf{I}(B,C). \tag{4.34}$$

We start with (4.33). The ratio in (4.31) can be written as

$$\frac{p\left(\boldsymbol{x}_A\boldsymbol{x}_C \mid \boldsymbol{x}_B\right)}{p\left(\boldsymbol{x}_A|\boldsymbol{x}_B\right)p\left(\boldsymbol{x}_C|\boldsymbol{x}_B\right)} = \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\boldsymbol{x}_C\right)p\left(\boldsymbol{x}_C \mid \boldsymbol{x}_B\right)}{p\left(\boldsymbol{x}_A|\boldsymbol{x}_B\right)p\left(\boldsymbol{x}_C|\boldsymbol{x}_B\right)}$$

$$= \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\boldsymbol{x}_C\right)}{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right)}. \tag{4.35}$$

Now we use the right hand side of (4.35) in the following product:

$$\frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right)}{p_A\left(\boldsymbol{x}_A\right)} \cdot \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\boldsymbol{x}_C\right)}{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right)}$$

$$= \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\boldsymbol{x}_C\right)}{p\left(\boldsymbol{x}_A\right)}. \tag{4.36}$$

Here $A \perp B \mid_{\mathbf{p}} C$ implies (2.15), and thus

$$\frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\boldsymbol{x}_C\right)}{p\left(\boldsymbol{x}_A\right)} = \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_C\right)}{p\left(\boldsymbol{x}_A\right)}. \tag{4.37}$$

Hence we get in the right hand side of (4.36) that

$$= \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_C\right)}{p\left(\boldsymbol{x}_A\right)}. \tag{4.38}$$

When we take logarithms of both sides of (4.35), (4.36) and (4.38) we obtain

$$\ln \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_B\right)}{p_A\left(\boldsymbol{x}_A\right)} + \ln \frac{p\left(\boldsymbol{x}_A\boldsymbol{x}_C \mid \boldsymbol{x}_B\right)}{p\left(\boldsymbol{x}_A|\boldsymbol{x}_B\right)p\left(\boldsymbol{x}_C|\boldsymbol{x}_B\right)} = \ln \frac{p\left(\boldsymbol{x}_A \mid \boldsymbol{x}_C\right)}{p\left(\boldsymbol{x}_A\right)}. \tag{4.39}$$

In view of (3.13) and (4.31) we multiply both sides of (4.39) by $p_{A\cup B\cup C}\left(\boldsymbol{x}_{A\cup B\cup C}\right)$ and sum over $\mathcal{X}_{A\cup B\cup C}$. This yields (4.33). The equality (4.34) is proved analogously. The equalities (4.33) and (4.34) entail (4.32). ∎

Theorem 4.4.3 Let $\mathbf{p}_\sigma$ be a known probability table on $\mathcal{X}$ such that $p_\sigma\left(\boldsymbol{x}\right) > 0$ for all $\boldsymbol{x} \in \mathcal{X}$, where σ is dependence structure of a polytree Pol$\mathcal{T}_d$. Then the algorithm MST-KRUSKAL $(\mathcal{G}, \mathbf{w})$ in **Algorithm 2** will find Pol$\mathcal{T}_d$.

Proof: Let $A = \{i\}$, $B = \{j\}$ and $D = \{k\}$ be three distinct nodes. We shall argue first that the inequality (4.32) is strict.

Now we assume that $i \perp j \mid_{\mathbf{p}_\sigma} k$. This happens by the Markov property (2.41) in the following cases

$$i \to k \to j, \quad i \leftarrow k \leftarrow j, \quad i \leftarrow k \to j \tag{4.40}$$

in Pol$\mathcal{T}_d$.

We are assuming that $p_\sigma(\boldsymbol{x}) > 0$, hence we get in all of the three cases above that $\mathbf{I}(i,k|j) > 0$ and $\mathbf{I}(k,j|i) > 0$. Hence in (4.32)

$$\min(\mathbf{I}(i,k), \mathbf{I}(j,k)) > \mathbf{I}(i,j). \tag{4.41}$$

As stated in the preceding, the MST-KRUSKAL algorithm assigns the two edges of largest weight to the tree, then examines the next edge with the next largest weight, and if it does not create a cycle, it is added to the tree, else it is discarded and the edge with the next largest weight is examined.

Hence the inequality (4.41) implies that the algorithm will not pick the edge (i,j), if there is in σ the node k between i and j.

The final case is $i \rightarrow k \leftarrow j$, of two colliding nodes in σ. In this case i and j are parents of k, and we have seen above in (4.26) that $i \perp_{\mathbf{P}_\sigma} j$, and hence $\mathbf{I}(i,j) = 0$. Thus the algorithm MST-KRUSKAL $(\mathcal{G}, \mathbf{w})$ in **Algorithm 2** will find the true tree skeleton of σ, i.e., a polytree without directions. ∎

The paper by Pearl and Rebane [63] develops an algorithm for directing the edges, too.

Additional extensions of the learning theory for the Chow-Liu trees and polytrees are given in [26] and [72].

4.4.2 Further Extensions and Applications

The Chow-Liu tree is equal to its junction tree, see [45, ch.10], and this enhances inference (or belief propagation) by means of this product approximation. The work in [56] checks and compares the space and time complexity of inference in a Chow-Liu tree with other models for query approximation.

The expressivity of Chow-Liu tree models is expanded by polytrees, and can be vastly enhanced by mixtures of Chow-Liu trees, see [55]. Of course, a mixture of tree dependent distributions is no longer a tree dependent distribution. Marina Meila, [55], has developed an efficient learning method by the EM-algorithm combined with the Chow-Liu algorithm.

An acceleration of the Chow-Liu algorithm is found in [54]. This algorithm takes advantage of the sparsity of the data in computation of the empirical marginals. In [40] one introduces a tree, whose nodes are subsets of variables. Joe Suzuki discusses in [75] an extension of the Chow-Liu trees to variables with continuous alphabets. It is shown in [12] that one can construct spanning trees by a χ^2 statistic, see (A.13) in the Appendix for a hint at the connection.

There is a lot of work on finding interesting patterns with applications to e-commerce, data analysis in molecular biology and text data mining. One example of interesting set of patterns is given by (frequent) itemsets and low entropy itemsets. Applications of Chow-Liu trees to finding itemsets are given in [36, 80, 66].

Chapter 5

Consistency of the Maximum Likelihood Estimate of the Chow-Liu Tree

5.1 The Chow-Liu Maximum Likelihood Tree

We shall now investigate the asymptotic properties of $\sigma^{\mathrm{ML}}(N)$ or, more precisely, of $\mathbf{W}\left(\mathcal{T}_d^{\mathrm{ML}}(N)\right)$, computed by CHOW-LIU-TREE ($\mathbf{D}$) in Algorithm 1, as the number N of independent observations in $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$ [1] increases to $+\infty$.

With the notations in section 4.3.1 above, the maximum likelihood estimate $\widehat{\mathbf{p}}_N^{\mathrm{ML}}$ is recalled as

$$\widehat{\mathbf{p}}_N^{\mathrm{ML}} = \arg \max_{\mathbf{p} \in \mathcal{P}(\mathcal{X}, \mathbf{T}_d)} \frac{1}{N} \sum_{n=1}^{N} \ln p\left(\boldsymbol{x}^{(n)}\right). \tag{5.1}$$

In view of the results in section 4.3.3 we have the following theorem.

Theorem 5.1.1 *[Chow-Liu Tree Learning]*

$$\widehat{\mathbf{p}}_N^{\mathrm{ML}} = \left\{\widehat{p}\left(\boldsymbol{x} \mid \sigma^{\mathrm{ML}}(N)\right)\right\}_{\boldsymbol{x} \in \mathcal{X}} \tag{5.2}$$

with $\sigma^{\mathrm{ML}}(N)$ given in (4.24) and

$$\widehat{p}\left(\boldsymbol{x} \mid \sigma^{\mathrm{ML}}(N)\right) = \prod_{r=1}^{d} \widehat{p}_{i_r}\left(x_{i_r}\right) \prod_{r=2}^{d} \frac{\widehat{p}_{i_r \cup j_r}\left(x_{i_r}, x_{j_r}\right)}{\widehat{p}_{i_r}\left(x_{i_r}\right) \widehat{p}_{j_r}\left(x_{j_r}\right)}, \tag{5.3}$$

[1] In the sequel $\mathbf{D}$ will also be denoted as X^N and $\underline{\boldsymbol{x}}^N$.

and where $\widehat{p}_{i_r}$ and $\widehat{p}_{i_r \cup j_r}$ are the empirical relative probabilities calculated in (4.19) and in (4.20), respectively.

■

5.2 An Aside on the True but Unknown Distribution

For the asymptotic analysis we are going to invoke the 'true' but unknown distribution $\mathbf{p}^{(o)} \in \mathcal{P}(\mathcal{X})$. The meaning of this notion, as passed on by the textbooks[2], is that a set of real data $\mathbf{D}$ is generated like it was sampled from a table of pseudo-random numbers with a suitable transformation to obtain the targeted distribution, and that this distribution represents the information in $\mathbf{D}$ as well as in all future data sets.

Serious criticisms have been levelled against such a concept. To begin with, if we take samples from a distribution in a class of models for some data, we are going to get some or many samples that deviate starkly from the patterns in the real set of data (for which the true but unknown distribution in the model class is supposed to be responsible). In fact it may be difficult to explain, how any finite set of real data can be seen as a set of samples of any distribution, unless we are successful in finding a good predicting distribution prior to any access to data. Furthermore, many of the complex data sets to be modelled today, e.g., in genetics and molecular biology, are the output of several layers of intelligent data processing based on statistical and other methods applied to primary sets of data. This can make it difficult to convince oneself about the physical presence of direct samples from a 'true' but unknown distribution in some known family of distributions, parametric or non-parametric.

One of the vocal critical reviewers of the notion of the true but unknown distribution, Jorma Rissanen, chooses to call the true but unknown distribution 'a metaphysical assumption' [65]. One recommended understanding of statistical inference without the true but unknown distribution, R. Solomonoff's inductive inference, is recapituled in a relatively non-technical manner in [69, 70].

In the preceding chapters, there was no necessity for assuming the true but unknown distribution. This was because we were in the first place approximating a known distribution or finding the reverse I-projection of the empirical distribution on tree dependent distributions.

We may, of course, engage ourselves in the asymptotic analysis merely as a piece of theoretical statistics.

[2]for a very precise statement, see, e.g., pp. 173−177 in H. Cramér: *Sannolikhetskalkylen och några av dess användningar*. Almqvist & Wiksell, Stockholm, 1966.

5.3 A Consistency Property

Suppose now that there is a table $\mathbf{p}^{(o)} \in \mathcal{P}(\mathcal{X})$, which is the 'true' but unknown probability density on $\mathcal{X}$. Then

$$\mathbf{p}_{\sigma^{(o)}} = \text{argmin}_{\mathbf{p}\in\mathcal{P}(\mathcal{X},\mathbf{T}_d)} D\left(\mathbf{p}^{(o)} \parallel \mathbf{p}\right) \tag{5.4}$$

is the tree dependent probability, which is the reverse I-projection of $\mathbf{p}^{(o)}$ and $\sigma^{(o)}$ is a corresponding Chow-Liu dependence structure. Let us observe that $\sigma^{(o)}$ needs not to be unique. If $\mathbf{p}^{(o)}$ happened to be in $\mathcal{P}(\mathcal{X}, \mathbf{T}_d)$, then of course $\mathbf{p}_{\sigma^{(o)}} = \mathbf{p}^{(o)}$.

If $\mathcal{T}_d^{(o)}$ is the tree corresponding to $\sigma^{(o)}$, then, as in (4.9), we have

$$\mathbf{W}\left(\mathcal{T}_d^{(o)}\right) = \sum_{r=2}^{d} \mathbf{I}^{(o)}(i_r^{(o)}, j_r^{(o)}),$$

where $\mathbf{I}^{(o)}(i_r^{(o)}, j_r^{(o)})$ are the mutual informations computed with $(\mathbf{p}_{\sigma^{(o)}})^{\downarrow i_r^{(o)}, j_r^{(o)}}$. This is the maximal tree weight in $\mathcal{P}(\mathcal{X}, \mathbf{T}_d)$ by theorem 4.2.1.

When we compute the reverse I-projection of $\widehat{\mathbf{p}}_N$ onto the set of tree dependent distributions we get, as just shown, the maximum likelihood estimate $\sigma^{\text{ML}}(N)$ of the Chow-Liu dependence structure based on the data $\mathbf{D}$ and the corresponding tree dependent distribution, i.e.,

$$\widehat{\mathbf{p}}_{\sigma^{\text{ML}}(N)} = \text{argmin}_{\mathbf{p}\in\mathcal{P}(\mathcal{X},\mathbf{T}_d)} D\left(\widehat{\mathbf{p}}_N \parallel \mathbf{p}\right), \tag{5.5}$$

where we write

$$\widehat{\mathbf{p}}_{\sigma^{\text{ML}}(N)} = \left\{\widehat{p}\left(\boldsymbol{x} \mid \sigma^{\text{ML}}(N)\right)\right\}_{\boldsymbol{x}\in\mathcal{X}}.$$

Let

$$\mathbf{W}\left(\mathcal{T}_d^{\text{ML}}(N)\right)$$

denote the Chow-Liu dependence tree weight based on $\sigma^{\text{ML}}(N)$ and $\widehat{\mathbf{p}}_{\sigma^{\text{ML}}(N)}$. We shall show that $\mathbf{W}\left(\mathcal{T}_d^{\text{ML}}(N)\right)$ converges almost surely to $\mathbf{W}\left(\mathcal{T}_d^{(o)}\right)$, i.e., to the maximum the tree weight w.r.t. $\mathbf{p}_{\sigma^{(o)}}$. The result is given in [17] without a proof.

Theorem 5.3.1

$$\mathbf{W}\left(\mathcal{T}_d^{\text{ML}}(N)\right) \to \mathbf{W}\left(\mathcal{T}_d^{(o)}\right) \quad \mathbf{p}^{(o)} - \text{a.s.}$$

as $N \to \infty$.

Proof: The strong law of large numbers implies in (4.12) that as $N \to \infty$

$$\mid \widehat{p}_N(\boldsymbol{x}) - p^{(o)}(\boldsymbol{x}) \mid \to 0, \quad \mathbf{p}^{(o)} - \text{a.s.},$$

for all $\boldsymbol{x} \in \mathcal{X}$. Since $\mathcal{X}$ is finite, we have furthermore

$$\max_{\boldsymbol{x}\in\mathcal{X}} \mid \widehat{p}_N(\boldsymbol{x}) - p^{(o)}(\boldsymbol{x}) \mid \to 0, \quad \mathbf{p}^{(o)} - \text{a.s.}.$$

This implies the almost sure convergence of all of the empirical marginal distributions $\widehat{p}_{ij}$ and $\widehat{p}_i(\xi)$ in

$$\widehat{\mathbf{I}}(i_r, j_r) = \sum_{\eta \in \mathcal{X}_{j_r}} \sum_{\xi \in \mathcal{X}_{i_r}} \widehat{p}_{i_r j_r}(\xi, \eta) \ln \frac{\widehat{p}_{i_r j_r}(\xi, \eta)}{\widehat{p}_{i_r}(\xi)\, \widehat{p}_{j_r}(\eta)}. \tag{5.6}$$

For any Chow-Liu dependence structure σ we let $\mathcal{T}_d \in \mathbf{T}_d$ be the corresponding Chow-Liu dependence tree and set

$$\mathbf{W}(\mathcal{T}_d; N) = \sum_{r=2}^{d} \widehat{\mathbf{I}}(i_r, j_r). \tag{5.7}$$

The intention is now to freeze a tree but update $\sum_{r=2}^{d} \widehat{\mathbf{I}}(i_r, j_r)$ for new samples $\boldsymbol{x}^{(N+1)}, \boldsymbol{x}^{(N+2)} \ldots$ by evaluating the plug-in estimates of mutual informations under the given tree dependence and then to run this in parallel for every $\mathcal{T}_d \in \mathbf{T}_d$. Then for every N

$$\mathbf{W}(\mathcal{T}_d; N) \leq \mathbf{W}\left(\mathcal{T}_d^{\mathrm{ML}}(N)\right). \tag{5.8}$$

By simultaneous convergence of each and every one of the empirical marginal distributions $\widehat{p}_{ij}$ and $\widehat{p}_i(\xi)$ we obtain, by continuity of $x \log x$ for $0 < x \leq 1$ with $0 \log 0 = 0$, for every $\mathcal{T}_d \in \mathbf{T}_d$, as $N \to \infty$, that

$$\mathbf{W}(\mathcal{T}_d; N) \to \mathbf{W}(\mathcal{T}_d) \quad \mathbf{p}^{(o)} - \text{a.s.}, \tag{5.9}$$

where

$$\mathbf{W}(\mathcal{T}_d) = \sum_{r=2}^{d} \mathbf{I}^{(o)}(i_r, j_r)$$

is computed using $\mathcal{T}_d$ and $\mathbf{p}^{(o)}$. Since $\mathbf{T}_d$ is a finite set, we have also

$$\max_{\mathcal{T}_d \in \mathbf{T}_d} \mid \mathbf{W}(\mathcal{T}_d; N) - \mathbf{W}(\mathcal{T}_d) \mid \to 0, \quad \mathbf{p}^{(o)} - \text{a.s.}. \tag{5.10}$$

Let now

$$\mathbf{T}_d^{(o)} = \left\{ \mathcal{T}_d \in \mathbf{T}_d \mid \mathbf{W}(\mathcal{T}_d) = \mathbf{W}\left(\mathcal{T}_d^{(o)}\right) \right\}. \tag{5.11}$$

Since $\mathbf{T}_d$ is a finite set, there is by (5.10) a positive δ such that

$$0 < \delta = \min_{\mathcal{T}_d \in \mathbf{T}_d \setminus \mathbf{T}_d^{(o)}} \mid \mathbf{W}\left(\mathcal{T}_d^{(o)}\right) - \mathbf{W}(\mathcal{T}_d) \mid .$$

Let us next choose N large enough, or $N > N_\delta$, so that $\mathbf{p}^{(o)}$ - almost surely

$$\max_{\mathcal{T}_d \in \mathbf{T}_d} \mid \mathbf{W}(\mathcal{T}_d; N) - \mathbf{W}(\mathcal{T}_d) \mid \leq \delta/2.$$

Since (5.10) holds, there is N_δ such that, if $N > N_\delta$, then there is a tree in $\mathbf{T}_d^{(o)}$ in (5.11), say $\mathcal{T}_d^{(o)}$, so that $\mathbf{W}\left(\mathcal{T}_d^{\mathrm{ML}}(N)\right) = \mathbf{W}\left(\mathcal{T}_d^{(o)}; N\right)$ and

$$\mid \mathbf{W}\left(\mathcal{T}_d^{\mathrm{ML}}(N)\right) - \mathbf{W}\left(\mathcal{T}_d^{(o)}\right) \mid \leq \delta/2.$$

In other words, for any $\varepsilon > 0$ with $\delta/2 > \varepsilon$ it holds that for $N > N_\varepsilon$

$$| \mathbf{W}\left(\mathcal{T}_d^{\mathrm{ML}}(N)\right) - \mathbf{W}\left(\mathcal{T}_d^{(o)}\right) | \leq \varepsilon.$$

$\mathbf{p}^{(o)}$ - almost surely. ∎

The result above does not assert convergence of the sequence of Chow-Liu dependence trees $\mathcal{T}_d^{\mathrm{ML}}(N)$. The next chapter will take a more detailed look at the nature of $\mathcal{T}_d^{\mathrm{ML}}(N)$. A learnability result, which is in a sense stronger than the one above, is found [39], who uses the model of PAC-learning and derives the sample complexity of learning a Chow-Liu tree.

Chapter 6

A Large Deviation Analysis of Learning of Chow-Liu Trees

6.1 An Outline of Large Deviations and the Sanov Theorem for Finite Alphabets

We shall next investigate the speed of convergence to zero of the probability of error in maximum likelihood estimation of the Chow-Liu dependence structure. This involves the technique of error exponents widely used in coding theorems [22] and is based on large deviation techniques.

Large deviation theory, see, e.g., [38], is concerned with events that have small probability and decrease to zero, as a function of some variable. Typically, the convergence to zero is exponential. We shall now briefly outline the large deviation result known as the Sanov Theorem in a finite space.

Let $\underline{\boldsymbol{x}}^N \stackrel{def}{=} \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}$ be a sequence of configurations $\boldsymbol{x}^{(n)}$ in $\mathcal{X} = \{a_1, \ldots, a_{|\mathcal{X}|}\}$, an auxiliary notation introduced for the present needs. We regard the configurations in $\underline{\boldsymbol{x}}^N$ as independent, identically distributed samples drawn from the distribution Q in $\mathcal{P}(\mathcal{X})$ (= the set of probability distributions on $\mathcal{X}$).

Then we take for each i the indicators

$$I_{a_i}\left(\boldsymbol{x}^{(n)}\right) = \begin{cases} 1 & \text{if } \boldsymbol{x}^{(n)} = a_i \\ 0 & \text{otherwise} \end{cases}$$

and denote the relative frequency of $a_i \in \mathcal{X}$ in $\underline{\boldsymbol{x}}^N$ by

$$P_{\underline{\boldsymbol{x}}^N}(a_i) = \frac{1}{N}\sum_{n=1}^{N} I_{a_i}\left(\boldsymbol{x}^{(n)}\right).$$

Authors in information theory, see [20, 23], call

$$\mathbf{p}_{\underline{\boldsymbol{x}}^N} = \{P_{\underline{\boldsymbol{x}}^N}(a_i)\}_{i=1}^{|\mathcal{X}|} \tag{6.1}$$

the **type** of of $\underline{\boldsymbol{x}}^N$, but point out also that this is nothing but the empirical distribution $\widehat{\mathbf{p}}_N$ (c.f., (4.12)) induced by $\underline{\boldsymbol{x}}^N$. Every type is in $\mathcal{P}(\mathcal{X})$. Then we define

$$\mathcal{P}_N = \text{ the set of types on } \mathcal{X} \text{ with fixed } N. \tag{6.2}$$

It may be of interest to observe the following fact.

Fact 10 If $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots \boldsymbol{X}^{(N)}$ are independent and identically distributed random variables distributed according to $Q(\boldsymbol{x})$, then the probability of $\underline{\boldsymbol{x}}^N$ depends only on its type, and, if Q^N is the product measure $\times_{i=1}^N Q$, then

$$Q^N\left(\underline{\boldsymbol{x}}^N\right) = e^{-N\left(H\left(P_{\underline{\boldsymbol{x}}^N}\right)+D\left(P_{\underline{\boldsymbol{x}}^N}\|Q\right)\right)}.$$

■

Let next $\mathcal{B} \subset \mathcal{P}(\mathcal{X})$ be a set of probability measures on $\mathcal{X}$, *which does not include the true measure* Q. Then the probability of interest (Q^N is the product measure $\times_{i=1}^N Q$) is

$$Pr\left(\widehat{\mathbf{p}}_N \in \mathcal{B}\right) = Q^N\left(\{\underline{\boldsymbol{x}}^N \mid \mathbf{p}_{\underline{\boldsymbol{x}}^N} \in \mathcal{B}\}\right).$$

We need one more definition. For $\Pi \subset \mathcal{P}(\mathcal{X})$ we set

$$D(\Pi \parallel Q) \stackrel{\text{def}}{=} \inf_{\mathbf{p} \in \mathcal{P}(\mathcal{X}) \bigcap \Pi} D(\mathbf{p} \parallel Q).$$

Then we may state the Sanov theorem, see [7, 37, 38].

Theorem 6.1.1 Let $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots \boldsymbol{X}^{(N)}$ be independent and identically distributed random variables distributed according to $Q(\boldsymbol{x})$. For $\mathcal{B} \in \mathcal{P}(\mathcal{X})$ such that $Q \notin \mathcal{B}$ and if the set $\mathcal{B}$ is such that

$$\lim_{N\to\infty} D\left(\mathcal{B} \bigcap \mathcal{P}_N \parallel Q\right) = D(\mathcal{B} \parallel Q), \tag{6.3}$$

then with

$$\mathbf{p}^* = \arg\min_{\mathbf{p}\in\mathcal{B}} D(\mathbf{p} \parallel Q).$$

we have

$$-\frac{1}{N} \ln Pr\left(\widehat{\mathbf{p}}_N \in \mathcal{B}\right) \to D(\mathbf{p}^* \parallel Q), \tag{6.4}$$

almost surely, as $N \to \infty$.

■

A pedagogical proof is found in [20, p. 293], who uses the method of types developed by Csiszár [23]. There are other conditions on $\mathcal{B}$ (than the one stated above) that also make the Sanov theorem valid, see, e.g., [23, p.2509].

A proof of the Sanov theorem starts with (we follow [23])

$$Pr\left(\widehat{\mathbf{p}}_N \in \mathcal{B}\right) = Q^N\left(\{\underline{\boldsymbol{x}}^N \mid \mathbf{p}_{\underline{\boldsymbol{x}}^N} \in \mathcal{B}\right) =$$

$$= \sum_{\mathbf{p} \in \mathcal{B} \bigcap \mathcal{P}_N} Q^N\left(A_{\mathbf{p}}^N\right),$$

where we have the type class

$$A_{\mathbf{p}}^N = \{\underline{\boldsymbol{x}}^N \in \mathcal{X}^N \mid \mathbf{p}_{\underline{\boldsymbol{x}}^N} = \mathbf{p}\}.$$

Then it holds by some algebra and combinatorics [20, ch. 12.1] that

$$\frac{1}{|\mathcal{P}(\mathcal{X})|} \cdot e^{-ND(\mathbf{p}\|Q)} \leq Q^N\left(A_{\mathbf{p}}^N\right) \leq e^{-ND(\mathbf{p}\|Q)}.$$

Hence we get

$$\frac{1}{|\mathcal{P}(\mathcal{X})|} \cdot e^{-ND(\mathcal{B}\bigcap\mathcal{P}_N\|Q)} \leq Pr\left(\widehat{\mathbf{p}}_N \in \mathcal{B}\right) \leq |\mathcal{P}(\mathcal{X})| e^{-ND(\mathcal{B}\bigcap\mathcal{P}_N\|Q)}$$

and the Sanov theorem follows.

6.2 Error event and error exponent

Next we shall present a portion of the recent work due to A. Willsky et.al. in [78]. First, it is proved using the large deviation techniques summarized above, that the most likely error in $\mathcal{T}_d^{\mathrm{ML}}(N)$, defined in the preceding chapter, is a spanning tree which differs from the true tree $\mathcal{T}_d^{(o)}$ by a single edge. The second result is the exact error exponent for the maximum likelihood estimation of $\mathcal{T}_d^{(o)}$.

The setting is simplified by assuming that $\mathcal{X}_i = \mathcal{X}$, so that any configuration $\boldsymbol{x}$ lies in $\mathcal{X}^d$.

Let $\underline{\boldsymbol{x}}^N \stackrel{def}{=} \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}$ be independent, identically distributed configurations in $\mathcal{X}^d$ drawn from $\mathbf{p}^{(o)}$ (the true but unknown distribution). The maximum likelihood Chow-Liu tree is $\mathcal{T}_d^{\mathrm{ML}}(N) = \left(\mathbf{l}, \sigma^{\mathrm{ML}}(N)\right) \in \mathbf{T}_d$. As above $\mathbf{p}_{\sigma^{(o)}}$ is the reverse I-projection of $\mathbf{p}^{(o)}$ and $\sigma^{(o)}$ is a corresponding Chow-Liu dependence structure (or spanning tree topology), so that $\mathcal{T}_d^{(o)} = \left(\mathbf{l}, \sigma^{(o)}\right)$.

The *error event* that the set of edges is not estimated correctly is

$$\mathcal{A}_N = \{\underline{\boldsymbol{x}}^N \mid \sigma^{\mathrm{ML}}(N) \neq \sigma^{(o)}\}. \tag{6.5}$$

Let $Pr = \times_1^N \mathbf{p}^{(o)}$ denote the N-fold product (sigma-additive) probability measure of sample $\underline{\boldsymbol{x}}^N$. The *error exponent* $K_{\mathbf{p}^{(o)}}$ is defined as

$$K_{\mathbf{p}^{(o)}} \stackrel{\mathrm{def}}{=} \lim_{N\to\infty} -\frac{1}{N} \ln Pr\left(\mathcal{A}_N\right), \tag{6.6}$$

in case the limit exists, which will be proved in the sequel. We can write

$$Pr\left(\mathcal{A}_N\right) \doteq e^{-NK_{\mathbf{p}^{(o)}}},$$

with the convention that if $\{a_n\}$ and $\{b_n\}$ are two sequences, we write $a_n \doteq b_n$ if and only if $\lim_{n\to\infty} \frac{1}{n} \ln \frac{a_n}{b_n} = 0$.

In words, if $K_{\mathbf{p}^{(o)}} > 0$, then the probability of error in maximum likelihood learning of Chow-Liu tree dependence decays exponentially in N, the number of training samples.

We are first going to define a simpler error event, the crossover error event, and find the error for it, and then exploit this to to derive $K_{\mathbf{p}^{(o)}}$.

6.3 Crossover error event and its error exponent

Since the maximum likelihood estimation of the Chow-Liu tree dependence finds the maximum weight spanning tree with the plug-in estimates of mutual informations as the edge weights, we are naturally led to consider the relative orders of plug-in estimates as a source of estimation error.

Let us consider two pairs of distinct nodes, $i, j, i', j' \in \mathbf{l}$, and set $e = (i, j)$ and $e' = (i', j')$. The marginal distributions (c.f. (2.5)) are

$$\mathbf{p}_e = (\mathbf{p}_{\sigma^{(o)}})^{\downarrow e} \in \mathcal{P}\left(\mathcal{X}^2\right), \quad \mathbf{p}_{e'} = (\mathbf{p}_{\sigma^{(o)}})^{\downarrow e'} \in \mathcal{P}\left(\mathcal{X}^2\right)$$

and

$$\mathbf{p}_{e,e'} = (\mathbf{p}_{\sigma^{(o)}})^{\downarrow (e,e')} \in \mathcal{P}\left(\mathcal{X}^4\right),$$

and induce the true mutual informations

$$\mathbf{I}^{(o)}(e), \quad \mathbf{I}^{(o)}(e').$$

We shall assume that the true distribution is such that

$$\mathbf{I}^{(o)}(e) > \mathbf{I}^{(o)}(e'). \tag{6.7}$$

Furthermore we have the empirical measure in (4.13) with its margins

$$\widehat{\mathbf{p}}_e = (\widehat{\mathbf{p}}_N)^{\downarrow e}, \quad \widehat{\mathbf{p}}_{e'} = (\widehat{\mathbf{p}}_N)^{\downarrow e'},$$

and

$$\widehat{\mathbf{p}}_{e,e'} = (\widehat{\mathbf{p}}_N)^{\downarrow e,e'} \in \mathcal{P}\left(\mathcal{X}_{e,e'}\right).$$

With these we compute the plug-in estimates

$$\widehat{\mathbf{I}}(e), \quad \widehat{\mathbf{I}}(e').$$

The *crossover event* (w.r.t. 6.7) is

$$\mathcal{C}_{e,e'} = \left\{\widehat{\mathbf{I}}(e') \geq \widehat{\mathbf{I}}(e)\right\}. \tag{6.8}$$

In words this event occurs, when we obtain a sample $\underline{\underline{\boldsymbol{x}}}^N$ such that the corresponding empirical distribution (or type, as discussed above) has margins such that $\widehat{\mathbf{I}}(e') \geq \widehat{\mathbf{I}}(e)$, while the true distribution satisfies (6.7).

As $N \to \infty$, the plug-in estimates $\widehat{\mathbf{I}}$ converge to the true values $\mathbf{I}^{(o)}$ with $\mathbf{p}^{(o)}$ - probability one. Hence the probability of $Pr\left(\widehat{\mathbf{p}}_{e,e'} \in \mathcal{C}_{e,e'}\right)$ falls to zero.

There seems to be a special difficulty with this formulation. This is that $\mathcal{C}_{e,e'}$ in $Pr\left(\widehat{\mathbf{p}}_{e,e'} \in \mathcal{C}_{e,e'}\right)$ depends also on the empirical distribution. Hence we shall consider the set of probability measures

$$\mathcal{B} = \{Q \in \mathcal{P}\left(\mathcal{X}^4\right) \mid \mathbf{I}(Q_{e'}) \geq \mathbf{I}(Q_e)\}. \tag{6.9}$$

Thus it holds that there is a crossover event, if it occurs that

$$\widehat{\mathbf{p}}_{e,e'} \in \mathcal{B}.$$

Hence $Pr\left(\widehat{\mathbf{p}}_{e,e'} \in \mathcal{B}\right) \to 0$, as $N \to \infty$. If this decrease is exponential, the *crossover error exponent* is

$$J_{e,e'} \stackrel{\text{def}}{=} \lim_{N\to\infty} -\frac{1}{N} \ln Pr\left(\widehat{\mathbf{p}}_{e,e'} \in \mathcal{B}\right). \tag{6.10}$$

In the theorem below we introduce the Kullback distance $D\left(Q \parallel \mathbf{p}_{e,e'}\right)$, where Q is a generic probability measure in $\mathcal{B} \subset \mathcal{P}\left(\mathcal{X}^4\right)$ and $\mathbf{p}_{e,e'}$ is the margin of the true distribution satisfying $\mathbf{I}^{(o)}(e) > \mathbf{I}^{(o)}(e')$.

We assume that $\mathbf{p}_{e,e'}$ is a positive distribution,i.e.,

$$p_{e,e'}\left(x_{i,j,i',j'}\right) > 0 \quad \text{for all } x_{i,j,i',j'} \in \mathcal{X}_{e,e'}.$$

This simplifies the proof, as we do not have to worry about possible infinities in the Kullback distances.

Theorem 6.3.1 Assume that $\mathbf{p}_{e,e'}$ is a positive distribution in $\mathcal{P}\left(\mathcal{X}^4\right)$ and that

$$\mathbf{I}^{(o)}(e) > \mathbf{I}^{(o)}(e'). \tag{6.11}$$

Then the crossover error rate $J_{e,e'}$ in (6.10) is given by

$$J_{e,e'} = \min_{Q\in\mathcal{P}(\mathcal{X}^4)\cap\Lambda} D\left(Q \parallel \mathbf{p}_{e,e'}\right), \tag{6.12}$$

where the constraint set is

$$\Lambda = \{Q \in \mathcal{P}\left(\mathcal{X}^4\right) \mid \mathbf{I}(Q_e) = \mathbf{I}(Q_{e'})\}, \tag{6.13}$$

and where Q_e and $Q_{e'} \in \mathcal{P}\left(\mathcal{X}^2\right)$ are defined for a $Q \in \mathcal{P}\left(\mathcal{X}^d\right)$ as

$$Q_e = Q^{\downarrow e}, Q_{e'} = Q^{\downarrow e'}$$

and satisfy

$$Q_e = \left(Q_{e,e'}\right)^{\downarrow e}, Q_{e'} = \left(Q_{e,e'}\right)^{\downarrow e'}.$$

The numbers $\mathbf{I}(Q_e)$ and $\mathbf{I}(Q_{e'})$ are the respective mutual informations.

The minimum in (6.12) is attained by some distribution $Q^*_{e,e'} \in \mathcal{P}\left(\mathcal{X}^4\right)$, which satisfies $\mathbf{I}(Q^*_e) = \mathbf{I}(Q^*_{e'})$ and $J_{e,e'} > 0$.

■

Proof: The proof is decomposed into four steps $I - IV$.

Step I invokes the *Sanov theorem* [7, 37] to show that the limit in (6.10) exists.

In Step II, we assume that a probability measure $Q^*_{e,e'}$ in $\mathcal{B}$ exists such that

$$D\left(Q^*_{e,e'} \parallel \mathbf{p}_{e,e'}\right) = \min_{Q \in \mathcal{B} \cap \mathcal{P}(\mathcal{X}^4)} D\left(Q \parallel \mathbf{p}_{e,e'}\right).$$

Then it is shown that $Q^*_{e,e'}$ must satisfy $\mathbf{I}(Q^*_{e'}) = \mathbf{I}(Q^*_e)$.

In Step III we prove the existence of the minimizer $Q^*_{e,e'}$, so that we may use min instead of inf in (6.12). These claims follow from the compactness of the constraint set Λ, and the Weierstrass extreme value theorem.

Step IV exploits the results in steps $I - III$ to show that $J_{e,e'} > 0$. This step requires the assumption that two selected pairs of nodes satisfy (6.11).

Step I As we assume that e, e' share no common nodes, then $\widehat{\mathbf{p}}_{e,e'} \in \mathcal{P}\left(\mathcal{X}^4\right)$. The Sanov theorem, see 6.1.1 in the Appendix, tells that

$$-\frac{1}{N} \ln Pr\left(\widehat{\mathbf{p}}_{e,e'} \in \mathcal{B}\right) \to \inf_{Q \in \mathcal{P}(\mathcal{X}^4) \cap \mathcal{B}} D\left(Q \parallel \mathbf{p}_{e,e'}\right), \tag{6.14}$$

for any set $\mathcal{B} \subseteq \mathcal{P}\left(\mathcal{X}^4\right)$ such that $\mathcal{B}$ satisfies certain extra properties to be given below, see also [23, p 2509] .

For the purpose of applying the Sanov theorem let us take

$$\mathcal{B} = \{Q \in \mathcal{P}\left(\mathcal{X}^4\right) \mid \mathbf{I}(Q_{e'}) \geq \mathbf{I}(Q_e)\}.$$

Then it holds that $\mathcal{B}$ is an open set, since $\mathbf{I}(Q_{e'})$ and $\mathbf{I}(Q_e)$ are continuous functions, so there can be no isolated points in $\mathcal{B}$ and $\mathcal{B}$ is not empty.

Let $\mathcal{P}_N$ be, c.f., (6.2), the set of types/empirical distributions on $\mathcal{X}_{e,e'}$ for $\underline{\boldsymbol{x}}^N$. Let us next set

$$\mathcal{B}_N = \mathcal{B} \cap \mathcal{P}_N$$

and compute

$$D\left(\mathcal{B}_N \parallel \mathbf{p}_{e,e'}\right) = \inf_{Q \in \mathcal{B} \cap \mathcal{P}_N} D\left(Q \parallel \mathbf{p}_{e,e'}\right),$$

which is finite, as $\mathcal{B}$ is non-empty. Similarly,

$$D\left(\mathcal{B} \parallel \mathbf{p}_{e,e'}\right) = \inf_{Q \in \mathcal{B} \cap \mathcal{P}(\mathcal{X}^4)} D\left(Q \parallel \mathbf{p}_{e,e'}\right),$$

Then, it follows by the openness of $\mathcal{B}$, as shown in [7, Lemma 5.2, p. 18] that

$$D\left(\mathcal{B}_N \parallel \mathbf{p}_{e,e'}\right) \to D\left(\mathcal{B} \parallel \mathbf{p}_{e,e'}\right),$$

as $N \to \infty$, which requires that $\mathbf{p}_{e,e'}$ is a positive distribution in $\mathcal{P}\left(\mathcal{X}^4\right)$, [7, loc.cit.]. This implies that the limit in (6.14) exists, see [7, p. 18−19].

Step II Let us now suppose that there exists Q^* in $\mathcal{B}$ such that

$$D\left(Q^* \parallel \mathbf{p}_{e,e'}\right) = \min_{Q \in \mathcal{B} \cap \mathcal{P}(\mathcal{X}^4)} D\left(Q \parallel \mathbf{p}_{e,e'}\right).$$

Now we establish that $Q^*_{e,e'}$, supposing it exists (as will be shown in Step III), will satisfy $\mathbf{I}(Q^*_{e'}) = \mathbf{I}(Q^*_e)$.

We define

$$g\left(Q\right) \stackrel{\text{def}}{=} \mathbf{I}\left(Q_{e'}\right) - \mathbf{I}\left(Q_e\right). \tag{6.15}$$

This is a continuous function, as mutual information for finite alphabets is composed of continuous functions.

As it is supposed that $Q^*_{e,e'}$ with the value $D\left(Q^*_{e,e'} \parallel \mathbf{p}_{e,e'}\right)$ is in $\mathcal{B}$, we may assume that it were $\mathbf{I}(Q^*_{e'}) > \mathbf{I}(Q^*_e)$. Then

$$g\left(Q^*_{e,e'}\right) > 0.$$

Since $g(\cdot)$ is continuous, there is a $\delta > 0$ such that

$$N_\delta\left(Q^*_{e,e'}\right) = \{R \mid\mid\mid R - Q^*_{e,e'} \parallel_\infty < \delta\},$$

where we use the variation distance defined in (A.17), so that image of $N_\delta\left(Q^*_{e,e'}\right)$ under $g(\cdot)$ is inside the open interval $(0, \infty)$.

Let us now consider the convex combination

$$Q^{**}_{e,e'} = (1 - \frac{1}{2\delta})Q^*_{e,e'} + \frac{1}{2\delta}\mathbf{p}_{e,e'}.$$

Then $Q^{**}_{e,e'} \in N_\delta\left(Q^*_{e,e'}\right)$ and hence is inside the set of feasible Q for

$$\inf_{Q \in \mathcal{B} \cap \mathcal{P}(\mathcal{X}^4)} D\left(Q \parallel \mathbf{p}_{e,e'}\right),$$

as $g\left(N_\delta\left(Q^*_{e,e'}\right)\right) \subseteq (0, \infty)$, i.e., $\mathbf{I}\left(Q^{**}_{e'}\right) - \mathbf{I}\left(Q^{**}_e\right) > 0$.

We shall now prove that

$$D\left(Q^{**}_{e,e'} \parallel \mathbf{p}_{e,e'}\right) < D\left(Q^*_{e,e'} \parallel \mathbf{p}_{e,e'}\right),$$

which contradicts the optimality of $Q^*_{e,e'}$. We have

$$D\left(Q^{**}_{e,e'} \parallel \mathbf{p}_{e,e'}\right) = D\left((1 - \frac{1}{2\delta})Q^*_{e,e'} + \frac{1}{2\delta}\mathbf{p}_{e,e'} \parallel \mathbf{p}_{e,e'}\right)$$

$$\leq (1 - \frac{1}{2\delta})D\left(Q^*_{e,e'} \parallel \mathbf{p}_{e,e'}\right) + \frac{1}{2\delta}D\left(\mathbf{p}_{e,e'} \parallel \mathbf{p}_{e,e'}\right),$$

since the Kullback distance is convex see Theorem A.1.7,

$$= (1 - \frac{1}{2\delta})D\left(Q^*_{e,e'} \parallel \mathbf{p}_{e,e'}\right)$$

since $D\left(\mathbf{p}_{e,e'} \parallel \mathbf{p}_{e,e'}\right) = 0$, and therefore

$$< D\left(Q^*_{e,e'} \parallel \mathbf{p}_{e,e'}\right).$$

Thus we conclude that the optimal solution must satisfy $\mathbf{I}(Q^*_{e'}) = \mathbf{I}(Q^*_e)$.

Step III As stated above, this step is to prove the existence of a minimizer $Q^*_{e,e'}$ so that we can write

$$D\left(Q^*_{e,e'} \parallel \mathbf{p}_{e,e'}\right) = \min_{Q \in \mathcal{P}(\mathcal{X}^4) \cap \Lambda} D\left(Q \parallel \mathbf{p}_{e,e'}\right),$$

where the constraint set Λ is defined in (6.13).

We note first that the Kullback distance $D\left(Q \parallel \mathbf{p}_{e,e'}\right)$ is real valued and continuous in both arguments (regarded as real vectors). Hence it suffices to prove that Λ in (6.13) is compact. It is clearly non-empty, since the uniform distribution $Q(\boldsymbol{x}_{e,e'}) = \frac{1}{|\mathcal{X}^4|}$ lies there. We shall show that Λ in (6.13) is bounded and closed. Boundedness is clear, since $\mathcal{P}\left(\mathcal{X}^4\right)$ is a bounded set of real numbers.

Thus with g from (6.15)

$$\Lambda = g^{-1}(0),$$

where $g^{-1}(0)$ is the inverse image of 0. Since $g(\cdot)$ is continuous, and $\{0\}$ is a closed set in the usual topology of the real line, it follows that $\mathcal{B}$ is closed. Hence there exists a minimizer $Q^*_{e,e'} \in \mathcal{B}$.

Step IV Hence, as $\Lambda \subset \mathcal{B}$, there exists $Q^* \in \mathcal{P}\left(\mathcal{X}^4\right)$ such that

$$D\left(Q^* \parallel \mathbf{p}_{e,e'}\right) = \min_{Q \in \cap \Lambda \cap \mathcal{P}(\mathcal{X}^4)} D\left(Q \parallel \mathbf{p}_{e,e'}\right).$$

As $\mathbf{p}_{e,e'}$ satisfies (6.11) in $\mathcal{P}\left(\mathcal{X}^4\right)$ and $Q^* \in \mathcal{B} \cap \mathcal{P}\left(\mathcal{X}^4\right)$, so that $D\left(Q^* \parallel \mathbf{p}_{e,e'}\right) > 0$. Hence the crossover error rate $J_{e,e'} > 0$ will be positive and is given as in (6.12).

■

6.4 Error Exponent for Structure Learning

Let us rewrite the error event in (6.5) as

$$\mathcal{A}_N = \{\underline{\boldsymbol{x}}^N \mid \mathcal{T}_d^{\mathrm{ML}}(N) \neq \mathcal{T}_d^{(o)}\} \tag{6.16}$$

and recall

$$K_{\mathbf{p}^{(o)}} \stackrel{\mathrm{def}}{=} \lim_{N \to \infty} -\frac{1}{N} \ln Pr\left(\mathcal{A}_N\right). \tag{6.17}$$

Then we define

$$\mathcal{U}_N\left(\mathcal{T}_d\right) = \begin{cases} \{\underline{\boldsymbol{x}}^N \mid \mathcal{T}_d^{\mathrm{ML}}(N) = \mathcal{T}_d\} & \text{if } \mathcal{T}_d \in \mathbf{T}_d \setminus \mathcal{T}_d^{(o)} \\ \emptyset & \text{otherwise} \end{cases} \tag{6.18}$$

Clearly $\mathcal{U}_N\left(\mathcal{T}_d\right) \cap \mathcal{U}_N\left(\mathcal{T}_d^{'}\right) = \emptyset$ as soon as $\mathcal{T}_d \neq \mathcal{T}_d^{'}$. Then we obtain in (6.16)

$$Pr\left(\mathcal{A}_N\right) = \sum_{\mathcal{T}_d \in \mathbf{T}_d \setminus \mathcal{T}_d^{(o)}} Pr\left(\mathcal{U}_N\left(\mathcal{T}_d\right)\right). \tag{6.19}$$

The large deviation exponent for each error event $\mathcal{U}_N(\mathcal{T}_d)$ is

$$\Upsilon(\mathcal{T}_d) = \lim_{N\to\infty} -\frac{1}{N}\ln Pr(\mathcal{U}_N(\mathcal{T}_d)), \tag{6.20}$$

whenever the limit exists. We identify the error event with the slowest rate of decay, or the *dominant error tree.*

Definition 6.4.1 A dominant error tree is a spanning tree $\mathcal{T}_d^* = (\mathbf{l}, \sigma^*)$ given by

$$\mathcal{T}_d^* = \arg \min_{\mathcal{T}_d \in \mathbf{T}_d \setminus \mathcal{T}_d^{(o)}} \Upsilon(\mathcal{T}_d). \tag{6.21}$$

■

A dominant error tree is, roughly speaking, the most likely candidate to be $\mathcal{T}_d^{\mathrm{ML}}(N)$ in case of error.

Theorem 6.4.1 The error exponent in (6.17) is the error exponent of the dominant error tree, or

$$K_{\mathbf{p}^{(o)}} = \Upsilon(\mathcal{T}_d^*). \tag{6.22}$$

Proof: By (6.19) and the convention on $\dot{=}$ we write

$$\begin{aligned} Pr(\mathcal{A}_N) &\doteq \sum_{\mathcal{T}_d \in \mathbf{T}_d \setminus \mathcal{T}_d^{(o)}} e^{-N\Upsilon(\mathcal{T}_d)} \\ &\doteq e^{-N\Upsilon(\mathcal{T}_d^*)}, \end{aligned} \tag{6.23}$$

by the principle of *worst exponent wins*, and since there is only a finite number of terms. ■

Consequently, the question is to find the dominant error tree. It turns out to be useful to study the crossover events (6.8). We consider thus a pair of nodes (or an edge in the complete graph) $e' = (u, v)$ such they are not neighbors in $\sigma^{(o)}$. There is, however, $\mathrm{Path}\big(e'; \sigma^{(o)}\big)$, defined as

$$\mathrm{Path}\big(e'; \sigma^{(o)}\big) \text{ a unique path of edges in } \sigma^{(o)} \text{ that connects } u \text{ and } v. \tag{6.24}$$

The argument for this is depicted in Figure 6.1. The edge e' and $\mathrm{Path}\big(e'; \sigma^{(o)}\big)$ form a cycle. Hence, if we remove any edge in $\mathrm{Path}\big(e'; \sigma^{(o)}\big)$ from $\sigma^{(o)}$ and replace it with e', the resulting set of edges

$$\Big(\sigma^{(o)} \setminus \{e\}\Big) \cup \{e'\}$$

is still a spanning tree.

Hence all such replacements are feasible outputs of the Chow-Liu algorithm, i.e., of maximum likelihood estimation. We do not, fortunately, need to consider all crossover events. By the worst exponent wins - principle we need only consider the crossover event between a non-neighbour node pair e' and its *dominant replacement edge*, $r(e')$, to be defined next, when determining $K_{\mathbf{p}^{(o)}}$.

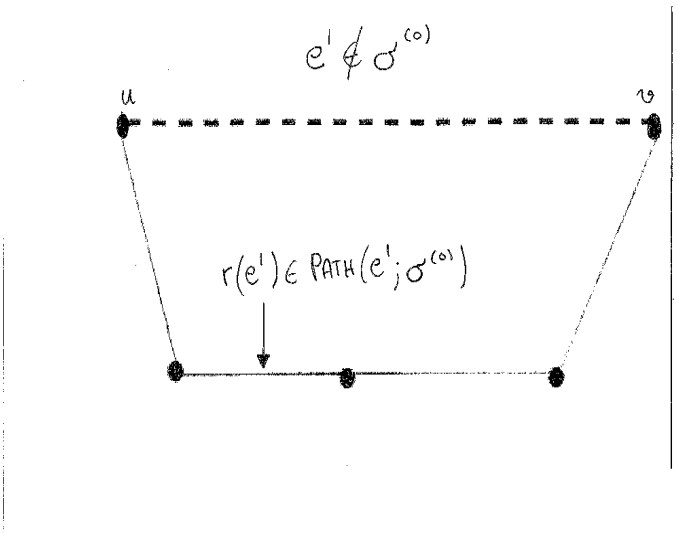

Figure 6.1: Dominant replacement edge

Definition 6.4.2 For each edge e, not in $\sigma^{(o)}$, its *dominant replacement edge* $r\left(e,\right) \in \sigma^{(o)}$ is defined as the edge in the unique path along $\sigma^{(o)}$ connecting the nodes in e, with the minimum crossover rate

$$r\left(e,\right) \stackrel{\text{def}}{=} \arg \min_{e \in \text{Path}\left(e,, \sigma^{(o)}\right)} J_{e,e,}, \tag{6.25}$$

where the crossover rate $J_{e,e,}$ is given in (6.12).

■

The crossover rate $J_{e,e,}$ in (6.12) is by construction defined for two pairs of nodes with no common node.

At this point we are ready to characterize the structure learning error exponent $K_{\mathbf{p}^{(o)}}$ in terms of the crossover rate between non-neighbour pairs and their dominant replacement edges.

Theorem 6.4.2 The error exponent for $\mathcal{A}_N = \{\underline{\boldsymbol{x}}^N \mid \mathcal{T}_d^{\text{ML}}(N) \neq \mathcal{T}_d^{(o)}\}$, i.e.,

the maximum likelihood tree estimation error exponent is given by

$$K_{\mathbf{p}^{(o)}} = J_{r(e^*),e^*} = \min_{e' \notin \mathcal{T}_d^{(o)}} \min_{e \in \mathrm{Path}\left(e',\sigma^{(o)}\right)} J_{e,e'}, \tag{6.26}$$

where $r\left(e^*\right)$ is the dominant replacement edge, defined in (6.25), associated to $e^* \in \sigma^{(o)}$ and e^* is the optimizing non-neighbor node pair

$$e^* = \arg \min_{e' \notin \sigma^{(o)}} J_{r(e'),e'}. \tag{6.27}$$

The dominant error tree $\mathcal{T}_d^* = (\mathbf{l}, \sigma^*)$ in (6.21) has the edge set

$$\sigma^* = \sigma^{(o)} \cup \{e^*\} \setminus \{r\left(e^*\right)\}. \tag{6.28}$$

Proof: We shall again decompose the proof into main steps and subcases. In Step I we show that σ^* in $\mathcal{T}_d^*$ will differ from $\sigma^{(o)}$ in exactly one edge.

In Step II the probability $Pr\left(\mathcal{A}_N\right)$ is bounded upwards by the elementary union bound (i.e., $Pr(A \bigcup B) \leq Pr(A) + Pr(B)$) and worst exponent wins - principle to conclude the that the rate that dominates is the minimum of $J_{r(e'),e'}$ over all possible non-neighbour node pairs $e' \notin \sigma^{(o)}$.

In Step III the probability $Pr\left(\mathcal{A}_N\right)$ is readily bounded downwards, which completes the proof.

Step I We show that σ^* in $\mathcal{T}_d^*$ will differ from $\sigma^{(o)}$ in exactly one edge. To do this, assume the contrary, that $\mathcal{T}_d^*$ differs from $\mathcal{T}_d^{(o)}$ by two edges. Let

$$\mathcal{T}_d^{\mathrm{ML}} = \mathcal{T}_d^{(o)} \setminus \{e_1, e_2\} \cup \{e_1', e_2'\},$$

where $e_1', e_2' \notin \sigma^{(o)}$ are the two edges that have replaced $e_1, e_2 \in \sigma^{(o)}$, respectively.

Since $\mathcal{T}_d^{ML} = \left(\mathbf{l}, \sigma^{\mathrm{ML}}\right)$ is a tree, these edges must satisfy (recall (6.24))

$$\{e_1, e_2\} \in \mathrm{Path}\left(e_1', \sigma^{(o)}\right) \cup \mathrm{Path}\left(e_2', \sigma^{(o)}\right)$$

for the tree property to be satisfied. Now we shall study $\Upsilon\left(\mathcal{T}_d^{\mathrm{ML}}\right)$ to establish the crucial property of the dominant error tree.

Case i) For $i = 1, 2$, $e_i \in \mathrm{Path}\left(e_i', \sigma^{(o)}\right)$ and $e_i \notin \mathrm{Path}\left(e_j', \sigma^{(o)}\right)$ for $i, j = 1, 2$, $i \neq j$. See Figure 6.2. We note first that

$$\mathbf{I}\left(\left(\mathbf{p}^{(o)}\right)_{e_i}^{\downarrow}\right) \geq \mathbf{I}\left(\left(\mathbf{p}^{(o)}\right)_{e_i'}^{\downarrow}\right)$$

in view of theorem 4.2.1. Let us recall (6.20)

$$\Upsilon\left(\mathcal{T}_d^{ML}\right) = \lim_{N \to \infty} -\frac{1}{N} \ln Pr\left(\mathcal{U}_N\left(\mathcal{T}_d^{ML}\right)\right),$$

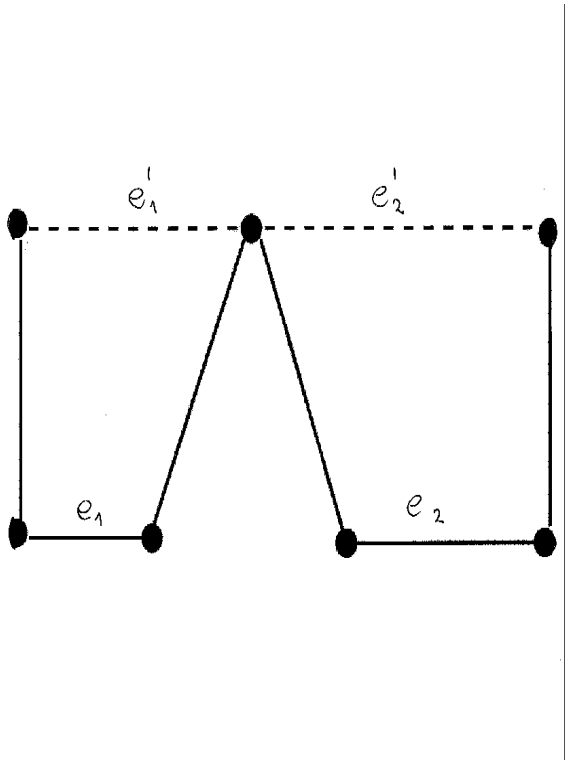

Figure 6.2: Case i)

We have that (by $P(A \bigcap B) \leq \max\{P(A), P(B)\}$)

$$\Upsilon\left(\mathcal{T}_d^{ML}\right) = \lim_{N\to\infty} -\frac{1}{N}\ln Pr\left(\bigcap_{i=1,2}\left(\mathbf{I}\left((\widehat{\mathbf{p}}_N)_{e_i}^{\downarrow}\right) \leq \mathbf{I}\left((\widehat{\mathbf{p}}_N)_{e_i'}^{\downarrow}\right)\right)\right)$$

$$\geq \max_{i=1,2}\lim_{N\to\infty} -\frac{1}{N}\ln Pr\left(\left(\mathbf{I}\left((\widehat{\mathbf{p}}_N)_{e_i}^{\downarrow}\right) \leq \mathbf{I}\left((\widehat{\mathbf{p}}_N)_{e_i'}^{\downarrow}\right)\right)\right)$$

$$= \max\{J_{e_1,e_1'}, J_{e_2,e_2'}\}. \tag{6.29}$$

We have that $J_{e_i,e_i'} = \Upsilon(\mathcal{T}_i)$, where $\mathcal{T}_i = \left(\mathbf{l}, \left(\sigma^{(o)} \setminus \{e_i\}\right) \cup \{e_i'\}\right)$. In view of theorem 6.4.1 the error exponent associated to the dominant error tree is $K_{\mathbf{p}^{(o)}} = \Upsilon(\mathcal{T}_d^*)$, where

$$\mathcal{T}_d^* = \arg \min_{\mathcal{T}_d \in \mathbf{T}_d \setminus \mathcal{T}_d^{(o)}} \Upsilon(\mathcal{T}_d).$$

Hence from (6.29), $\mathcal{T}_d^{ML}$ cannot be the dominant error tree, and the dominant error tree should differ from $\mathcal{T}^{(o)}$ by one and only one node.

Cases The additional cases can be handled in similar manner.

Step II

$$Pr\left(\mathcal{A}_N\right) = Pr\left(\bigcup_{e_{,}\notin\sigma^{(o)}}\left\{e_{,}\text{ replaces any } e\in\text{Path}\left(e_{,};\sigma^{(o)}\right)\text{ in }\mathcal{T}_d^{ML}\right\}\right)$$

$$= Pr\left(\bigcup_{e_{,}\notin\sigma^{(o)}}\bigcup_{e\in\text{Path}\left(e_{,};\sigma^{(o)}\right)}\left\{e_{,}\text{ replaces } e\text{ in }\mathcal{T}_d^{ML}\right\}\right)$$

and by the union bound

$$\leq \sum_{e_{,}\notin\sigma^{(o)}}\sum_{e\in\text{Path}\left(e_{,};\sigma^{(o)}\right)} Pr\left(\left\{e_{,}\text{ replaces } e\text{ in }\mathcal{T}_d^{ML}\right\}\right)$$

and by definition of crossover rate and as the rate of the event the output of the Chow-Liu $\mathcal{T}_d^{ML}$ is (6.20),

$$\leq \sum_{e_{,}\notin\sigma^{(o)}}\sum_{e\in\text{Path}\left(e_{,};\sigma^{(o)}\right)} Pr\left(\left\{\mathbf{I}\left(\left(\widehat{\mathbf{p}}_N\right)_e^{\downarrow}\right)\leq\mathbf{I}\left(\left(\widehat{\mathbf{p}}_N\right)_{e,}^{\downarrow}\right)\right\}\right)$$

$$\doteq \sum_{e_{,}\notin\sigma^{(o)}}\sum_{e\in\text{Path}\left(e_{,};\sigma^{(o)}\right)} e^{-NJ_{e,e_{,}}}$$

$$\doteq e^{-N\min_{e_{,}\notin\sigma^{(o)}}\min_{e\in\text{Path}\left(e_{,};\sigma^{(o)}\right)}J_{e,e_{,}}},$$

where we used the worst exponent wins principle once more. Thus we conclude

$$Pr\left(\mathcal{A}_N\right)\leq e^{-NJ_{r(e^*),e^*}}$$

by the definition of the dominant replacement edge $r\left(e_{,}\right)$ and the dominant non-neigbour edge e^* in (6.25) and (6.27), respectively.

Step III The lower bound is trivial from the observation that if $e^*\notin\sigma^{(o)}$ replaces $r\left(e^*\right)$, then the error $\mathcal{A}_N$ occurs, or

$$\left\{e^{ast}\notin\sigma^{(o)}\text{ replaces } r\left(e^*\right)\right\}\subset\mathcal{A}_N$$

and

$$Pr\left(\mathcal{A}_N\right)\geq Pr\left(\left\{e^*\notin\sigma^{(o)}\text{ replaces } r\left(e^*\right)\right\}\right)$$

$$\doteq e^{-NJ_{r(e^*),e^*}}.$$

Hence

$$Pr\left(\mathcal{A}_N\right)\doteq e^{-NJ_{r(e^*),e^*}},$$

which proves the assertion. ∎

Chapter 7

Stochastic Complexity and the Predictive Chow-Liu Likelihood

7.1 Redundant Source Coding and Statistical Learning

Let us now modify our notational apparatus to the effect that the distribution $\mathbf{p}$ in $\mathcal{P}(\mathcal{X})$ depends (continuously) on a real valued *parameter* θ in a compact $\Theta \subset R^n$. We write this as $\mathbf{p}_\theta$. Let the corresponding joint probability of independent and identically distributed random variables $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ with distribution $\mathbf{p}_\theta$ be

$$\mathbf{p}_\theta^N = \left\{ \prod_{i=1}^{N} p_\theta \left(\boldsymbol{x}^{(i)} \right) \right\}_{\boldsymbol{x}^{(1)} \in \mathcal{X}, \ldots, \boldsymbol{x}^{(N)} \in \mathcal{X}}.$$

We take a generic probability measure $\mathbf{q} \in \mathcal{P}\left(\mathcal{X}^N\right)$. Then the corresponding Kullback distance is written as

$$D\left(\mathbf{p}_\theta^N \parallel \mathbf{q}\right).$$

Let now $w(\theta)$ be a prior probability density on Θ. Then we compute

$$R_w(\mathbf{q}) \stackrel{\text{def}}{=} \int_\Theta D\left(\mathbf{p}_\theta^N \parallel \mathbf{q}\right) dw(\theta). \tag{7.1}$$

We shall now find the $\mathbf{q}$ minimizing $R_w(\mathbf{q})$.

Fact 11 Let next

$$\mathbf{m}^* = \arg \min_{\mathbf{q} \in \mathcal{P}(\mathcal{X}^N)} R_w(\mathbf{q}). \tag{7.2}$$

Then (see [2])

$$m^*\left(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}\right)=\int_\Theta\prod_{i=1}^N p_\theta\left(\boldsymbol{x}^{(i)}\right)dw(\theta). \tag{7.3}$$

To prove this we write (by some abuse of our notations) that

$$R_w\left(\mathbf{q}\right)=\int_\Theta D\left(\mathbf{p}_\theta^N\parallel\mathbf{m}^*\right)dw(\theta)+\int_\Theta\sum_{\boldsymbol{x}\in\mathcal{X}^N}\mathbf{p}_\theta^N\left(\boldsymbol{x}\right)\ln\left[\frac{\mathbf{m}^*(\boldsymbol{x})}{q(\boldsymbol{x})}\right]dw(\theta),$$

$$=\int_\Theta D\left(\mathbf{p}_\theta^N\parallel\mathbf{m}^*\right)dw(\theta)+\sum_{\boldsymbol{x}\in\mathcal{X}^N}\left[\int_\Theta\mathbf{p}_\theta^N\left(\boldsymbol{x}\right)dw(\theta)\right]\ln\left[\frac{\mathbf{m}^*(\boldsymbol{x})}{q(\boldsymbol{x})}\right]$$

$$=\int_\Theta D\left(\mathbf{p}_\theta^N\parallel\mathbf{m}^*\right)dw(\theta)+\sum_{\boldsymbol{x}\in\mathcal{X}^N}\mathbf{m}^*(\boldsymbol{x})\ln\left[\frac{\mathbf{m}^*(\boldsymbol{x})}{q(\boldsymbol{x})}\right].$$

Thus the conclusion follows.

■

Fact 12 There is furthermore a result, see [14, ch. 1.2.2], establishing that

$$\min_{\mathbf{q}}\max_{\Theta}D\left(\mathbf{p}_\theta^N\parallel\mathbf{q}\right)=\max_w\int_\Theta D\left(\mathbf{p}_\theta^N\parallel\mathbf{m}^*\right)dw(\theta).$$

Furthermore, the least favorable w can often be taken as a *Jeffrey's prior*, see section D.2 for a intuitive definition.

■

The distribution $\mathbf{m}^*$ is called the *predictive distribution* of $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_N$. Furthermore, if $X^N=\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}\}$,

$$SC\left(X^N;\mathcal{M}\right)\stackrel{\text{def}}{=}-\ln m^*\left(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}\right)$$

used to be called the *stochastic complexity* [89] of the family of models

$$\mathcal{M}=\{\mathbf{p}_\theta\mid\theta\in\Theta\}.$$

This does not necessarily presuppose an underlying true distribution in $\mathcal{M}$ or elsewhere. Jorma Rissanen, who introduced the concept of stochastic complexity in the sense stated above, has himself moved further to a revised notion of stochastic complexity, see, e.g., [64, 65], which is free from dependence on the prior w.

Stochastic complexity can be understood as a generalization of Shannon's information, $-\log\prod_{i=1}^N p_\theta\left(\boldsymbol{x}^{(i)}\right)$, which depends on a single θ, whereas $SC\left(X^N;\mathcal{M}\right)$

is defined with respect to the whole $\mathcal{M}$. Rissanen has shown, see e.g. [65], that SC plays an essential role in statistical learning, as SC is a tight lower bound on the total predictive code length, and as it is a basis for the MDL (= minimum description length) principle in model choice. MDL approximates SC in a non-predictive manner, and asserts that the best model in $\mathcal{M}$ explaining X^N is the one which can best compress the data X^N adding to the length the bits needed for the compression of the model.

In [9] V. Balasubramanian uses tools of statistical mechanics to introduce the notion of *razor* of a model family, which can be seen as a version of SC under the assumption of a true distribution generating the data. The razor is a measure of the trade-off between simplicity and accuracy in description of the data X^N sampled from a true distribution. A formula for the razor is found in Appendix D.

We shall next compute the predictive distribution and stochastic complexity for a family of $\mathbf{p}_\theta \in \mathcal{P}(\mathcal{X}, \mathbf{T}_d)$, i.e., for tree dependent distributions. In the family of models we restrict ourselves to binary alphabets and a related parameter space, which makes it possible to use a suitable special representation of $\mathbf{p}_\theta^N$.

7.2 Chow Representation of Likelihood

We consider the Chow-Liu dependence structure with binary alphabet $\mathcal{X}_i$ at each node of the tree, i.e., $x_i \in \{0, 1\}$. Let B^d denote the binary hypercube and let $\boldsymbol{x}$ denote its elements with d binary components, i.e.,

$$\boldsymbol{x} \in B^d = \left\{ \boldsymbol{x} \mid \boldsymbol{x} = (x_i)_{i=1}^d, x_i \in \{0, 1\} \right\}.$$

Let us now consider the generic joint Chow-Liu tree dependent distribution factorized, for some permutation of nodes, so that

$$p(\boldsymbol{x}) = p_1(x_1) \cdot p(x_2|x_{j_2}) \ldots p\left(x_{d-1}|x_{j_{d-1}}\right) \cdot p(x_d|x_{j_d}). \tag{7.1}$$

The conditional probabilities $p(x_i|x_{j_i})$ are written as

$$p(x_i|x_{j_i}) = \left(\theta_i^{x_i}(1-\theta_i)^{(1-x_i)}\right)^{x_{j_i}} \cdot \left(\phi_i^{x_i}(1-\phi_i)^{(1-x_i)}\right)^{1-x_{j_i}}, \tag{7.2}$$

where for $i = 2, \ldots, d$ we have the (unknown) parameters

$$\theta_i = p(x_i = 1|x_{j_i} = 1), \tag{7.3}$$

and

$$\phi_i = p(x_i = 1|x_{j_i} = 0), \tag{7.4}$$

and

$$p(x_1) = \theta_1^{x_1}(1-\theta_1)^{(1-x_1)}.$$

We introduce

$$p_{\theta_i,\phi_i}(x_i|x_{j_i}) \equiv p(x_i|x_{j_i}) \tag{7.5}$$

and

$$p_{\theta_1}(x_1) \equiv p(x_1)$$

We set from (7.3) and (7.4)

$$\underline{\theta} = (\theta_1, \ldots, \theta_d), \quad \underline{\phi} = (\phi_2, \ldots, \phi_d).$$

The probability of the configuration $\boldsymbol{x}$ given the tree $\mathcal{T}_d$ and the parameters $\underline{\theta}$ and $\underline{\phi}$ from (7.2) is now represented as

$$p_{\underline{\theta},\underline{\phi}}(\boldsymbol{x}) = p_{\theta_1}(x_1) \cdot \prod_{i=2}^{d} p_{\theta_i,\phi_i}(x_i|x_{j_i}). \tag{7.6}$$

Let $X^t = \left\{\boldsymbol{x}^{(l)}\right\}_{l=1}^t$ be a set of configuations $\boldsymbol{x}^{(l)} \in B^d$, which are seen as independent samples of the variables labelling the tree $\mathcal{T}_d$. Note that the data is supposed to be complete, no binary symbol are missing in any $\boldsymbol{x}^{(l)}$. Then we obtain from (7.6) the *Chow representation*

$$\prod_{l=1}^{t} p_{\underline{\theta},\underline{\phi}}(\boldsymbol{x}^{(l)}) = \theta_1^{n_1}(1-\theta_1)^{t-n_1} \prod_{i=2}^{d} \theta_i^{n_i(1,1)}(1-\theta_i)^{n_i(0,1)} \cdot \phi_i^{n_i(1,0)}(1-\phi_i)^{n_i(0,0)}, \tag{7.7}$$

where, as all $\boldsymbol{x}$ are instantiations of the same dependence tree,

$$n_i(1,1) = \sum_{l=1}^{t} x_i^{(l)} x_{j_i}^{(l)}, \quad n_i(1,0) = \sum_{l=1}^{t} x_i^{(l)} \left(1 - x_{j_i}^{(l)}\right),$$

$$n_i(0,1) = \sum_{l=1}^{t} \left(1 - x_i^{(l)}\right) x_{j_i}^{(l)}, \quad n_i(0,0) = \sum_{l=1}^{t} \left(1 - x_i^{(l)}\right)\left(1 - x_{j_i}^{(l)}\right),$$

for $i = 2, \ldots, d$, and

$$n_1 = \sum_{l=1}^{t} x_1^{(l)}.$$

Obviously, $n_i(1,1)$ counts the number of times we have simultaneously $x_i^{(l)} = 1$ and $x_{j_i}^{(l)} = 1$ in $X^t = \left\{\boldsymbol{x}^{(l)}\right\}_{l=1}^t$ and the interpretations of the rest of the counts above are also obvious. We set also

$$n_i(1), n_{j_i}(1), n_i(0), n_{j_i}(0)$$

as the count of ones (1) in X^t at position i e.t.c..

Let next Θ and Φ denote respective copies of the d-fold and $d-1$-fold product of the unit interval, then $\underline{\theta} \in \Theta$ and $\underline{\phi} \in \Phi$.

Given a prior probability density $w\left(\underline{\theta}, \underline{\phi}\right)$ on $\Theta \times \Phi$, we obtain the predictive likelihood corresponding to the Chow expansion as

$$m\left(X^t|\mathcal{T}_d\right) = \int_{\Theta} \int_{\Phi} \prod_{l=1}^{t} p_{\underline{\theta},\underline{\phi}}\left(\boldsymbol{x}^{(l)}\right) w\left(\underline{\theta}, \underline{\phi}\right) d\underline{\theta} d\underline{\phi}. \tag{7.8}$$

We can choose the prior $w\left(\underline{\theta},\underline{\phi}\right)$ by *local meta independence* of parameters, see [25] for a precise definition of the fancy sounding concept,

$$w\left(\underline{\theta},\underline{\phi}\right)=\prod_{i=1}^{d}h\left(\theta_i\right)\prod_{i=2}^{d}g\left(\phi_i\right). \tag{7.9}$$

7.2.1 Expressions and an Asymptotic Expansion of the Stochastic Complexity of a Chow-Liu Tree

From the formulas above we shall derive an expression of the stochastic complexity for any Chow-Liu tree.

We assume that every $h\left(\theta_i\right)$ and $g\left(\phi_i\right)$ is a $\mathrm{Be}(\alpha_1,\alpha_2)$ distribution given in (D.1) in Appendix D.1.

Then, by applications of (7.7), the Beta integral in (D.2) and (7.9) on (7.8), we obtain the stochastic complexity for any tree $\mathcal{T}_d$ as

$$\begin{aligned}-\log m\left(X^t|\mathcal{T}_d\right)= \qquad & (7.10)\\ =\log\frac{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)}{\Gamma\left(\alpha_1+\alpha_2\right)}&+\log\frac{\Gamma\left(t+\alpha_1+\alpha_2\right)}{\Gamma\left(n_1+\alpha_1\right)\cdot\Gamma\left(t-n_1+\alpha_2\right)}\\ +\sum_{i=2}^{d}\log\frac{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)}{\Gamma\left(\alpha_1+\alpha_2\right)}&\frac{\Gamma\left(n_i(1,1)+n_i(0,1)+\alpha_1+\alpha_2\right)}{\Gamma\left(n_i(1,1)+\alpha_1\right)\cdot\Gamma\left(n_i(0,1)+\alpha_2\right)}\\ +\sum_{i=2}^{d}\log\frac{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)}{\Gamma\left(\alpha_1+\alpha_2\right)}&\frac{\Gamma\left(n_i(1,0)+n_i(0,0)+\alpha_1+\alpha_2\right)}{\Gamma\left(n_i(1,0)+\alpha_1\right)\cdot\Gamma\left(n_i(0,0)+\alpha_2\right)},\end{aligned}$$

In the next step all the hyperparameters are chosen as $\alpha_1=\alpha_2=1/2$. This defines in fact the prior w a product of Jeffreys' priors in (7.9). With this we derive an asymptotic expansion of $-\log m\left(X^t|\mathcal{T}_d\right)$.

Theorem 7.2.1 If $m\left(X^t|\mathcal{T}_d\right)$ is evaluated assuming both local meta independence and Jeffreys' prior or the Beta distribution $\mathrm{Be}(1/2,1/2)$ for each of the parameters, then

$$\begin{aligned}-\log m\left(X^t|\mathcal{T}_d\right)=\log\frac{t!}{\Gamma\left(n_1+1/2\right)\cdot\Gamma\left(t-n_1+1/2\right)}&\\ +t\cdot\sum_{i=2}^{d}\mathbf{h}\left(n_i(1)/t\right)-t\cdot\sum_{i=2}^{d}\widehat{\mathbf{I}}_{i,j_i}&\\ +\frac{1}{2}\sum_{i=2}^{d}\log\left(n_{j_i}(1)\right)+\log\left(n_{j_i}(0)\right)+C,& \qquad (7.11)\end{aligned}$$

where $\mathbf{h}$ is the binary entropy function, and $\widehat{\mathbf{I}}_{i,j_i}$ is the empirical mutual information on the edge (i,j_i) in $\mathcal{T}_d$ and C is bounded in t.

■

7.2.2 Proof of the Representation in (7.10)

The integral on the right hand side of (7.8) will in view of (7.9) be factorized as

$$m\left(X^t|\mathcal{T}_d\right) = I_1 \cdot I_2 \cdot I_3, \tag{7.12}$$

where

$$I_1 = \int_0^1 \theta_1^{n_1} \left(1-\theta_1\right)^{t-n_1} h\left(\theta_1\right) d\theta_1, \tag{7.13}$$

$$I_2 = \prod_{i=2}^{d} \int_0^1 \theta_i^{n_i(1,1)} \left(1-\theta_i\right)^{n_i(0,1)} h\left(\theta_i\right) d\theta_i, \tag{7.14}$$

and

$$I_3 = \prod_{i=2}^{d} \int_0^1 \phi_i^{n_i(1,0)} \left(1-\phi_i\right)^{n_i(0,0)} g\left(\phi_i\right) d\phi_i. \tag{7.15}$$

There is an explicit expression for each of the factors I_1, I_2 and I_3, as the prior densities $h(\cdot)$ and $g(\cdot)$ are Beta densities. The Beta integral (D.2) gives, e.g., in each factor of I_2 in (7.14)

$$\int_0^1 \theta_i^{n_i(1,1)} \left(1-\theta_i\right)^{n_i(0,1)} h\left(\theta_i\right) d\theta_i =$$
$$\frac{\Gamma\left(\alpha_1+\alpha_2\right)}{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)} \frac{\Gamma\left(n_i(1,1)+\alpha_1\right)\cdot\Gamma\left(n_i(0,1)+\alpha_2\right)}{\Gamma\left(n_i(1,1)+n_i(0,1)+\alpha_1+\alpha_2\right)}.$$

Thus we have

$$I_2 = \prod_{i=2}^{d} \frac{\Gamma\left(\alpha_1+\alpha_2\right)}{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)} \frac{\Gamma\left(n_i(1,1)+\alpha_1\right)\cdot\Gamma\left(n_i(0,1)+\alpha_2\right)}{\Gamma\left(n_i(1,1)+n_i(0,1)+\alpha_1+\alpha_2\right)},$$

$$I_3 = \prod_{i=2}^{d} \frac{\Gamma\left(\alpha_1+\alpha_2\right)}{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)} \frac{\Gamma\left(n_i(1,0)+\alpha_1\right)\cdot\Gamma\left(n_i(0,0)+\alpha_2\right)}{\Gamma\left(n_i(1,0)+n_i(0,0)+\alpha_1+\alpha_2\right)},$$

as well as

$$I_1 = \frac{\Gamma\left(\alpha_1+\alpha_2\right)}{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)} \frac{\Gamma\left(n_1+\alpha_1\right)\cdot\Gamma\left(t-n_1+\alpha_2\right)}{\Gamma\left(t+\alpha_1+\alpha_2\right)}.$$

7.2.3 Proof of Theorem 7.2.1

With $\alpha_1 = \alpha_2 = 1/2$ we obtain the generic term (denoted by $E_{1/2}^{(2)}$) in I_2 (see (7.10)) as

$$\begin{aligned} E_{1/2}^{(2)} &\equiv \frac{\Gamma\left(\alpha_1+\alpha_2\right)}{\Gamma\left(\alpha_1\right)\cdot\Gamma\left(\alpha_2\right)} \frac{\Gamma\left(n_i(1,1)+\alpha_1\right)\cdot\Gamma\left(n_i(0,1)+\alpha_2\right)}{\Gamma\left(n_i(1,1)+n_i(0,1)+\alpha_1+\alpha_2\right)} \\ &= \frac{1}{\pi} \frac{\Gamma\left(n_i(1,1)+1/2\right)\cdot\Gamma\left(n_i(0,1)+1/2\right)}{\Gamma\left(n_i(1,1)+n_i(0,1)+1\right)}. \end{aligned}$$

By invoking, c.f. [46], a formula (D.5) for Stirling approximation of Euler's Gamma function Γ in $-\log E^{(2)}_{1/2}$, this entails

$$-\log E^{(2)}_{1/2} = (n_i(1,1)+n_i(0,1))\,\mathbf{h}\left(\hat{\theta}_i\right) + \frac{1}{2}\log\left(n_i(1,1)+n_i(0,1)\right) + C, \quad (7.16)$$

where C is a bounded in t and $\mathbf{h}(x) = -x\log x - (1-x)\log(1-x)$, $0 \le x \le 1$, is the binary entropy function (in natural logarithms) of the empirical distribution

$$\left(\hat{\theta}_i, 1-\hat{\theta}_i\right) = \left(\frac{n_i(1,1)}{n_i(1,1)+n_i(0,1)}, \frac{n_i(0,1)}{n_i(1,1)+n_i(0,1)}\right). \quad (7.17)$$

Here $\hat{\theta}_i$ is the maximum likelihood estimate (based on X^t) of $\theta_i = P\left(x_i = 1 | x_{j_i)} = 1\right)$.

For a generic term (denoted by $-\log E^{(3)}_{1/2}$) in $-\log I_3$ in (7.10) we obtain in the same way

$$-\log E^{(3)}_{1/2} = (n_i(1,0)+n_i(0,0))\,\mathbf{h}\left(\hat{\phi}_i\right) + \frac{1}{2}\log\left(n_i(1,0)+n_i(0,0)\right) + C, \quad (7.18)$$

where $\hat{\phi}_i$ is the maximum likelihood estimate of $\phi_i = P(x_i = 1 | x_{j_i} = 0)$.

Next we consider the result of inserting the terms

$$(n_i(1,1)+n_i(0,1))\,\mathbf{h}\left(\hat{\theta}_i\right)$$

and

$$(n_i(1,0)+n_i(0,0))\,\mathbf{h}\left(\hat{\phi}_i\right)$$

in the right hand side of (7.10). This gives the following expression

$$\sum_{i=2}^{d}\left[(n_i(1,1)+n_i(0,1))\,\mathbf{h}\left(\hat{\theta}_i\right) + (n_i(1,0)+n_i(0,0))\,\mathbf{h}\left(\hat{\phi}_i\right)\right].$$

The generic term in the sum is

$$(n_i(1,1)+n_i(0,1))\,\mathbf{h}\left(\hat{\theta}_i\right) + (n_i(1,0)+n_i(0,0))\,\mathbf{h}\left(\hat{\phi}_i\right).$$

This expression is by definition of the binary entropy function $\mathbf{h}$, see (A.3), equal to

$$\begin{aligned} &= -n_i(1,1)\log\left(\frac{n_i(1,1)}{n_i(1,1)+n_i(0,1)}\right) - n_i(0,1)\log\left(\frac{n_i(0,1)}{n_i(1,1)+n_i(0,1)}\right) \\ &- n_i(1,0)\log\left(\frac{n_i(1,0)}{n_i(1,0)+n_i(0,0)}\right) - n_i(0,0)\log\left(\frac{n_i(0,0)}{n_i(1,0)+n_i(0,0)}\right). \end{aligned} \quad (7.19)$$

Let us introduce the auxiliary quantities

$$n_{j_i}(1) = n_i(1,1)+n_i(0,1), n_{j_i}(0) = n_i(1,0)+n_i(0,0), \quad (7.20)$$

and

$$n_i(1) = n_i(1,1) + n_i(1,0), n_i(0) = n_i(0,1) + n_i(0,0). \tag{7.21}$$

Then we have as an identity from the right hand side of (7.19)

$$-n_i(1,1)\log\left(\frac{n_i(1,1)}{n_i(1,1)+n_i(0,1)}\right) - n_i(0,1)\log\left(\frac{n_i(0,1)}{n_i(1,1)+n_i(0,1)}\right)$$

$$-n_i(1,0)\log\left(\frac{n_i(1,0)}{n_i(1,0)+n_i(0,0)}\right) - n_i(0,0)\log\left(\frac{n_i(0,0)}{n_i(1,0)+n_i(0,0)}\right) =$$

$$= -n_i(1,1)\log\left(\frac{n_i(1,1)/t \cdot n_i(1)/t}{n_{j_i}(1)/t \cdot n_i(1)/t}\right) - n_i(0,1)\log\left(\frac{n_i(0,1)/t \cdot n_i(0)/t}{n_{j_i}(1)/t \cdot n_i(0)/t}\right)$$

$$-n_i(1,0)\log\left(\frac{n_i(1,0)/t \cdot n_i(1)/t}{n_{j_i}(0)/t \cdot n_i(1)/t}\right) - n_i(0,0)\log\left(\frac{n_i(0,0)/t \cdot n_i(0)/t}{n_{j_i}(0)/t \cdot n_i(0)/t}\right).$$

The quantities in the right hand side of the last equality can be regrouped as

$$-n_i(1,1)\log\left(n_i(1)/t\right) - n_i(0,1)\log\left(n_i(0)/t\right)$$

$$-n_i(1,0)\log\left(n_i(1)/t\right) - n_i(0,0)\log\left(n_i(0)/t\right)$$

$$-n_i(1,1)\log\left(\frac{n_i(1,1)/t}{n_{j_i}(1)/t \cdot n_i(1)/t}\right) - n_i(0,1)\log\left(\frac{n_i(0,1)/t}{n_{j_i}(1)/t \cdot n_i(0)/t}\right)$$

$$-n_i(1,0)\log\left(\frac{n_i(1,0)/t}{n_{j_i}(0)/t \cdot n_i(1)/t}\right) - n_i(0,0)\log\left(\frac{n_i(0,0)/t}{n_{j_i}(0)/t \cdot n_i(0)/t}\right).$$

The first four terms are equal to

$$-n_i(1,1)\log\left(n_i(1)/t\right) - n_i(0,1)\log\left(n_i(0)/t\right)$$

$$-n_i(1,0)\log\left(n_i(1)/t\right) - n_i(0,0)\log\left(n_i(0)/t\right) =$$

$$-\left(n_i(1,1)+n_i(1,0)\right)\log\left(n_i(1)/t\right) - \left(n_i(0,1)+n_i(0,0)\right)\log\left(n_i(0)/t\right) =$$

$$-n_i(1)\log\left(n_i(1)/t\right) - n_i(0)\log\left(n_i(0)/t\right) = t \cdot h\left(n_i(1)/t\right).$$

The remaining terms are

$$-n_i(1,1)\log\left(\frac{n_i(1,1)/t}{n_{j_i}(1)/t \cdot n_i(1)/t}\right) - n_i(0,1)\log\left(\frac{n_i(0,1)/t}{n_{j_i}(1)/t \cdot n_i(0)/t}\right)$$

$$-n_i(1,0)\log\left(\frac{n_i(1,0)/t}{n_{j_i}(0)/t \cdot n_i(1)/t}\right) - n_i(0,0)\log\left(\frac{n_i(0,0)/t}{n_{j_i}(0)/t \cdot n_i(0)/t}\right) =$$

$$-t \cdot \widehat{\mathbf{I}}_{i,j_i},$$

where, as in the preceding,

$$\widehat{\mathbf{I}}_{i,j_i} = \sum_{u=0}^{1}\sum_{v=0}^{1} \widehat{p}_{i,j_i}(u,v) \log \frac{\widehat{p}_{i,j_i}(u,v)}{\widehat{p}_i(u) \cdot \widehat{p}_{j_i}(v)} \tag{7.22}$$

is the plug-in estimate of the mutual information over the edge (i, j_i). ∎

7.2.4 Thresholding, Chow-Liu Forests and Checks of Significance

The theory and applications of the Chow-Liu algorithm and its extensions have been presented above. There is, however, an aspect of the methodology that seems to have received much less attention. This is the question of validity or statistical significance of the output of the algorithm. Needless to say, the method will produce a tree for completely random data when presented as input to CHOW-LIU-TREE Algorithm 1.

The statistical learning of the tree by minimizing $-\frac{1}{t}\log m\left(X^t|\mathcal{T}_d\right)$ in (7.11) as a function of the dependence structure $\mathcal{T}_d$. The purpose of stochastic complexity is to find a trade-off between accuracy of description of data and complexity of the model. We can do this in the following manner, c.f., [75].

First the Chow-Liu tree is estimated using the standard CHOW-LIU-TREE Algorithm 1. This gives $(i_r, j_r)_{r=1}^d$ (we renumber the nodes). Then (7.11) can be evaluated. If it happens that

$$\widehat{\mathbf{I}}_{i,j_i} < \frac{1}{2t}\left(\log\left(n_{j_i}(1)\right) + \log\left(n_{j_i}(0)\right)\right),$$

then one removes the edge (i, j_i) from the tree and the expression

$$-\widehat{\mathbf{I}}_{i,j_i} + \frac{1}{2t}\left(\log\left(n_{j_i}(1)\right) + \log\left(n_{j_i}(0)\right)\right)$$

is lowerbounded by zero in the right hand side (7.11). This improves (i.e.,reduces) the value of $-\frac{1}{t}\log m\left(X^t|\mathcal{T}_d\right)$. This thresholding technique can make the tree disconnected, and produces a forest of Chow-Liu trees. Clearly this procedure leads to reduction of stochastic complexity $-\log m\left(X^t|\mathcal{T}_d\right)$. A study of this is found in [27].

In [5] one derives asymptotic properties for uniform random spanning trees, see Appendix C, as the number of d increases to infinity. Since we are studying tree dependent probability distributions forlarge d, it should be possible to develop some suitable results [5] for testing the significance of an estimated Chow-Liu tree.

7.3 Classifiers aided by Chow-Liu Forests

The technique above is also applied to taxonomical analysis of various bacterial fingerprinting or to population genetic questions in [18, 32]. The general outline is as follows. We recall the notions in section 2.3. Thus C is an alphabet of class indices c. Then $(\boldsymbol{x}, c) \in B^d \times C$. We have

$$p\left(\boldsymbol{x}, c\right) = p\left(\boldsymbol{x} \mid c\right) q\left(c\right).$$

An algorithm for unsupervised classification of multivariate binary data that minimizes stochastic complexity is established. A classification is viewed as a

model of the data. The model is represented by class-conditional tree dependent probability distributions $p(\boldsymbol{x} \mid c)$. Within each class, trees are established from data by maximizing mutual information between pairs of nodes using a maximum weight spanning tree algorithm. The classifiers are trained by minimization of stochastic complexity (SC), which also selects the number of classes.

We define for any given class c_j the class membership indicators

$$u_j^{(l)} = 1 \text{ if } \mathbf{x}^{(l)} \in c_j$$

and

$$u_j^{(l)} = 0 \text{ otherwise,}$$

and incorporate these in the table

$$U^t = \left\{ u_j^{(l)} \right\}_{l=1,j=1}^{t,k}.$$

Then we extend the predictive likelihood in (7.8) to

$$m_k\left(X^t | U^t, \mathbf{T}_d\right) = \prod_{j=1}^{k} \int_{\Theta_j} \int_{\Phi} \prod_{l=1}^{t} \left[p_{\underline{\theta}_j, \underline{\phi}_j} \left(\boldsymbol{x}^{(l)} \mid \mathcal{T}_j \right) w\left(\underline{\theta}_j, \underline{\phi}_j\right) \right]^{u_j^{(l)}} d\underline{\theta}_j d\underline{\phi}_j$$

where $\mathbf{T}_d = \{\mathcal{T}_1 \ldots, \mathcal{T}_k\}$ (a tree for each class). This gives the stochastic complexity

$$m_k\left(X^t | U^t, \mathbf{T}_d\right) = -\log \prod_{j=1}^{k} \int_{\Theta_j} \int_{\Phi_j} \prod_{l=1}^{t} \left[p_{\underline{\theta}_j, \underline{\phi}_j} \left(\boldsymbol{x}^{(l)} \mid \mathcal{T}_j \right) w\left(\underline{\theta}, \underline{\phi}\right) \right]^{u_j^{(l)}} d\underline{\theta}_j d\underline{\phi}_j.$$

The expression of $m_k\left(X^t | U^t, \mathbf{T}_d\right)$ can be expanded as in theorem 7.2.1 We could think of finding a classification by maximizing this as function of U^t for every k. However, prevalence is needed for a complete account of stochastic complexity. This part of the problem is handled by a different manner in [18].

The prevalence of class c_j is $\lambda_j = \Pr(u_j^{(l)} = 1)$, and $\{\lambda_j\}_{1 \geq j \geq n}$ is a probability distribution over j.
We write

$$\Pr(U^t) = \int_{\Lambda} \prod_{j=1}^{k} \lambda_j^{t_j} \psi(\lambda) d\lambda$$

here t_j is the number of items in c_j. Then

$$\Pr(U^t) = \int_{\Lambda} \prod_{j=1}^{k} \lambda_j^{t_j} \psi(\lambda) d\lambda$$

We take the prior $\psi(\lambda)$ to be Jeffreys' prior (= a Dirichlet density with hyperparameters =1/2) and then

$$-\log \Pr(U^t) = \log \frac{\Gamma\left(\sum_{j=1}^{k} t_j + \frac{1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} \prod_{j=1}^{k} \frac{\Gamma\left(\frac{1}{2}\right)}{\Gamma\left(t_j + \frac{1}{2}\right)}.$$

Then the final criterion is minimization of

$$SC_k\left(X^t, U^t|\mathbf{T}_d\right) = -\log \Pr(U^t) - \log m_k\left(X^t|\mathbf{T}_d, U^t\right).$$

This is to be minimized as a function of $\mathbf{T}_d$ and U^t The interpretation is description length: $-\log m\left(X^t|\mathbf{T}_d, U^t\right)$ is the number of bits needed to describe the data vectors as members of the classes and $-\log \Pr(U^t)$ is the number of bits needed to describe the classes. We outline an algorithm for minimizing $SC_k\left(X^t, U^t|\mathbf{T}_d\right)$, c.f., [27].

Step 1. Fix k, set $w = 0$ and store an arbitrary (random) $U_{(w)}$.

Step 2. For $U_{(w)}$ find the maximum likelihood estimates $\widehat{\Theta}_{(w)}$ and $\widehat{\lambda}_{(w)}$ yielding

$$\frac{1}{t}\sum_{l=1}^{t}\sum_{j=1}^{k}\left[u_j^{(l)}\log p_{\underline{\widehat{\theta}}_j,\underline{\widehat{\phi}}_j}\left(x^{(l)}\right)\right] + \sum_{j=1}^{k}\widehat{\lambda}_j\log\widehat{\lambda}_j.$$

Step 3. Find the structure $\widehat{\mathcal{T}}_{d(w)}$ maximizing for each l

$$\frac{t_l}{t}\sum_{i=2}^{d} I_{i,j_i}$$

using the the algorithm MST-KRUSKAL 2. Apply the thresholding correction to the mutual information.

Step 4. Given $\widehat{\Theta}_{(w)}$, $\widehat{\lambda}_{(w)}$, and $\widehat{\mathcal{T}}_{d(w)}$ determine $U_{(w+1)} = \left\{(u_j^{(l)})_{(w+1)}\right\}_{j,l=1}^{t,k}$ using

$$(u_j^{(l)})_{(w+1)} = \begin{cases} 1 & \text{if } c_*^{(l)} = j \\ 0 & \text{otherwise,} \end{cases}$$

where

$$c_*^{(l)} = \arg\max_{1\le j\le k} p_{\underline{\widehat{\theta}}_j,\underline{\widehat{\phi}}_j}\left(x^{(l)} \mid c_j\right)\widehat{\lambda}_j.$$

Step 5. If $U_{(w+1)} = U_{(w)}$, then stop, otherwise set $w = w+1$ and go to 2..

A more streamlined idea is implemented in

Algorithm 3 MINIMIZE $SC_k\left(X^t, U^t|\mathbf{T}_d\right)$

1: given $\mathbf{T}_{(w)}$ determine $U_{(w+1)}$ by iteratively $\forall l$ by first removing $x^{(l)}$ from its cluster and then reinserting it where it reduces SC the most.
2: if then (k $= k_{\min}\|\mathrm{SC}\left[U_{(w)}\right] <$ SC $[U_{\max}]$) then $U_{\max} \leftarrow U_{(w)}$

Application 1 We describe an application to bacterial classification.

Over the past decade the advancement of molecular biology has led to the development of bacterial taxonomies based on detection of the naturally occurring DNA polymorphisms. These polymorphisms are a result of point mutations or rearrangements in the DNA.

A restriction endonuclease recognizes a specific sequence of nucleotide pairs and cleaves it. The number and locations of restriction sites vary with nucleotide sequence. The higher the similarity of the two DNA sequences compared, the closer the cleavage pattern.

The DNA fingerprinting technique known as AFLP (=Amplified Fragment Length Polymorphism) is based on the selective amplification of genomic restriction fragments by PCR (=Polymerase Chain Reaction) to differentiate bacterial strains at the subgeneric level and consists of three steps:

(a) Digestion of total cellular DNA with two restriction enzymes and ligation of restriction halfsite-specific adaptors to all restriction fragments.

(b) Selective amplification of these fragments with two PCR primers that have corresponding adaptor- and restriction-site-sequences as their target sites.

(c) Electrophoretic separation of the PCR products on a gel.

In electrophoresis, each organism is characterized by a banding pattern, which are thus a direct reflection of the genetic relationship between the bacterial strains examined and therefore these banding patterns can be considered as genomic fingerprints allowing computational algorithms for characterization (typing) and identification purposes. A collection of fingerprints can be translated into a set X^t of binary variables, band present (1) or band absent (0). In this application we discuss a classification of bacteria (suggesting an additional insight into the taxonomy in [81]) using the binary data and tree-aided classification.

The t =505 FAFLP (=Fluorescent Amplified Fragment Length Polymorphism, AFLP without radioactive materials) fingerprints of genomes of *Vibrio* strains listed in [81] were binarized. This gives us d =994 bit *Vibrionaceae* data. The preceding algorithm was then applied to this set of data with several randomized initial $U_{(w)}$. A recent survey of statistical analysis of AFLP is found in [11].

The following table gives some values for the complexities observed. The SC-values were computed using

$$SC_k\left(X^t, U^t|\mathbf{T}_d\right) = -\log P(U^t) - \log m_k\left(X^t|\mathbf{T}_d, U^t\right)$$

and by estimating a Chow-Liu tree for all classes in all classifications. In order to establish k, the number of classes, this algorithm is used repeatedly for different k. The table refers to Binclass, which is a software for minimizing stochastic complexity based on the naíve approximation of $\mathbf{p}$ [33]. This is an example, where the Chow-Liu tree has an appreciable impact, whereas there are sets of data, where the tree-aided classifiers do not gain much over the naíve classifiers.

		SC
1	Binclass (50 trials/cluster size)	156029.7
2	Thompson [81]	147960.1
3	Chow-Liu tree (5trials/cluster size)	121021.2

The next table shows the Rand similarities between the classifications above.

	1	2	3
1	1.0000	.98603	.77212
2	.98603	1.0000	.78031
3	.77212	.78031	1.0000

Because classification by minimization of SC compresses data, it comes as no surprise that the optimum number of clusters (classes) found by the algorithm turns out to be smaller than the number of clusters (=69) established in [81]. The last mentioned clustering is obtained from a hierarchic clustering method which are standard procedures in analysis of AFLP [11].

The classification minimizing SC should in this context be assessed with respect to its biological relevance. By inspection of the results one finds that 49 of the 69 clusters in Thompson have remained intact or have all but one or two of their strains in the same larger class in the classification minimizing SC.

Another way to look at the biological meaning of the result is by means of the class memberships of the *type strains*. The clusters in [81] can in broad terms be described as falling in three categories: (a) isolates with genomes related to only one known type strain, (b) isolates clustering to one or more type strains, and (c) isolates with genome unrelated to any type strain (less than 45 % similarity).

It is clear that Bayes rule (as applied in *Step 4.* above) does not measure similarity in the same explicit biological sense as, e.g., Dice's coefficient. Hence it is interesting to note that in the classification minimizing SC those 49 classes that have remained intact have the corresponding type strains in the same cluster in the classification minimizing SC or are clusters that do not have any type strain either in [81] or in the classification minimizing SC.

■

Appendix A

Appendix: Some Formulas of Information Theory

A.1 Kullback Distance

Let $\mathcal{X} = \{x_1, \cdots, x_L\}$ be an alphabet and let

$$\mathbf{f} := (f(x_1), \cdots, f(x_L))$$

and

$$\mathbf{g} := (g(x_1), \cdots, g(x_L))$$

be two probability distributions defined on $\mathcal{X}$. Then their *relative entropy* or the *Kullback distance* between $\mathbf{f}$ and $\mathbf{g}$ is defined by

$$D(\mathbf{f} \parallel \mathbf{g}) = \sum_{i=1}^{L} f(x_i) \log \frac{f(x_i)}{g(x_i)}. \tag{A.1}$$

Here we use the conventions $0 \cdot \log \frac{0}{g(x_i)} = 0$ and $f(x_i) \log \frac{f(x_i)}{0} = \infty$. The logarithm is the natural logarithm unless otherwise stated.

A.1.1 Entropy

The entropy $H(\mathbf{f})$ is defined as

$$H(\mathbf{f}) = -\sum_{i=1}^{L} f(x_i) \log f(x_i). \tag{A.2}$$

Suppose $\boldsymbol{X}$ is a random variable with the alphabet $\mathcal{X} = \{x_1, \cdots, x_L\}$ and distribution $\mathbf{f}$. Then we use $H(\boldsymbol{X})$ as another notation for the entropy in (A.2). $H(\boldsymbol{X})$ is a measure of the uncertainty in bits (=binary information units) of the random variable $\boldsymbol{X}$. It is also a lower bound for the number of bits (binary digits) needed on the average to describe the random variable, c.f., [65].

Example A.1.1 (Binary entropy function) For the special case $\mathcal{X} = \{x_1, x_2\}$, with $p := f_X(x_1)$,

$$\mathbf{h}(p) := -p \log_2(p) - (1-p) \log_2(1-p) \tag{A.3}$$

is the *(binary) entropy function*. This is also the entropy of a Bernoulli random variable $X \in Be(p)$ with $\mathcal{X} = \{0, 1\}$).

■

A.1.2 Examples

Example A.1.2 (Information Content) If $\boldsymbol{X}$ is a random variable with the distribution $\mathbf{f} = (f(x_1), \cdots, f(x_L))$, any probabilty distribution on an alphabet of L symbols, and if $\mathbf{g} = (1/L, \cdots, 1/L)$ is the uniform distribution, then

$$D(\mathbf{f} \parallel \mathbf{g}) = \log L - H(X).$$

This quantity is sometimes known as the *information content (of* $\mathbf{f}$*)*.

■

Example A.1.3 (Two Bernoulli distributions) Let $\mathcal{X} = \{0, 1\}$ and $0 \leq p \leq 1$ and $0 \leq g \leq 1$. Let $\mathbf{f} = (1-p, p)$ and $\mathbf{g} = (1-g, g)$ be the two Bernoulli distributions $Be(p)$ and $Be(g)$, respectively. Then

$$D(\mathbf{f} \parallel \mathbf{g}) = (1-p) \cdot \log \frac{1-p}{1-g} + p \cdot \log \frac{p}{g}. \tag{A.4}$$

We can also rewrite this as

$$D(\mathbf{f} \parallel \mathbf{g}) = -(1-p) \cdot \log(1-g) - p \cdot \log g - \mathbf{h}(p),$$

where $\mathbf{h}(p)$ is the binary entropy function (A.3) in natural logarithm.

■

A.1.3 Calibration

Let us set in (A.4) of the preceding example

$$D(Be(p) \parallel Be(g)) = -(1-p) \cdot \log(1-g) - p \cdot \log g - \mathbf{h}(p), \tag{A.5}$$

which is easily plotted as a function of g for any fixed p or vice versa. These plots suggest a number of the general properties of the relative entropy like that it is nonnegative, becomes zero only if $p = g$ and that the relative entropy is quadratic in small neighborhoods of p.

The cases where $D(\mathbf{f} \parallel \mathbf{g}) = 0$ and $D(\mathbf{f} \mid \mathbf{g}) = \infty$ are easily interpreted in an intuitive sense. There is a way to understand the other values of $D(\mathbf{f} \parallel \mathbf{g})$ by introducing the following idea of *calibration*. Let $D(\mathbf{f} \parallel \mathbf{g}) = k$ be the value of the Kullback distance between any two probability distributions on the same alphabet. Then we look for *calibration of* k or a number $g(k)$ such that the equation

$$D(Be(1/2) \parallel Be(g(k))) = k \tag{A.6}$$

is satisfied. This calibration tells us that the Kullback distance between $\mathbf{f}$ and $\mathbf{g}$ is the same as between a Bernoulli distribution $Be(1/2)$, a random or 'fair' choice of value in $\{0, 1\}$, and a Bernoulli distribution $Be(g(k))$. Therefore, the closer to $1/2$ the calibration is, the more similar $\mathbf{f}$ and $\mathbf{g}$ are. The calibration can be found explicitly, since

$$D(Be(1/2) \parallel Be(g)) = -\frac{1}{2}\log(4g(1-g)), \tag{A.7}$$

which gives an algebraic equation of second degree, where one value of the calibration can be solved as

$$g(k) = \frac{1}{2}\left(1 + \sqrt{1 - e^{-2k}}\right). \tag{A.8}$$

A.1.4 $D(\mathbf{f} \parallel \mathbf{g}) \geq 0$

Next we give a general proof of the important fact that the relative entropy is nonnegative.

Proposition A.1.4 *For any probability distributions* $\mathbf{f}$ *and* $\mathbf{g}$ *on the same alphabet*

$$D(\mathbf{f} \parallel \mathbf{g}) \geq 0. \tag{A.9}$$

Proof: Let $\boldsymbol{X}$ be a random variable that has the distribution $\mathbf{f}$. We write $p(x)$ and $g(x)$ for the generic values of the probability of $x \in \mathcal{X}$ in the two distributions. Then we have

$$D(\mathbf{f} \parallel \mathbf{g}) = E\left[\log \frac{p(\boldsymbol{X})}{g(\boldsymbol{X})}\right]$$

and this equals

$$D(\mathbf{f} \parallel \mathbf{g}) = -E\left[\log \frac{g(\boldsymbol{X})}{p(\boldsymbol{X})}\right].$$

Since $\phi(x) = -\log x$ is a convex function we have that

$$-E\left[\log \frac{g(\boldsymbol{X})}{p(\boldsymbol{X})}\right] \geq -\log E\left[\frac{g(\boldsymbol{X})}{p(\boldsymbol{X})}\right],$$

where we have used Jensen's inequality. But

$$E\left[\frac{g(\boldsymbol{X})}{p(\boldsymbol{X})}\right] = \sum_{i=1}^{L} f(x_i)\frac{g(x_i)}{f(x_i)} = 1$$

and since $\log 1 = 0$, we have proved our assertion. ∎

A.1.5 Further inequalities

The following way of rewriting (A.9) is known as

Gibbs' inequality :

$$\sum_{i=1}^{L} p_i \log \frac{1}{p_i} \leq \sum_{i=1}^{L} p_i \log \frac{1}{g_i} \tag{A.10}$$

for any $p_i \geq 0$ and $g_i \geq 0$ with $\sum_{i=1}^{L} p_i = \sum_{i=1}^{L} g_i = 1$. ∎

There is a sharper lower bound for Kullback distance. For this we prove a generalization of Gibbs' inequality that will be used below to show convexity of the Kullback distance.

Lemma A.1.5 *[Log sum inequality] If $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ are non-negative numbers, then*

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq \left(\sum_{i=1}^{n} a_i \right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}. \tag{A.11}$$

Proof: We assume that the numbers are positive, as can be done without loss of generality. Let us note that $\phi(t) = t \log t$ is a strictly convex function. This means that

$$\sum_{i=1}^{n} \lambda_i \phi(t_i) \geq \phi \left(\sum_{i=1}^{n} \lambda_i t_i \right),$$

where $\sum_{i=1}^{n} \lambda_i = 1$, $\lambda_i \geq 0$. Then we set $\lambda_i = b_j / \sum_{i=1}^{n} b_i$ and $t_i = a_i / b_i$ and obtain

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} b_i} \log \frac{a_i}{b_i} \geq \sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} b_i} \log \sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} b_i}$$

as was to be proved. ∎

Lemma A.1.6 For any $p_i \geq 0$ and $g_i \geq 0$ with $\sum_{i=1}^{L} p_i = \sum_{i=1}^{L} g_i = 1$ we have

$$\sum_{i=1}^{L} p_i \ln \frac{p_i}{g_i} \leq \sum_{i=1}^{L} \frac{(p_i - g_i)^2}{g_i}. \tag{A.12}$$

If there is any $g_i = 0$, then both sides are infinite.

Proof:

$$\sum_{i=1}^{L} p_i \ln \frac{p_i}{g_i} = \sum_{i=1}^{L} p_i \ln \left(1 + \left(\frac{p_i - g_i}{g_i} \right) \right).$$

We use now the elementary inequality[1] $\ln(1 + x) \leq x$ if $x \geq -1$. Then

$$\leq \sum_{i=1}^{L} p_i \left(\frac{p_i - g_i}{g_i} \right) \leq \sum_{i=1}^{L} ((p_i - g_i) + g_i) \left(\frac{p_i - g_i}{g_i} \right)$$

[1] The *IT-olikhet* (= IT-inequality), c.f., p. 19, R. Johansson: *Informationsteori- grundvalen för telekommunikation.* Studentlitteratur, Lund, 1988

$$= \sum_{i=1}^{L} \frac{(p_i - g_i)^2}{g_i} + \sum_{i=1}^{L} (p_i - g_i) = \sum_{i=1}^{L} \frac{(p_i - g_i)^2}{g_i}.$$

■

The quantity $\sum_{i=1}^{L} \frac{(p_i - g_i)^2}{g_i}$ is the χ^2 statistic. In fact one can show by a series expansion that

$$D(\mathbf{p} \parallel \mathbf{g}) = \sum_{i=1}^{L} p_i \ln \frac{p_i}{g_i} = \frac{1}{2}\chi^2 + \dots \tag{A.13}$$

A.1.6 The Convexity of the Kullback distance

The following theorem and the proof are taken from [20].

Theorem A.1.7 If $(\mathbf{f}_1, \mathbf{g}_1)$ and $(\mathbf{f}_2, \mathbf{g}_2)$ are two pairs of probability distributions on $\mathcal{X}$, then

$$D(\lambda\mathbf{f}_1 + (1-\lambda)\mathbf{f}_2 \parallel \lambda\mathbf{g}_1 + (1-\lambda)\mathbf{g}_2) \leq \lambda D(\lambda\mathbf{f}_1 \parallel \mathbf{g}_1) + (1-\lambda)D(\mathbf{f}_2 \parallel \mathbf{g}_2) \tag{A.14}$$

for all $0 \leq \lambda \leq 1$.

Proof: We apply the inequality (A.11) with a_1, a_2, b_1, b_2 as stated below, to a generic term in the left hand side of (A.14) to get

$$\overbrace{\lambda f_1(x_i)}^{=a_1} + \overbrace{(1-\lambda) f_2(x_i)}^{=a_2} \ln \frac{\lambda f_1(x_i) + (1-\lambda) f_2(x_i)}{\underbrace{\lambda g_1(x_i)}_{=b_1} + \underbrace{(1-\lambda) g_2(x_i)}_{=b_2}}$$

$$\leq \lambda f_1(x_i) \ln \frac{\lambda f_1(x_i)}{\lambda g_1(x_i)} + (1-\lambda) f_2(x_i) \ln \frac{(1-\lambda) f_2(x_i)}{(1-\lambda) g_2(x_i)}.$$

When this is summed over x_i, $i = 1, \dots, L$, we get the convexity as claimed. ■

A.1.7 Pinsker's Inequality

Let

$$\| \mathbf{f} - \mathbf{g} \|_{L_1} \stackrel{\text{def}}{=} \sum_{i=1}^{L} | f(i) - g(i) | .$$

Then we have

Lemma A.1.8 *[Pinsker's inequality]*

$$D(\mathbf{f} \parallel \mathbf{g}) \geq \frac{1}{2} \| \mathbf{f} - \mathbf{g} \|_{L_1}^2 . \tag{A.15}$$

■

A proof is found in [20, p.300]. It can easily be seen that ([20, p.299])

$$\frac{1}{2} \parallel \mathbf{f} - \mathbf{g} \parallel_{L_1} = \max_{B \subseteq \mathcal{X}} \mid P_{\mathbf{f}}(B) - P_{\mathbf{g}}(B) \mid, \tag{A.16}$$

where in the right hand we have $P_{\mathbf{f}}(B) = \sum_{x_i \in B} f(x_i)$ and analogously for $P_{\mathbf{g}}(B)$. The expression in the left hand side of (A.16) is the *variation distance* between $\mathbf{f}$ and $\mathbf{g}$ and is denoted by

$$\parallel \mathbf{f} - \mathbf{g} \parallel_{\infty} \stackrel{\text{def}}{=} \max_{B \subseteq \mathcal{X}} \mid P_{\mathbf{f}}(B) - P_{\mathbf{g}}(B) \mid . \tag{A.17}$$

A.2 A Diagram for Interpretation of Mutual Information and the Related Entropy Identities.

We copy in Figure A.1 the important diagram from [29, p. 431][2].

In the diagram we make the following identifications with respect to the notations in the main body of text:

$$\mathbf{I}(A, B) \leftrightarrow H_T$$

$$H(A, B) \leftrightarrow H(X, Y)$$

$$H(B) \leftrightarrow H(X), H(A) \leftrightarrow H(Y)$$

$$H(A|B) \leftrightarrow H(Y|X), H(B|A) \leftrightarrow H(X|Y)$$

Then the Figure is read as describing a flow of entropy (like a liquid) passing a noisy (leaky) channel from left to right. In the left a sender has a source of messages with the entropy $H(B)$, of which the amount $H(B|A)$ vanishes in the channel. Here $\mathbf{I}(A, B) = H(B) - H(B|A)$ is the difference.

The left side receives noisy messages with the entropy $H(A)$, of which $H(A|B)$ was not a part of the transmission into the channel but is due to noise. Thus the information received is $\mathbf{I}(A, B) = H(A) - H(A|B)$. Hence, by symmetry, $\mathbf{I}(A, B)$ is the same for the left end and the right end of the channel, and is properly called the mutual information. An umpire gets to see the messages in both ends of the channel, and has consequently the total entropy $H(A, B) = H(A) + H(B) - \mathbf{I}(A, B)$.

[2] A figure with the same insight is found in [51, p.149]

und (14)

$$H_T = H(y) - H(y|x) \quad \text{in bit/Zeichen} \tag{16b}$$

oder unter Beachtung von (14b) bzw. (15b)

$$H_T = H(x) + H(y) - H(x, y) \quad \text{in bit/Zeichen.} \tag{16c}$$

Der Transinformationsgehalt H_T kann als Mittelwert nie negativ werden, obwohl für ein einzelnes Paar x_i, y_j bei falscher Übertragung negative Werte vorkommen.

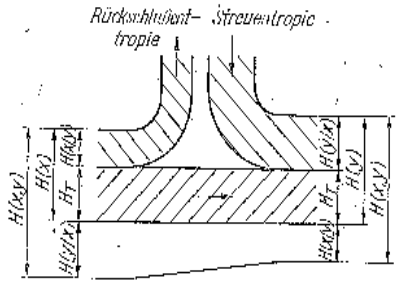

Bild 16.6
Schematisches Verteilungsdiagramm für Entropiebeiträge bei der Informationsübertragung

Die Bedeutung der einzelnen Entropiegrößen können wir durch ein schematisches Verteilungsdiagramm veranschaulichen (Bild 16.6). Daraus lesen wir ab: Durch die Störung wird ein Teil der Sendeentropie $H(x)$, die Rückschlußentropie $H(x|y)$, nicht ausgenutzt, und ein Teil der Empfangsentropie $H(y)$, die Streuentropie $H(y|x)$, stammt nicht von der Übertragung. Die tatsächlich übertragene Information ist nur der Transinformationsgehalt H_T.

Figure A.1: The relationships for mutual information

Appendix B

Appendix: Description Length for Storing of $\mathbf{p}_{\mathcal{S}}$

We compute the description length needed for storing $\mathbf{p}_{\mathcal{S}}$. In (2.29) we have

$$p\left(\boldsymbol{x} \mid \mathcal{S}\right) = p_{A_1}\left(\boldsymbol{x}_{A_1}\right) \prod_{i=2}^{k} p\left(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}\right), \boldsymbol{x} \in \mathcal{X}.$$

Consider (A_i, B_i). We describe first the number of variables in B_i by the cardinality $\mid B_i \mid$ followed by the index of the set B_i in some enumeration of all sets $\begin{pmatrix} d \\ \mid B_i \mid \end{pmatrix}$. We can encode the number $\mid B_i \mid$ using $\log_2 d$ bits and we can encode the index of the set B_i using $\log_2 \begin{pmatrix} d \\ \mid B_i \mid \end{pmatrix}$ bits. The same argument holds for A_i. So we get the description length (in logarithms to the base 2)

$$\mathrm{DL}_{(A_i,B_i)} = 2\log_2 d + \log_2 \begin{pmatrix} d \\ \mid B_i \mid \end{pmatrix} + \log_2 \begin{pmatrix} d \\ \mid A_i \mid \end{pmatrix}. \tag{B.1}$$

In addition we need to describe the probability table $p\left(\boldsymbol{x}_{A_i} \mid \boldsymbol{x}_{B_i}\right)$. We need to store $\mid B_i \mid (\mid A_i \mid -1)$ numbers. This representation length depends on the number of bits used for each numeric entry, which we count as $\frac{1}{2}\log_2 c$. Thus the encoding length of one conditional probability table is

$$\mathrm{DL}\left(A_i \mid B_i\right) = \frac{1}{2} \mid B_i \mid (\mid A_i \mid -1) \log_2 c.$$

For A_1, B_1 the description length is modified in an obvious manner.

Hence the total description length of $\mathcal{S}$ is

$$\mathrm{DL}_{\mathcal{S}} = \sum_{i=1}^{k} \left[2\log_2 d + \log_2 \begin{pmatrix} d \\ \mid B_i \mid \end{pmatrix} + \log_2 \begin{pmatrix} d \\ \mid A_i \mid \end{pmatrix} \right]$$

and the total description length of the conditional probability table is

$$\mathrm{DL}_{\mathrm{cpt}} = \sum_{i=1}^{k} \mathrm{DL}\left(A_i \mid B_i\right).$$

Thus the total description length the product approximation $\mathbf{p}_{\mathcal{S}}$ is

$$\mathrm{DL}\left(\mathbf{p}_{\mathcal{S}}\right) = \mathrm{DL}_{\mathcal{S}} + \mathrm{DL}_{\mathrm{cpt}}. \tag{B.2}$$

Appendix C

Appendix: Spanning trees with Uniform Distribution

C.1 An Algorithm due to D.J. Aldous

There are several algorithms, [61] and the references therein, for generating a spanning tree $\mathcal{T}_d$ on a set of nodes, say $\mathbf{l}$, as a sample from the uniform distribution, i.e. so that $Pr(\mathcal{T}_d) = \frac{1}{d^{d-2}}$. David Aldous has in [4] publlished the following invention.

Algorithm 4 ALDOUS ($\mathbf{l}$)

1: Fix $d \geq 2$
2: For $2 \leq i \leq d$ connect node i to the node $j_i = \min(U_i, i-1)$, where $U_2, \ldots, U_n$ are independent and uniform on $\mathbf{l} = \{1, 2, \ldots, d\}$.
3: Relabel the nodes $1, 2, \ldots, d$ as $\pi(1), \pi(2), \ldots, \pi(d)$, where π is a uniform random permutation of $1, 2, \ldots, d$.

Proposition C.1.1 The random tree $\mathcal{T}_d$ produced by the Algorithm ALDOUS is uniform, i.e.,

$$Pr(\mathcal{T}_d) = \frac{1}{d^{d-2}}. \tag{C.1}$$

∎

Appendix D

Appendix: Prior Distributions and the Razor; Stirling-Binet formula

D.1 Beta Density

With $\alpha_i > 0$ for $i = 1, 2$, we say that a random variable θ with values in $[0, 1]$ has a $\mathrm{Be}(\alpha_1, \alpha_2)$ distribution, if θ has the probability density

$$h(\theta) = \begin{cases} \frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\cdot\Gamma(\alpha_2)} \theta^{\alpha_1-1} (1-\theta)^{\alpha_2-1} & 0 \leq \theta \leq 1 \\ 0 & \text{elsewhere.} \end{cases} \tag{D.1}$$

The Beta integral is

$$\int_0^1 \theta^{\alpha_1-1}(1-\theta)^{\alpha_2-1} d\theta = \frac{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)}{\Gamma(\alpha_1+\alpha_2)}. \tag{D.2}$$

D.2 Jeffreys' Prior and the Razor

For a parametric model $\boldsymbol{X} \sim f(x|\theta)$, where $f(x|\theta)$ is a density with respect to μ and is differentiable w.r.t to $\theta \in R$, we define $I(\theta)$, *Fisher information* of x, as

$$I(\theta) = \int_{\mathcal{X}} \left(\frac{\partial \log f(x|\theta)}{\partial \theta} \right)^2 f(x|\theta)\, d\mu(x)$$

Exact conditions for existence of $I(\theta)$ can be given. Then Jeffreys' Prior is

$$\pi(\theta) := \frac{\sqrt{I(\theta)}}{\int_\Theta \sqrt{I(\theta)} d\theta} \tag{D.3}$$

assuming that the standardizing integral in the denominator exists. Otherwise the prior is improper.

Example D.2.1 Let $f(x \mid \theta) = \theta^x \cdot (1-\theta)^{1-x}$, $x = 0, 1$, $0 \leq \theta \leq 1$. Then the Jeffrey's prior $\pi(\theta)$ is the density of $\mathrm{Be}\left(\frac{1}{2}, \frac{1}{2}\right)$.

■

The razor of V. Balasubramanian [9] is then formally expressed with π from (D.3) given as

$$R_N = \int_{\Theta} \pi(\theta)\, e^{-ND(\mathbf{p}\|\mathbf{p}_\theta)} d\theta. \tag{D.4}$$

One should study the derivation of this for any further understanding.

D.3 The Binet Formula

Binet formula for Stirling approximation of Euler's Gamma function is

$$\ln \Gamma(s) \approx \frac{\ln(2\pi)}{2} + \left(s - \frac{1}{2}\right) \ln s - s + C \tag{D.5}$$

This is found in
E.T. Whittaker & G.N. Watson: *A Course in Modern Analysis (fourth ed.)*, Cambridge University Press, Cambridge 1990, pp. 248−249.

Bibliography

[1] A.V. Aho, J. Hopcroft & J.D. Ullman: *Data Structures and Algorithms.* Addison-Wesley, Reading Massachusetts, 1985.

[2] J. Aitchison: Goodness of prediction fit. *Biometrika*,62, 1975, pp. 547−181.

[3] S.M. Aji & R.J.McEliece: The generalized distributive law. *IEEE Transactions on Information Theory*, 46, 2000, pp. 325−343.

[4] D. Aldous: The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal on Discrete Mathematics*, 3, 1990, pp. 450−465

[5] D. Aldous: The continuum random tree I. *The Annals of Probability*, 19, 1991, pp. 14−28.

[6] S. Anolouva, P. Fischer, S. Pölt & H.U. Simon: Probably Almost Bayes Decisions. *Information and Computation*, 129, 1996, pp.63−71.

[7] R.R. Bahadur: *Some Limit Theorems in Statistics.* SIAM, Philadelhia, 1971.

[8] K.S. Balagani & V.V. Phoha: On the Relationship between Dependence Tree Classification Error and Bayes Error Rate. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2007, 29, pp. 1866-1868.

[9] V. Balasubramanian: Statistical inference, Occam's razor, and statistical mechanics on the space of probability distributions. *Neural computation*,9, 1997, pp. 349−368

[10] C. Beeri, R. Fagin, D. Maier & M. Yannakakis: On the desirability of acyclic database schemes. *Journal of the ACM*, 30, 1983, pp. 479−513.

[11] A. Bonin, D.Ehrich & S. Manel: Statistical analysis of amplified fragment length polymorphism data: a toolbox for molecular ecologists and evolutionists. *Molecular Ecology*, 16, 2007, pp. 3737−3758

[12] C. Borgelt: On Identifying Tree-Structured Perfect Maps. *KI 2003: Advances in Artificial Intelligence.* Lecture Notes in Computer Science, Volume 2821, Springer Berlin / Heidelberg, 2003 pp.385−395.

[13] C.J. Butz: *The relational database theory of Bayesian networks.* Ph.D.-thesis, University of Regina, 2000.

[14] O. Catoni:*Statistical learning theory and stochastic optimization: Ecole d'Eté de Probabilités de Saint-Flour XXXI-2001.* Springer Verlag, Berlin, 2004.

[15] J. Cheng & R. Greiner, J. Kelly, D. Belly & W. Liu: Learning Bayesian Networks from data: An information-theory based approach. *Artificial Intelligence* 2003, 137, pp. 43−90.

[16] C.K. Chow & C.N. Liu: Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Transactions on Information Theory,* 1968, 14, pp. 462−467.

[17] C.K. Chow & T.J. Wagner: Consistency of an estimate of tree-dependent probability distributions. *IEEE Transactions on Information Theory.* 1973, 19 , pp. 369−371.

[18] J. Corander, M. Gyllenberg & T. Koski: Learning genetic population structures using minimization of stochastic complexity. *Entropy*, 12, 2010, pp. 1102−1124

[19] T. Cormen, C.E. Leiserson & R.L. Riverson: *Introduction to Algorithms.* The MIT Press, Cambridge Mass. 1991.

[20] T.M. Cover & J.A. Thomas: *Elements of Information Theory.* J. Wiley & Sons, Inc., New York, 1991.

[21] I. Csiszár: I-Divergence Geometry of Distributions. *Annals of Probability,* 3, 1975, pp. 146−158.

[22] I. Csiszár & J. Körner: *Information Theory. Coding Theorems for Discrete Memoryless Systems.* Akadémiai Kiadó, Budapest, 1986.

[23] I. Csiszár: The method of types. *IEEE Transactions on Information Theory*,44, 1998, pp. 2505−2523.

[24] I. Csiszár & F. Matus: Information projections revisited. *IEEE Transactions on Information Theory,* 49, 2003, pp. 1474− 1490.

[25] A.P. Dawid & S.L. Lauritzen: Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics*, 21, 1993, pp. 1272−1317.

[26] S. Dasgupta: Learning polytrees. *Uncertainty in Artificial Intelligence* '99.

[27] M. Ekdahl: *Approximations of Bayes Classifiers for Statistical Learning of Clusters.* Linköping Studies in Science and Technology, Thesis No. 1230, Linköping 2006.

[28] M. Ekdahl & T. Koski: Bounds for the Loss in Probability of Correct Classification Under Model Based Approximation. *The Journal of Machine Learning Research*, 7, 2006, pp. 2449−2480.

[29] G. Fritzsche: *Theoretische Grundlagen der Nachrichtentechnik.* VEB Verlag Technik, Berlin, 1987.

[30] H.N. Gabow, Z. Galil, T. Spencer & R.E. Tarjan: Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6, 1986, pp. 109−122,

[31] J. Grim: On structural approximating multivariate discrete probability distributions. *Kybernetika* (Prague), 1984, 20, pp. 1−17.

[32] M. Gyllenberg, M., J.Carlsson, & T. Koski: Bayesian Network Classification of Binarized DNA Fingerprinting Patterns. In *Mathematical Modelling and Computing in Biology and Medicine*; Capasso, V. Ed.; Progetto Leonardo, Bologna, 2003 , pp. 60−66.

[33] M. Gyllenberg, T. Koski , & T. Lund : *BinClass: A Software Package for Classifying Binary Vectors User"s Guide.* Technical Report: TUCS-TR-411, 2001
`http://portal.acm.org/citation.cfm?id=893341`

[34] D.J. Hand, H. Mannila & P. Smyth: *Principles of Data Mining.* The MIT Press, 2001.

[35] J. Hartmanis: Application of Some Basic Inequalities for Entropy. *Information and Control*, 2, 1959, pp. 199−213.

[36] H. Heikinheimo, E. Hinkkanen, H. Mannila, T. Mielikäinen & J.K. Seppänen: Finding Low Entropy Trees from Binary Data. *KDD·07*, 2007, pp. 350−359.

[37] W. Hoeffding: Asymptotically Optimal Tests for Multinomial Distributions. *The Annals of Mathematical Statistics*, 36, 1965), pp. 369−401, also in N.I. Fisher and P.K. Sen Editors: *The Collected Works of Wassily Hoeffding*, Springer Verlag, New York, 1994, pp. 431−472.

[38] F. den Hollander: *Large Deviations.* Fields Institute Monograph, American Mathematical Society, Providence, R.I., 2000.

[39] K.U. Höffgen: Learning and robust learning of product distributions. *Proceedings of the sixth annual conference on Computational Learning Theory*, 1993, pp. 77−83.

[40] K. Huang, I. King & M.R. Lyu: Constructing a large node Chow-Liu tree based on frequent itemsets. *Proceedings of the International Conference on Neural Information Processing*, 2002.

[41] R. Jirousek: A survey of methods used in probabilistic expert systems for knowledge integration. *Knowledge-Based Systems*, 3, 1990, pp. 7−12.

[42] H.G. Kellerer: Verteilungsfunktionen mit gegebenen Marginalverteilungen. *Zeitschrift für Wahrschenlichkeitstheorie und Verwandte Gebiete*, 3, 1964, pp. 247−270.

[43] H.G. Kellerer: Indecomposable marginal problems. *Advances in probability distributions with given marginals: beyond the copulas*, 1991, Springer Verlag, Berlin, pp. 139−149.

[44] S. Kolahi & L. Libkin: An information-theoretic analysis of worst-case redundancy in database design. *ACM Transactions on Database Systems*, 35, 2010, pp. 1−32.

[45] T. Koski & J. Noble: *Bayesian Networks. An Introduction.* J. Wiley Sons, New York, London, 2009.

[46] R.E. Krichevsky & V.K. Trofimov: The performance of universal encoding. *IEEE Transactions on Information Theory*, 27, 1981, pp. 199−207.

[47] H.H. Ku & S. Kullback: Approximating Discrete Probability Distributions. *IEEE Transactions on Information Theory,* 1969, 15 , pp. 444−447.

[48] L. Kučera: *Combinatorial Algorithms.* Adam Hilger, Bristol, 1990.

[49] W. Lam & F. Bacchus: Learning Bayesian networks: an approach based on the MDL principle. *Computational Intelligence*, 10, 1994, pp. 269−293

[50] P.M. Lewis II: Approximating Probability Distributions to Reduce Storage Requirements. *Information and Control*, 2, 1959, pp. 1959.

[51] D.J. MacKay: *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2004 (reprint with corrections).

[52] F.M. Malvestuto: Existence of extensions and product extensions for discrete probability distributions. *Discrete Mathematics,* 69, 61−77, 1988.

[53] H. Mannila & K-J. Räihä: *The design of relational databases*, 1992, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA

[54] M. Meila: An accelerated Chow and Liu algorithm: fitting tree distributions to high-dimensional sparse data. *MIT AI Memo 1652, CBCL memo 169.*

[55] M. Meila-Predoviciu: *Learning with mixtures of trees*. PhD-thesis, Department of Electrical Engineering and Computer Science, MIT, 1999.

[56] D. Pavlov & H. Mannila: Beyond Independence: Probabilistic Models for Query Approximation on Binary Transaction Data. *IEEE Transactions on Knowledge and Data Engineering*, 15, 2003, pp. 1409−1421.

[57] J. Pearl: *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann: San Francisco, 1988.

[58] J. Pearl, D. Geiger & T. Verma: Conditional independence and its representations. *Kybernetika* (Prague), 25, 33−44, 1989.

[59] A. Perez: ϵ-admissible simplifications of the dependence structure of random variables. *Kybernetika* (Prague), 13, 1979, pp. 439−449.

[60] A. Perez & M. Studenỳ: Comparison of two methods for approximation of probability distributions with prescribed marginals. *Kybernetika* (Prague), 43, 2007, pp. 591−618.

[61] J.G. Propp, and D.B. Wilson: How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, 27, 1998, pp. 170−217.

[62] L.R. Rabiner: A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, 1989, pp. 257−286.

[63] G. Rebane & J. Pearl: The recovery of causal poly-trees from statistical data. *AAAI-87 Workshop on Uncertainty in AI*, Seattle, Washington, 1987.

[64] J. Rissanen: Stochastic Complexity in Learning. *Journal of Computer and System Science* 1997, 55, pp. 89-95.

[65] J. Rissanen: *Lectures on statistical modeling theory.* Helsinki Institute of Information Technology, Helsinki, 2004.
`http://www.lce.hut.fi/teaching/S-114.300/lectures.pdf`

[66] C. Sha, D. Tao & A. Zhou: Finding Dependency Trees from Binary Data. *IEEE 8th International Conference on Computer and Information Technology Workshops.* IEEE, 2008, pp. 80−85.

[67] C.E. Shannon: The lattice theory of information. *Claude Elwood Shannon Collected Papers*, Edited by N.J.A. Sloane and A.D. Wyner, IEEE Press, New York, 1993.

[68] G. Shafer: *Probabilistic Expert Systems.* SIAM, Philadelphia, 1996.

[69] R. Solomonoff: Two kinds of probabilistic induction. *The Computer Journal*, 42, 1999, pp. 256− 260.

[70] R. Solomonoff: Three kinds of probabilistic induction: Universal distributions and convergence theorems. *The Computer Journal*, 51, 2008, pp. 566−571.

[71] P. Spirtes, C.N. Glymour & R. Scheines: *Causation, prediction, and search.* The MIT Press, 2001.

[72] N. Srebro: Maximum likelihood bounded tree-width Markov networks *Artificial intelligence*,143,2003, pp. 123−138.

[73] B. Streitberg: Lancaster Interactions Revisited. *Annals of Statistics*, 18, 1990, pp. 1878−1885.

[74] M. Studenỳ: *Probabilistic Conditional Indpendence Structures*, Springer Verlag, London, 2005.

[75] J. Suzuki: Learning Bayesian belief networks based on the MDL principle: An efficient algorithm using the branch and bound technique. *IEICE Transactions on Information and Systems E Series D*, 1999, 356−367

[76] J. Suzuki: A Generalization of the Chow-Liu Algorithm and its Application to Statistical Learning. *Arxiv preprint arXiv:1002.2240*
`http://arxiv.org/pdf/1002.2240`

[77] G. Szekeres & F.E. Binet: On Borel fields over finite sets, *The Annals of Mathematical Statistics*,28, 1957, 494−498.

[78] V.Y.F. Tan, A. Anadkumar, L. Tong, A.S. Willsky: A Large-Deviation Analysis of the Maximum-Likelihood Learning of Markov Tree. *International Symposium on Information Theory, ISIT 2009*, IEEE Press 2009. London, New York, 2009.

[79] R.E. Tarjan: *Data Structures and Network Algorithms.* SIAM, Philadelphia, 1983.

[80] N. Tatti & H. Heikinheimo: Decomposable Families of Itemsets. *TKK Report in Information and Computer Science*, TKK-ICS-R1, Espoo, 2008.

[81] F.L. Thompson, B. Hoste, K. Vandenmeulebroecke and J. Swings (2001): Genomic Diversity Amongst *Vibrio* Isolates from Different Sources Determined by Fluorescent Amplified Fragment Length Polymorphism. *Systematic and Applied Microbiology*, 24, pp. 520−538.

[82] J. Vomlel: *Methods of probabilistic knowledge integration.* PhD Thesis, ČVUT v Praze (Czech Technical University in Prague), 1999.

[83] N.N. Vorobév: Consistent families of measures and their extensions. *Theory of Probability and its Applications*, 7, 1962, pp. 147−162.

[84] D. Wedelin: Efficient estimation and model selection in large graphical models. *Statistics and Computing*, 6, 1996, pp. 313−323.

[85] D.B. West: *Introduction to Graph Theory 2nd Ed.*, Prentice-Hall, 2000.

[86] J. Williamson: *Bayesian Nets and Causality*, Oxford University Press, 2005.

[87] S.K.M. Wong, C.J. Butz & Y. Xiang: A method for implementing a probabilistic model as a relational database. *Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 556−564.

[88] S.K.M. Wong: An extended relational data model for probabilistic reasoning *Journal of Intelligent Information Systems*,9, 1997, pp. 181−202.

[89] J.K. Yamanishi: A decision-theoretic extension of stochastic complexity and its applications to learning. *IEEE Transactions on Information Theory*, 44, 1998, pp. 1424−1439.