MASTER'S THESIS

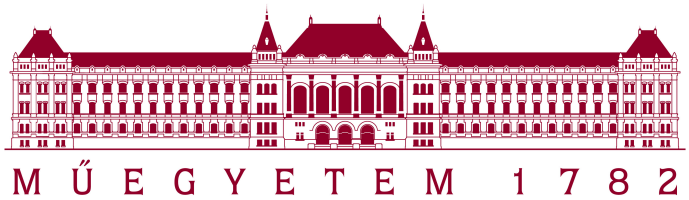
Graphical models and some related algorithms

Máté Baranyi

Supervisor:

Marianna Bolla

Doctor Habil
Department of Stochastics,
Institute of Mathematics,
Budapest University of Technology and Economics



BME
2018

Contents

Introduction **2**

1 Notation and background **4**
1.1 Graphical models in general 4
1.2 The vertex elimination process 6

2 Examples of graphical models **11**
2.1 A directed graphical model: Bayesian network 11
2.1.1 d-separation and the Bayes-ball algorithm 12
2.2 An undirected graphical model: Markov random field 14
2.3 A discrete MRF: Graphical log-linear models 19
2.3.1 Contingency tables 19
2.3.2 The log-linear model 20
2.3.3 Iterative proportional scaling 24
2.4 A continuous MRF: Gaussian graphical models 25
2.4.1 Partitioned covariance matrices and partial correlations 25
2.4.2 The model and the covariance selection 28

3 Chordality and decomposable graphical models **31**
3.1 Triangulated graphs 31
3.2 Methods to achieve chordality and to work with such graphs 36
3.2.1 Moralization 37
3.2.2 Triangulation 37
3.2.3 Junction tree construction 38
3.3 Belief propagation on junction trees 43
3.3.1 A stylized example 46
3.4 Decomposable Gaussian graphical models 52

Conclusions **54**

Introduction

In this work we will give an introduction into the theory of probabilistic graphical models (or simply just graphical models) and show some related algorithms. The purpose of graphical models is to provide a clearer, computationally more efficient way to look at multidimensional probability spaces. The aim of these graph-based representations is to give us a picture about the dependencies and conditional independences that hold between the variables.

In the first chapter we introduce the notation and the most important theoretical background which is needed to understand the rest of the work. We also introduce here the method of vertex elimination. It is crucial for the characterization of an important graph property, called chordality, in the third chapter.

In the second chapter we show examples of graphical models and try to point out connections between the different types. Two of the most commonly used models are the Bayesian networks and the Markov random fields. We start with Bayesian networks, which are widely used models in case of directed graph representations. The main properties of these were summarized in the '80s by Judea Pearl [16] and others. Then we continue with Markov random fields (sometimes Markov networks). These are so common among undirected representations that some use (incorrectly) the term undirected graphical model for these. The Markov random field was introduced as the general setting for the Ising model, also in the '80s. We will describe the log-linear models too, which are not necessary graphical models, but they have their statistical and information theoretical importance. We also give a brief introduction into multidimensional normal distributions and the Gaussian graphical models, which are built on them. We stick mainly to discrete variables, however we close the second and the third chapter with supplements about continuous models, namely the Gaussian models.

Again, in the third chapter we thoroughly investigate an important graph property, called chordality. This is equivalent to the graph being decomposable, which opens the door to many applications. In Sec. 3.1 we examine equivalences of the decomposable graph property. Since this is a desired feature of the underlying graph, in Sec. 3.2 we show commonly used methods to identify these graphs or modify a given graph in order to get a chordal representation.

At the same place we also introduce the junction tree structure, which will be used in Sec. 3.3, where we give a detailed description of the junction-tree algorithm. This is a specific tree-structure of the cliques of the graph. It applies the belief propagation, also known as sum-product message passing method on the junction tree structure of the graph. We go through every steps which are needed in the direct application of the algorithm. The goal of this algorithm is to find marginal distributions for the variables in a graphical model, possibly after absorbing some evidences. The marginal distribution of a single variable (without any previous knowledge) is simply the summation of the joint probability over all other variables, however this becomes computationally intractable very quickly. This method however makes the marginalization more effective, but it can only work if the probability distribution is a Gibbs-distribution with respect to the given graph (this will be introduced in the context of Markov random fields), and the graph representation of the probability space is triangulated (chordal), possibly after some modifications.

We show in Sec. 3.3.1 a detailed application based on the stylized example of [14]. Here we use most of the described tools while we apply the junction tree algorithm.

Summarizing, our main purpose is to thoroughly understand the theoretical background of graphical models, and by examining the decomposable models, pointing out some connections between these.

Chapter 1

Notation and background

1.1 Graphical models in general

To be able to characterize these models we will stick to some restrictions on the notations to avoid misunderstanding. In a graphical model representation of a multivariate probability space the picture looks more or less like this:

Let $G = (V, E)$ be the graph representation of the desired multivariate probability space. We work with **simple graphs**, meaning it has at most one edge for each vertex-pair and has no vertex linked to itself. There is a one-to-one compliance between the vertices (V) and the random variables, therefore we regularly say things like probability distribution over G or with respect to G. When it does not cause any confusion sometimes we use the vertex, node and variable words interchangeably. We try to represent the dependencies between the random variables with edges (E) in a way that these meet with the specifications of the chosen model.

1. The letters G, V, E will be used only for the graph representation:
   (a) $G = (V, E)$ is the graph itself;
   (b) $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices/nodes (denoted with lower case letters);
   (c) $E \subset V \times V$ is the set of edges: $(i, j) \in E$ or $i \to j$ will mean a directed edge from i to j , meanwhile $[i, j] \in E$ or $i - j$ will mean an undirected edge.
2. To every $i \in V$ it belongs a random variable X_i:
   (a) The random variables in the probability space are $X_1, X_2, \ldots, X_n$ (denoted with upper case letters). For every vertex $i \in V$, X_i is a discrete (one-dimensional) random variable or (later) continuous real-valued.

(b) For any subset $A = \{a_1, a_2, \ldots, a_m\} \subset V$, $\mathbf{X}_A$ will denote the following random vector: $\mathbf{X}_A = \mathbf{X}_{\{a_1,a_2,\ldots a_m\}} := (X_{a_1}, X_{a_2}, \ldots, X_{a_m})$

Above X_i takes its values from the finite set $\mathcal{X}_i$ (or later $\mathcal{X}_i$ can be $\mathbb{R}$), thus $\mathbf{X}_V$ (or sometimes just $\mathbf{X}$) will take them from $\mathcal{X}_V := \bigtimes_{i=1}^{n} \mathcal{X}_i$ and for $A \subset V$ respectively. A possible outcome is sometimes called a configuration or a state.

For a given graph $G = (V, E)$, the subgraph induced by a subset $A \subset V$ is as usual:

$$G(A) = (A, E(A)), \quad \textit{where} \quad E(A) = \{(x, y) \in E \quad \text{or} \quad [x, y] \in E \mid x, y \in A\} \ .$$

We shorten the name subgraph induced by a set of vertices to subgraph of vertices, which again just keeps those vertices and the edges present among them in a given graph.

Independence does not often occur in complex systems. Conditional independence however does often arise and can lead to significant representational and computational savings. This is a generalization of independence, where two pieces of a system become independent once we observe a third piece. In the theory of graphical models the concept of conditional independence plays an important role. We will walk around this concept first.

Let (X, Y, Z) be a random vector whose coordinates take values from (possible different) finite sets. We use the conventional short notations for the probability of a configuration:

$$\mathrm{p}_{\mathrm{XY}}(\cdot, \cdot) = \mathrm{Prob}\,(X = \cdot, Y = \cdot) \quad \text{and}$$

$$\mathrm{p}_{\mathrm{X|Y}}(\cdot, \cdot) = \mathrm{Prob}\,(X = \cdot \mid Y = \cdot) \ .$$

Definition 1.1. *We say that*

- *X and Y are independent if:*
  - *$X \perp\!\!\!\perp Y$ (notation);*
  - $\mathrm{p}_{\mathrm{XY}} = \mathrm{p}_{\mathrm{X}}\, \mathrm{p}_{\mathrm{Y}}$*;*
  - $\mathrm{p}_{\mathrm{X|Y}} = \mathrm{p}_{\mathrm{X}}$*.*
- *X and Y are conditionally independent given Z:*
  - *$X \perp\!\!\!\perp Y \mid Z$ (notation);*
  - $\mathrm{p}_{\mathrm{XY|Z}} = \mathrm{p}_{\mathrm{X|Z}}\, \mathrm{p}_{\mathrm{Y|Z}}$*;*
  - $\mathrm{p}_{\mathrm{X|YZ}} = \mathrm{p}_{\mathrm{X|Z}}$*.*

In the above definitions and later on as well we mean by the product of functions the point-wise product of them over the product domain:

$$\mathrm{p_X}\,\mathrm{p_Y}(\cdot,\cdot) := \mathrm{p_X}(\cdot)\,\mathrm{p_Y}(\cdot)\ ,$$

and by equality of such functions we mean point-wise equality.

Marginalization is a key concept in multidimensional probability theory. Since we will confine ourselves mostly to discrete variables in the sequel, thus for a subset $A \subset V$ of the examined variables, the A-marginal of a function $f_V : \mathcal{X}_V \to \mathbb{R}$, with other words its marginalization over $V \setminus A$, is the function $f_A : \mathcal{X}_A \to \mathbb{R}$ below:

$$f_A(\mathbf{x}_A) := \sum_{\substack{\mathbf{y}_V \in \mathcal{X}_V: \\ \mathbf{y}_A = \mathbf{x}_A}} f_V(\mathbf{y}_V) = \sum_{\mathbf{x}_{V\setminus A} \in \mathcal{X}_{V\setminus A}} f_X(\mathbf{x}_A, \mathbf{x}_{V\setminus A}), \quad \forall \mathbf{x}_A \in \mathcal{X}_A\ .$$

Note that f is not necessary a probability mass or probability density function. Briefly, we will use the notation:

$$f_A(\mathbf{x}_A) := \left(\sum_{V\setminus A} f_V\right)(\mathbf{x}_A)\ ,$$

or shortly $f_A := \sum_{V\setminus A} f_V$.

1.2 The vertex elimination process

Algorithms operating on graphical models usually require a special structure from the underlying graph, the distribution and sometimes an ordering of its vertices, variables.

In this section we will focus on vertex elimination. This procedure is crucial in the introduction of perfect elimination orderings, which are necessary in the characterization of chordal graphs and junction trees, which appears in the forthcoming chapters. This section is based on articles of Rose, Tarjan, Yannakakis and Lueker, [20, 17]. For us the undirected version of vertex elimination is more important, but we will have some remarks about the directed case. These will appear separated in boxes, to avoid confusion.

Definition 1.2. *For a graph $G = (V, E)$ with n vertices, an $\alpha : V \to \{1, 2, \ldots, n\}$ bijection is called an **ordering** of V and $G_\alpha = (V, E, \alpha)$ is called an **ordered graph**.*

As a more convenient notation for any pair of vertices $i,\ j$, $i <_\alpha j$ will mean that $\alpha(i) < \alpha(j)$ and similarly $min(i,\ j)$ will mean $\alpha^{-1}(min(\alpha(i),\ \alpha(j)))$.

Definition 1.3. *In an ordered graph $G_\alpha = (V, E, \alpha)$ for a given $v \in V$, the set of* ***monotonely adjacent*** *vertices are:*

$$\mathrm{MAdj}(v) = \mathrm{Adj}(v) \cap \{x \in V |\ v <_\alpha x\}\ ,$$

where $\mathrm{Adj}(v)$ *is the neighbors of* v*:*

$$\mathrm{Adj}(v) = \{j \in V \mid [v, j] \in E\}\ .$$

Recall that $(v, x) \in E$ means $v \to x$, a directed edge. Whereas $[v, x] \in E$ means $v - x$, an undirected edge. As usual in graph theory we can think of an undirected edge as an edge directed in both ways ($v \leftrightarrow x$).

Definition 1.4. *In a graph G for a vertex v, the* ***deficiency*** *of v is the following subset of $V \times V$:*

$$\begin{aligned} D(v) &= \{(i,j) |\ \{i,j\} \subset \mathrm{Adj}(v),\ j \notin \mathrm{Adj}(i),\ i \neq j\} \\ &= \{(i,j) |\ v - i,\ v - j,\ i \neq j,\ i \neq j\}. \end{aligned}$$

In the directed case it becomes:

$$\begin{aligned} D(v) &= \{(i,j) |\ v \in \mathrm{Adj}(i),\ j \in \mathrm{Adj}(v),\ j \notin \mathrm{Adj}(i),\ i \neq j\} \\ &= \{(i,j) |\ i \to v,\ v \to j,\ i \nrightarrow j,\ i \neq j\}. \end{aligned}$$

Similarly we can define the monotone deficiency:

Definition 1.5. *In an ordered graph G_α for a vertex v, the* ***monotone deficiency*** *of v is the following subset of $V \times V$:*

$$\begin{aligned} MD(v) &= \{(i,j) |\ \{i,j\} \subset \mathrm{MAdj}(v),\ j \notin \mathrm{Adj}(i),\ i \neq j\} \\ &= \{(i,j) |\ v - i,\ v - j,\ i \neq j,\ i \neq j,\ v <_\alpha min(i,\ j)\} \end{aligned}$$

In the directed case it becomes:

$$\begin{aligned} MD(v) &= \{(i,j) |\ v \in \mathrm{Adj}(i),\ j \in \mathrm{Adj}(v),\ j \notin \mathrm{Adj}(i),\ i \neq j,\ v <_\alpha min(i,\ j)\} \\ &= \{(i,j) |\ i \to v,\ v \to j,\ i \nrightarrow j,\ i \neq j,\ v <_\alpha min(i,\ j)\}. \end{aligned}$$

Definition 1.6. *The* v***-elimination graph*** *of* G *is defined as:*

$$G_v = (V \setminus \{v\}, E(V \setminus \{v\}) \cup D(v))$$

Note that the v-elimination graph is not equivalent to the graph that we got if we just simply delete v. The v-elimination ensures that we don't lose any paths. In other words it means that we add edges such that the vertices in $\mathrm{Adj}(v)$ become adjacent.

Definition 1.7. *For an ordered graph* G_α*, the* ***elimination process*** *is:*

$$P(G_\alpha) := [G_0, G_1, \ldots, G_{n-1}],$$

where $G_0 = G$ *and* $G_i = (G_{i-1})_{\alpha^{-1}(i)}$ *for* $i > 0$.

Definition 1.8. *For an ordered graph* G_α*, we define its* ***fill-in*** *as:*

$$F(G_\alpha) = \cup_{i=1}^{n-1} D_{i-1}(\alpha^{-1}(i)),$$

where $D_{i-1}(\alpha^{-1}(i))$ *is the deficiency of* $\alpha^{-1}(i)$ *in* G_{i-1}.

Definition 1.9. *The* ***elimination graph*** *of* G *by ordering* α *is defined as:*

$$G^\star_\alpha = (V, E \cup F(G_\alpha)).$$

Definition 1.10. *For a given graph* G*,* α *is a* ***perfect elimination ordering*** *(or zero fill-in ordering) if and only if* $F(G_\alpha) = \emptyset$*. A graph with such an ordering is called a* ***perfect elimination graph****.*

$F(G_\alpha) = \emptyset$ means that $D_{i-1}(\alpha^{-1}(i)) = \emptyset$ for $i \in \{1, \ldots, n-1\}$. This means that along the $P(G_\alpha)$ elimination process:

$$G_i = G(V \setminus \cup_{l=1}^{i} \{\alpha^{-1}(l)\})$$

Definition 1.11. *For a given graph* G*, an* α *elimination ordering is:*

- ***minimal*** *if* $\nexists$ β *ordering s.t.* $F(G_\beta) \subset F(G_\alpha)$*,*
- ***minimum*** *if* $\nexists$ β *ordering s.t.* $|F(G_\beta)| < |F(G_\alpha)|$.

Any elimination graph $G^\star_\alpha$ is a perfect elimination graph, since α is a perfect ordering of this graph. Any perfect ordering of a graph is minimum, and any minimum ordering is minimal. If a graph is a perfect elimination graph, any minimal ordering is perfect.

Rose also defines monotone transitivity as follows:

Definition 1.12. *An ordered graph $G_\alpha = (V, E, \alpha)$ is **monotone transitive** if for all $x \in V$:*

$$i \in \mathrm{MAdj}(x),\ j \in \mathrm{MAdj}(x) \quad \Rightarrow \quad i \in \mathrm{Adj}(j).$$

This definition is symmetric, it can not be true in a graph which contains only one-way edges.

Lemma 1.1. *For an ordered graph G_α, the followings are equivalent:*

- *G_α is monotone transitive,*
- *$\forall\ v \in V:\ MD(v) = \emptyset$,*
- *$P(G_\alpha)$ is a perfect elimination process.*

Lemma 1.2. *An α ordering is perfect if:*

$$i - j,\ i - k,\ i <_\alpha min(j,\ k)\ \Rightarrow\ j = k\ or\ j - k\ .$$

In other words, in a monotone transitive graph, vertex elimination adds no edges. Monotone transitive graphs can be characterized by their cycle structure and their minimal separators.

Definition 1.13. *A graph G is **triangulated** if for every cycle C of length $k > 3$ there is an edge of G joining two nonconsecutive vertices of C. These edges are called chords of the cycle.*

Definition 1.14. *A **separator** of a graph $G = (V, E)$ is a subset $S \subset V$ such that the subgraph $G(V \setminus S)$ consists of two or more connected components. A **minimal separator** is a separator for which any subset of it is not a separator. Similarly, given $a, b \in V$ with $a \neq b$ an a, b-separator is a separator such that a and b are in distinct components after the separation; and a minimal a, b-separator is a minimal one of these.*

Theorem 1.1. *For a graph $G = (V, E)$ the followings are equivalent:*

- *$\exists\ \alpha$ ordering such that G_α is monotone transitive,*
- *G is triangulated,*
- *for every non-adjacent $x, y \in V$ pair, the minimal separator is a complete subgraph.*

Finding out if a graph is perfect elimination graph is similar to the problem of testing a directed graph for transitivity.

Definition 1.15. *The transitive closure of a graph $G = (V, E)$ is the graph $G^+ = (V, E^+)$, where $(i \to j) \in E^+$ if and only if $i \neq j$ and there is a path from i to j in G.*

Definition 1.16. *A G graph is transitive if $G = G^+$.*

Lemma 1.3. *For an ordered directed graph $G = (V, E, \alpha)$, $(i \to j) \in E \cup F(G_\alpha)$ if and only if there exists a path $(i = v_0, v_1, \ldots, v_k = j)$ in G such that:*

$$v_l <_\alpha min(i,\ j) \quad for \quad \forall\ 0 < l < k.$$

Lemma 1.4. *Let $G = (V, E, \alpha)$ be an ordered graph, where α is a perfect elimination ordering. For arbitrary $x \in V$ let $G' = (V, E \cup D(x))$. Then α is a perfect elimination ordering of G' as well.*

Corollary 1.1. *If $G = (V, E)$ is a perfect elimination graph, then for any $x \in V$ $G_x = (V \setminus \{x\}, E(V \setminus \{x\}) \cup D(x))$ is also a perfect elimination graph.*

Corollary 1.2. *If $G = (V, E)$ is a perfect elimination graph and $\exists\ x \in V$ with $D(x) = \emptyset$, then there is a perfect elimination ordering α such that $\alpha^{-1}(1) = x$.*

Similarly as before, F is a minimal fill-in if $\nexists\ H \subset F$ such that H is a fill-in.

Lemma 1.5. *Let $G = (V, E)$ be a perfect elimination graph and let $F \neq \emptyset$ be a fill-in for it. (Note that $G' = (V, E \cup F)$ is also a perfect elimination graph.) Then $\exists f \in F$ such that $F \setminus \{f\}$ is also a fill-in.*

From this lemma the next theorem follows:

Theorem 1.2. *Let $G = (V, E, \alpha)$ be an ordered graph. Then α is a minimal elimination ordering if and only if for all $f \in F(G_\alpha)$, the graph $G^\star_\alpha - f := (V, E \cup (F(G_\alpha) \setminus \{f\}))$ is not a perfect elimination graph, $F(G_\alpha) \setminus \{f\}$ is not a fill-in.*

Theorem 1.3. *A graph G is triangulated if and only if it has a perfect elimination (or zero fill-in) ordering.*

Chandran et al. [2] model the set of all perfect elimination orderings of a chordal graph as the basic words of an antimatroid and use this connection as part of an algorithm for efficiently listing all perfect elimination orderings of a given chordal graph.

Chapter 2

Examples of graphical models

2.1 A directed graphical model: Bayesian network

We speak about a Bayesian network when the graph representation of our multivariate probability space is a directed acyclic graph (DAG).

Definition 2.1. *A simple directed graph $G = (V, E)$ is called a **directed acyclic graph** if there is an ordering $\sigma : V \to \{1, 2, \ldots, n\}$ of the vertices such that for all $(i, j) \in E, \sigma(i) < \sigma(j)$, meaning there can not be a directed cycle in G.*

Note that for a DAG such an ordering is a topological ordering. Now let $(X_1, X_2, \ldots, X_n)$ be a random vector whose coordinates take values from (possible different) finite sets. Then with the well-known chain-rule, the joint probability can be written as:

$$\mathrm{p}_{\mathrm{X}_1\mathrm{X}_2\ldots\mathrm{X}_n} = \mathrm{p}_{\mathrm{X}_1}\, \mathrm{p}_{\mathrm{X}_2|\mathrm{X}_1}\, \mathrm{p}_{\mathrm{X}_3|\mathrm{X}_1\mathrm{X}_2} \cdots \mathrm{p}_{\mathrm{X}_n|\mathrm{X}_1\ldots\mathrm{X}_{n\text{-}1}} \ .$$

Some terms in the above equation become simpler if we know or suppose some conditional independences between certain variables. After the possible simplifications we can build as DAG as follows, from $i = 1$ to n:

- assign a vertex to every variable: $X_i \to v_i$,
- allow an edge $v_i \to v_j$ if there is a factor of the form $\mathrm{p}_{\mathrm{X}_j|\ldots\mathrm{X}_i\ldots}$.

The result of this procedure is always a DAG, however the joint distribution does not determine the graph uniquely, it depends on the ordering of the variables, and if the distribution is not strictly positive, then the simplifications are not unique either.

Definition 2.2. *If we already have a DAG on n vertices, then we say that the* $\mathrm{p}_{\mathrm{X}_1\mathrm{X}_2\ldots\mathrm{X}_n}$ *fulfills the* ***directed factorization*** *property if:*

$$\mathrm{p}_{\mathrm{X}_1\mathrm{X}_2\ldots\mathrm{X}_n} = \prod_{i=1}^{n} \mathrm{p}_{\mathrm{X}_i|\mathrm{X}_1\ldots\mathrm{X}_{i-1}} = \prod_{i=1}^{n} \mathrm{p}_{\mathrm{X}_i|\mathrm{X}_{\mathrm{Par}(i)}} \ ,$$

where $\mathrm{Par}(i)$ *means the parents of vertex i in the DAG:*

$$\mathrm{Par}(i) = \{j \in V \mid (j,i) \in E\}$$

and the ordering of the variables is a topological ordering.

The edges in a BN represent casual dependencies between the variables. Nodes that are not connected represent variables that are conditionally independent of each other given the parents of the higher numbered vertex. This property is called the **Directed Local Markov property** of the distribution of the random vector $\mathbf{X} = (X_1, \ldots, X_n)$. More precisely let

$$\mathrm{Ant}(i) = \{i-1, \ldots, 1\} \setminus \mathrm{Par}(i)$$

denote the set of **anteriors** of i (the set of its non-descendants except its parents). Then the property means that

$$X_i \perp\!\!\!\perp \mathbf{X}_{\mathrm{Ant}(i)} | \mathbf{X}_{\mathrm{Par}(i)} \quad \text{holds} \quad \forall\, i = 1, \ldots, n \ . \tag{2.1}$$

I.e. X_i (future) and $\mathbf{X}_{\mathrm{Ant}(i)}$ (past) are independent conditioned on $X_{\mathrm{Par}(i)}$ (present). This generalizes the fundamental property of Markov-chains, in which case G is just a directed path.

2.1.1 d-separation and the Bayes-ball algorithm

Bayesian networks encode the conditional independence properties of the probability space. We can determine if a conditional independence holds in a BN by using a graph separation criterion called **d-separation** (see [16] for more details), which stands for direction-dependent separation.

In a given directed acyclic graph G, two vertices are d-separated by the vertex-subset S if there is no **active path** between them. The active paths are those which are not **blocked** by S. Note that this blocking is not identical to separation. The formal definition of active paths and blocking is somewhat complex. In the following definitions, by path in a DAG we mean, after forgetting the edge directions, an undirected path. Sometimes these are called chains.

According to [13], in a DAG, a path from a to b is blocked by S if the path contains a vertex c such that:

- either $c \in S$, and arrows do not meet (not a sink) at c;
- or $c \notin S$, and arrows do meet (sink) at c, or c has no descendants in S.

Descendants of a vertex are those vertices which are reachable from the given vertex on a directed path. However, an alternative definition is given in [25]. In a DAG, a path from a to b is active given S if for every vertex c on the path:

- $c \in S \cup \mathrm{Par}(S)$ if arrows meet (sink) at c;
- $c \in V \setminus S \setminus \{a, b\}$ if it is a transition vertex.

For the terms sink and transition see Fig. 2.1. The Bayes-ball algorithm gives a nice graphical interpretation.

In [7] it is shown that for every DAG G there exists (can be constructed) a probability distribution $\mathrm{p_V}$ which embodies all the conditional independences displayed in G. Also there is no stronger criterion than d-separation, to check the conditional independences of a BN:

$$\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B \mid \mathbf{X}_S \quad \Leftrightarrow \quad A \text{ and } B \text{ are d-separated by } S \ .$$

If the left hand side of the above term is fulfilled for every partitioning of the vertices, then it is called **Directed Global Markov** property.

Bayes-ball algorithm

Again, to see if $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B \mid \mathbf{X}_C$ in a BN for some disjoint subsets $A, B, C \subset V$, we need to check if every variable in A is d-separated from every vertices in B by the vertices of C. In other words, we have to check $X_a \perp\!\!\!\perp X_b \mid \mathbf{X}_C$ for all $a \in A$ and $b \in B$.

Graphically if we condition on the vertices of C, we shade them in (see the blue nodes on Fig. 2.2). This is the starting step. Since we have to check paths, we have to consider how the possible triples of vertices may occur. For this see Fig. 2.1, the names are from [26].

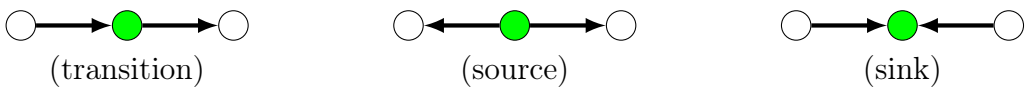


Figure 2.1: The three types of triples

The Bayes-ball algorithm works as follows: we shade all nodes in C, place balls at each node in A (or B), let them bounce around according to the ten rules below,

and then check if any of the balls reach any of the nodes in B (or A). If there is such a ball we reject the conditional independence, otherwise accept it.

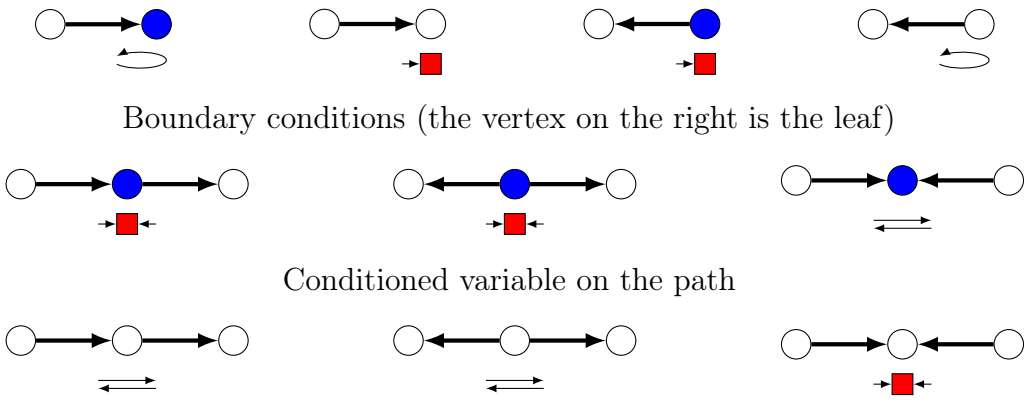


Figure 2.2: The 10 rules of Bayes-ball

Note that balls in this setup can travel opposite to the edge directions. The edge directions only indicate which rule a ball should follow during its journey.

On Fig. 2.2 the boundary conditions show what happens when a ball reaches a leaf: it can bounce back (turn arrow) or stuck there (red rectangle). The other conditions show what happens when a ball tries to pass through a vertex. The red rectangles mean that a ball is stucked there, can not go further, and the two arrows mean that a ball can pass through.

2.2 An undirected graphical model: Markov random field

If the random variables in the probability space satisfy one of the Markov properties with respect to some undirected graph then it is called a Markov random field. A BN is always a DAG, whereas an MRF is undirected and possibly cyclical. There are different type of Markov properties which can be satisfied.

Definition 2.3. *The Global Markov property* ***(GM)*** *of a joint distribution with respect to an undirected graph G is defined as follows:*

$$X_A \perp\!\!\!\perp X_B \mid X_S \tag{2.2}$$

holds for any vertex-subset S separating disjoint vertex-subsets A and B.

Definition 2.4. *The Local Markov property* ***(LM)*** *of a joint distribution with re-*

spect to an undirected graph G is defined as

$$X_i \perp\!\!\!\perp X_{V\setminus \mathrm{Cl}(i)} \mid X_{\mathrm{Adj}(i)}, \qquad \forall\, i \in V\ , \tag{2.3}$$

where $\mathrm{Cl}(i) = \mathrm{Adj}(i) \cup \{i\}$.

Definition 2.5. *The Pairwise Markov property* ***(PM)*** *of a joint distribution with respect to an undirected graph G holds if*

$$X_i \perp\!\!\!\perp X_j \mid X_{V\setminus\{i,j\}}, \qquad \forall\, i \neq j\ (\in V)\ . \tag{2.4}$$

It is easy to see that:

$$(GM) \Rightarrow (LM) \Rightarrow (PM)\ .$$

So the Global Markov property is stronger than the Local Markov property, which in turn is stronger than the Pairwise one.

These properties are hard to check by definition, but luckily we have a sufficient and necessary condition for positive probability distributions given by Hammersly and Clifford. For this we need an extra property.

Definition 2.6. *A probability distribution* $\mathrm{p_V}$ *is a* ***Gibbs distribution with respect to an undirected graph*** *G if* $\mathrm{p_V}$ *factorizes over the cliques (maximal complete subgraphs) of G, more precisely:*

$$\mathrm{p_{X_1 \ldots X_n}} = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C\ , \quad \textit{where:} \tag{2.5}$$

- $\mathcal{C} = \{C \subset V\ :\ C \textit{ is a clique of } G\}$,
- ψ_C *is a positive (real-valued) function for all $C \in \mathcal{C}$ over the possible states of the variables in C,*
- $Z = \sum_{X_1} \sum_{X_2} \cdots \sum_{X_n} \prod_{C\in\mathcal{C}} \psi_C$ *is a normalizing constant, which is a marginalization of the above product over all possible states (configurations) of the n variables.*

In some literature, e.g. [13], this property is called undirected factorization property. The connection with Def. 2.2 will have importance later, for example in Sec. 3.2.1.

In graph theory the cliques are often the complete subsets of a graph. However in statistics (and that is the convention we will follow) the cliques are usually the maximal complete subsets. We will use the above notation $\mathcal{C}$ for the set of cliques over a graph in the sequel.

The ψ_C functions are sometimes referred to as factor potentials (if the domain is not a clique for example) or clique potentials. Sometimes they are defined as they already include Z. Note that a conflicting terminology is in use: the word potential is often applied to the logarithm of ψ_C. It happens because in statistical mechanics, $\log(\psi_C)$ has a direct interpretation as the potential energy of a configuration over C. The normalizing constant is sometimes called partition function.

It is also worth noting that these functions can be multiplied together or split up in different ways. Therefore without loss of generality the above definition can be restated to include every complete subsets, not just the maximal ones.

Theorem 2.1 (Hammersly–Clifford theorem)**.** *A strictly positive probability distribution with respect to an undirected graph G satisfies the PM property if and only if it is a Gibbs distribution with respect to this G.*

The relationship between Markov random fields and Gibbs distributions was initiated by R. Dobrushin and F. Spitzer in the context of statistical mechanics. The theorem is named after J. Hammersley and P. Clifford. They proved the equivalence in [8] in 1971. Their proof is rather complex, here we show an easier one.

Proof. For the backward direction take an arbitrary i vertex and see the conditional probability of it given its neighbors:

$$
\begin{aligned}
\mathrm{p}_{\mathrm{X_i|X_{Adj(i)}}} &= \frac{\mathrm{p}_{\mathrm{X_{Cl(i)}}}}{\mathrm{p}_{\mathrm{X_{Adj(i)}}}} = \frac{\sum\limits_{V\setminus \mathrm{Cl}(i)} \prod\limits_{C\in\mathcal{C}} \psi_C}{\sum\limits_i \sum\limits_{V\setminus \mathrm{Cl}(i)} \prod\limits_{C\in\mathcal{C}} \psi_C} = \\
&= \frac{\sum\limits_{V\setminus \mathrm{Cl}(i)} \prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C \prod\limits_{\substack{C\in\mathcal{C}\\ i\notin C}} \psi_C}{\sum\limits_i \sum\limits_{V\setminus \mathrm{Cl}(i)} \prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C \prod\limits_{\substack{C\in\mathcal{C}\\ i\notin C}} \psi_C} = \frac{\prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C \sum\limits_{V\setminus \mathrm{Cl}(i)} \prod\limits_{\substack{C\in\mathcal{C}\\ i\notin C}} \psi_C}{\sum\limits_i \prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C \sum\limits_{V\setminus \mathrm{Cl}(i)} \prod\limits_{\substack{C\in\mathcal{C}\\ i\notin C}} \psi_C} = \\
&= \frac{\prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C}{\sum\limits_i \prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C} = \frac{\prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C}{\sum\limits_i \prod\limits_{\substack{C\in\mathcal{C}\\ i\in C}} \psi_C} \cdot \frac{\prod\limits_{\substack{C\in\mathcal{C}\\ i\notin C}} \psi_C}{\prod\limits_{\substack{C\in\mathcal{C}\\ i\notin C}} \psi_C} = \\
&= \frac{\prod\limits_{C\in\mathcal{C}} \psi_C}{\sum\limits_i \prod\limits_{C\in\mathcal{C}} \psi_C} = \frac{\mathrm{p}_{\mathrm{X_V}}}{\mathrm{p}_{\mathrm{X_{V\setminus\{i\}}}}} = \mathrm{p}_{\mathrm{X_i|X_{V\setminus\{i\}}}} \quad .
\end{aligned}
$$

This shows that the LM property is fulfilled, thus PM also.

For the forward direction let's take the following construction:

$$f_S(\cdot) := \prod_{A \subset S} \left(\mathrm{p}_{\mathrm{X_A}}(\cdot)\right)^{(-1)^{|S|-|A|}}$$

The domain of f_S is of course the possible configurations of the random vector X_S. The exponent is 1 if the difference between the sizes of S and A is even and -1 if odd. We have to show that:

1. $\mathrm{p_V}(\cdot) = \prod_{S \subset V} f_S(\cdot)$ for all configurations;
2. $f_S(\cdot) = 1$ if S is not a complete subgraph of G.

Let's see the first point:

$$\begin{aligned}
\prod_{S \subset V} f_S &= \prod_{S \subset V} \prod_{A \subset S} \mathrm{p_{X_A}}^{(-1)^{|S|-|A|}} \\
&= \prod_{A \subset V} \prod_{k=0}^{|V|-|A|} \mathrm{p_{X_A}}^{\binom{|V|-|A|}{k}(-1)^k} \\
&= \prod_{A \subset V} \mathrm{p_{X_A}}^{\sum_{k=0}^{|V|-|A|}\binom{|V|-|A|}{k}(-1)^k} = \mathrm{p_V} \ .
\end{aligned}$$

The first transformation comes from the fact that $A \subset S \subset V$ and counting the possible subsets (S's) in which the A can be in, separating them by the size ($|S|-|A|$). More precisely, the running variable k counts the number of vertices we should add to A to reach the size of an S, and the binomial coefficient shows the number of such subsets (S's). It is a well known trick, actually a consequence of binomial theorem, that:

$$\sum_{k=0}^{n} \binom{n}{k} (-1)^k = (1-1)^n = 0 \quad \forall\, n > 0 \ .$$

Using this, the exponent becomes zero for all $A \subset V$ except V itself.

For the second point consider an arbitrary $S \subset V$ which is not a complete subgraph. In this case we have $a, b \in S$ which are not connected to each other. We see the following transformations:

$$\begin{aligned}
f_S &= \prod_{A \subset S} \mathrm{p_{X_A}}^{(-1)^{|S|-|A|}} \\
&= \prod_{T \subset S \setminus \{a,b\}} \mathrm{p_{X_T}}^{(-1)^{|S|-|T|}} \mathrm{p_{X_{T\cup\{a\}}}}^{(-1)^{|S|-|T|-1}} \mathrm{p_{X_{T\cup\{b\}}}}^{(-1)^{|S|-|T|-1}} \mathrm{p_{X_{T\cup\{a,b\}}}}^{(-1)^{|S|-|T|-2}} \\
&= \prod_{T \subset S \setminus \{a,b\}} \left[\frac{\mathrm{p_{X_T}}\, \mathrm{p_{X_{T\cup\{a,b\}}}}}{\mathrm{p_{X_{T\cup\{a\}}}}\, \mathrm{p_{X_{T\cup\{b\}}}}} \right]^{(-1)^{|S|-|T|}}
\end{aligned}$$

Now we use the following formula:

$$\frac{\mathrm{p}_{\mathrm{X_T}}(\cdot)}{\mathrm{p}_{\mathrm{X_{T\cup\{a\}}}}(\cdot)} = \frac{\sum_a \mathrm{p}_{\mathrm{X_a|X_T}}(\cdot)\,\mathrm{p}_{\mathrm{X_T}}(\cdot)}{\mathrm{p}_{\mathrm{X_a|X_T}}(\cdot)\,\mathrm{p}_{\mathrm{X_T}}(\cdot)} = \frac{\sum_a \mathrm{p}_{\mathrm{X_a|X_T}}(\cdot)\,\mathrm{p}_{\mathrm{X_{T\cup\{b\}}}}(\cdot)}{\mathrm{p}_{\mathrm{X_a|X_T}}(\cdot)\,\mathrm{p}_{\mathrm{X_{T\cup\{b\}}}}(\cdot)}$$
$$= \frac{\sum_a \mathrm{p}_{\mathrm{X_a|X_{T\cup\{b\}}}}(\cdot)\,\mathrm{p}_{\mathrm{X_{T\cup\{b\}}}}(\cdot)}{\mathrm{p}_{\mathrm{X_a|X_{T\cup\{b\}}}}(\cdot)\,\mathrm{p}_{\mathrm{X_{T\cup\{b\}}}}(\cdot)} = \frac{\mathrm{p}_{\mathrm{X_{T\cup\{b\}}}}(\cdot)}{\mathrm{p}_{\mathrm{X_{T\cup\{a,b\}}}}(\cdot)}$$

For the first step we used the Bayes-theorem. In the second step we could simplify the the ratio by $\mathrm{p}_{\mathrm{X_T}}$since it is independent of the running variable a, but instead we just switch to $\mathrm{p}_{\mathrm{X_{T\cup\{b\}}}}$. The next identity is true because a and b are conditionally independent given T (using PM property). By the above formula we can see that every terms in the product is 1, thus the second point is proved. Therefore we have a factorization over the complete subsets, which can be easily grouped to have a factorization over the cliques. □

At this point we have that under the condition of Thm. 2.1:

$$(GM) \Rightarrow (LM) \Rightarrow (PM) \Leftrightarrow \text{Gibbs w.r.t } G\ .$$

Actually it can be shown that any Gibbs distribution with respect to a graph G fulfills the GM property. Therefore under the condition of Thm. 2.1 the four condition becomes equivalent:

$$(GM) \Leftrightarrow (LM) \Leftrightarrow (PM) \Leftrightarrow \text{Gibbs w.r.t } G\ .$$

In general the potentials do not have a probabilistic interpretation, but we can say something like: values (configurations) with higher potential are more probable.

If we don't have a graph representation yet, but we know that our joint distribution factorizes over some subsets of the variables:

$$\mathrm{p_V} := \mathrm{p}_{\mathrm{X_1 \ldots X_n}} = \frac{1}{Z} \prod_{A \subset \{X_1, \ldots X_n\}} \psi_A\ , \tag{2.6}$$

then we can easily create a graph G by assigning each variable of the probability space to a vertex (as usual) and to every available ψ_A we place a clique over the vertices assigned to the variables in A. By this method $\mathrm{p_V}$ will be trivially a Gibbs distribution over the graph G created.

2.3 A discrete MRF: Graphical log-linear models

2.3.1 Contingency tables

The observations for the components of the random vector $\mathbf{X}_V = (X_1, \ldots, X_n)$ can be collected and stored in a so-called **contingency table** structure. The entries are coming from the sample space (state space) $\mathcal{X}_V = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ as before. These possible n-tuples $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}$ are called **cells**. Altogether there are $\prod_{i=1}^{n} |\mathcal{X}|$ cells. Under contingency table we understand the cells and cell counts, $\mathrm{N}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$, where the non-negative integer $\mathrm{N}(\mathbf{x})$ is the number of observations for the random vector $\mathbf{X} = (X_1, \ldots, X_n)$ that fall in the cell $\mathbf{x}$ out of the total m observations. In other words, m is the sample size and $m = \sum_{\mathbf{x}\in\mathcal{X}} \mathrm{N}(\mathbf{x})$. When m is kept fixed, the joint distribution of a cell count, $N(\mathbf{x})$ as random variable, is multinomial with parameters m and $\mathrm{p_V}(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$:

$$\mathrm{Prob}\left(\mathrm{N}(\mathbf{x}) = c_{\mathbf{x}} \ , \quad \mathbf{x} \in \mathcal{X}_V\right) = \frac{m!}{\prod_{\mathbf{x}\in\mathcal{X}_V} c_{\mathbf{x}}!} \prod_{\mathbf{x}\in\mathcal{X}_V} \mathrm{p_V}(\mathbf{x})^{c_{\mathbf{x}}} \ . \tag{2.7}$$

In the most simple **saturated model**, the parameters are only constrained by the sampling procedure. Under multinomial sampling the ML-estimate of the parameters is obtained by equating the count $\mathrm{N}(\mathbf{x})$ to the multinomial expectation $m \cdot \mathrm{p_V}(\mathbf{x})$, hence $\hat{\mathrm{p_V}}(\mathbf{x}) = \frac{\mathrm{N}(\mathbf{x})}{m}$ for all $\mathbf{x} \in \mathcal{X}_V$, which is the empirical distribution.

With some restrictions on the marginal distributions we can define more special models. We can define marginals for the contingency table similarly as for the subset of variables earlier. The A-marginal of the contingency table corresponding to a given subset of the variables $\mathbf{X}_A = \{X_i : \ i \in A\}$, with $A \subset V$, is given by the marginal cell counts as:

$$\mathrm{N}(\mathbf{x}_A) = \sum_{\substack{\mathbf{x}'\in\mathcal{X}_V: \\ \mathbf{x}'_A = \mathbf{x}_A}} \mathrm{N}(\mathbf{x}') = \sum_{\mathbf{x}_{V\setminus A}\in\mathcal{X}_{V\setminus A}} \mathrm{N}(\mathbf{x}_A, \mathbf{x}_{V\setminus A}) \qquad \forall\, \mathbf{x}_A \in \mathcal{X}_A = \bigtimes_{i\in A} \mathcal{X}_i \ .$$

Therefore the A-marginal counts form a $|A|$-dimensional contingency table of $\prod_{i\in A} |\mathcal{X}_i|$ cells, and there are $\binom{n}{|A|}$ possible $|A|$-dimensional marginals ($|A| = 1, \ldots, n$). Likewise, the A-marginal distribution of the $\{\mathrm{p_V}(\mathbf{x}) : \ \mathbf{x} \in \mathcal{X}_V\}$ distribution is defined as earlier:

$$\mathrm{p_A}(\mathbf{x}_A) = \sum_{\substack{\mathbf{x}'\in\mathcal{X}_V: \\ \mathbf{x}'_A = \mathbf{x}_A}} \mathrm{p_V}(\mathbf{x}') = \sum_{\mathbf{x}_{V\setminus A}\in\mathcal{X}_{V\setminus A}} \mathrm{p_V}(\mathbf{x}_A, \mathbf{x}_{V\setminus A}) \qquad \forall\, \mathbf{x}_A \in \mathcal{X}_A \ .$$

2.3.2 The log-linear model

Given a set $\Gamma := \{A : A \subset V\}$ called **generating class**, we can define the **log-linear model** as follows:

$$\ln \mathrm{p_V}(\mathbf{x}) = f_0 + \sum_{A \in \Gamma} f_A(\mathbf{x}_A) \ , \tag{2.8}$$

where the individual terms represent interactions ($f_A : \mathcal{X}_A \to \mathbb{R}$ functions) corresponding to $A \in \Gamma$. f_A depends on $\mathbf{x}$ only through it's A-marginal $\mathbf{x}_A$, and the constant term f_0 corresponds to $\emptyset \in \Gamma$ (it also fits into the forthcoming hierarchical structure of Γ). This is in accord with the potential representation of the previous section, see Eq. 2.6.

Hierarchical log-linear models

We can consider the **hierarchical** log-linear models, where the following restriction on the generating class holds:

$$A \in \Gamma \quad \text{and} \quad A' \subset A \ \Rightarrow \ A' \in \Gamma \quad \forall\, A \in \Gamma \ ,$$

and some normalizing conditions are also needed (see [5]).

Note that any generating class of a log-linear model can be extended to meet the above condition by introducing constant interaction functions for the missing subsets. Therefore it is enough to consider those elements of a generating class which are not contained by an other one. These are called maximal interactions and from now on a generating class Γ will be defined by these only.

If $\mathrm{p_V}$ obeys a hierarchical log-linear model, it means that it can be constructed as the product of functions defined on its lower dimensional marginals up to a certain dimension. In other words, the product contain all the lower order interactions and main effects of the maximal interactions we wish to study.

In a special class of hierarchical log-linear models, the graphical ones, the generating class is specified with only the maximal interaction sets:

$$\mathcal{C} = \{C \,|\, C \text{ is a clique of the underlying graph}\} \ ,$$

thus $\Gamma = \mathcal{C}$. In this case, there is another equivalent form of Eq. (2.8) that uses an

exponential parameterization and shows that we are in **exponential family**:

$$\begin{aligned} \mathrm{p_V}(\mathbf{x}) = p_\theta(\mathbf{x}) &= \exp\left\{\sum_{C\in\mathcal{C}} \langle \theta_C \cdot I_C\rangle(\mathbf{x}) - Z(\theta)\right\} \\ &= \exp\left\{\sum_{C\in\mathcal{C}} \sum_{\mathbf{y}_C\in\mathcal{X}_C} \theta_{C,\mathbf{y}_C} \cdot I_{C,\mathbf{y}_C}(\mathbf{x}) - Z(\theta)\right\}. \end{aligned}$$

Here $\theta = \{\theta_C : C \in \mathcal{C}\}$ is the canonical parameter, where

$$\theta_C = \{\theta_{C,\mathbf{y}_C}\,,\, \mathbf{y}_C \in \mathcal{X}_C\} \in \mathbb{R}^{|\mathcal{X}_C|}$$

is a vector, and so θ is a $\sum_{C\in\mathcal{C}} |\mathcal{X}_C|$-dimensional vector, which dimension is usually less than $|\mathcal{X}_V| = \prod_{i=1}^n |\mathcal{X}_i|$.

The canonical statistic I_C also takes values in $\{0,1\}^{|\mathcal{X}_C|}$ for all $C \in \mathcal{C}$. In fact, the I_C's are multiple indicator functions consisting of usual indicator functions of all possible states in $\mathcal{X}_C$. More exactly,

$$I_C = \{I_{C,\mathbf{y}_C}\,,\, \mathbf{y}_C \in \mathcal{X}_C\} \in \mathbb{R}^{|\mathcal{X}_C|},$$

where $I_{C,\mathbf{y}_C}(\mathbf{x})$ is 1 if $\mathbf{x}_C = \mathbf{y}_C$ and 0 otherwise. Obviously $\langle .,. \rangle$ denoted the usual inner product in the above finite-dimensional spaces. $Z(\theta)$ is the log-partition function, which does not depend on $\mathbf{x} \in \mathcal{X}_V$. In accord with Eq. (2.8):

$$f_C(\mathbf{x}) := f_C(\mathbf{x}_C) = \sum_{\mathbf{y}_C\in\mathcal{X}_C} \theta_{C,\mathbf{y}_C} \cdot I_{C,\mathbf{y}_C}(\mathbf{x}) = \langle \theta_C, I_C\rangle \ .$$

In exponential family, the sum of the canonical statistics over the sample, i.e. the frequencies of the cells within the cliques:

$$\mathrm{N}(\mathbf{y}_C) = \sum_{i=1}^{\mathrm{m}} I_{C,\mathbf{y}_C}(\mathbf{x}^{(i)}) \ ,$$

are the sufficient statistics entering into the estimation of the mean value parameters (here m denotes the sample size).

The mean value parameters (in other words, moment parameters) are the expectations of the cell counts:

$$\mu(\mathbf{x}) := \mathbb{E}(\mathrm{N}(\mathbf{x})) \ .$$

In regular exponential families, there is a one-to-one correspondence between the mean value and the canonical parameters, see [23]. Further, the ML-estimate of the

mean value parameters come from the moment-matching equations

$$\mu(\mathbf{x}_C) = \mathrm{N}(\mathbf{x}_C), \quad \mathbf{x}_C \in \mathcal{X}_C \ , \ C \in \mathcal{C} \ .$$

This system of equations is solved by the Iterative Proportional Scaling algorithm of page 24.

Graphical log-linear models

Again, our log-linear model is hierarchical, so it suffices to store only the maximal interactions of the generating class Γ. Further we assume that each variable is included in at least one interaction, in other words, all main effects are present: $\cup_{A\in\Gamma} = V$.

To characterize this subclass, the following hypergraph notions are useful. The generating class Γ uniquely defines the following hypergraph H:

- the vertices correspond to the variables and constitute the vertex set $V = \{1, \ldots, n\}$ as before,
- the hyperedges are the elements of the maximal interactions of Γ.

As the model is hierarchical, the subsets of the maximal interactions are also interactions (elements of Γ), but they are not hyperedges in H.

The **interaction graph** $G^H = (V, E)$ corresponding to H, or equivalently, to the hierarchical log-linear model with generating class Γ, is defined in the obvious way: its vertex set is again V, and two vertices are connected if and only if they are together in at least one interaction.

Note that different hierarchical models may have the same interaction graph. However there is a class of models where there is a one-to-one correspondence between the model and its interaction graph. Therefore the interaction graph is capable to describe such a model.

Definition 2.7. *The hierarchical log-linear model with generating class Γ is **graphical** if the hyperedges in the hypergraph H defined above are identical to the cliques of the interaction graph G^H.*

For example, when the generating class is

$$\Gamma = \{\{1, 2\}, \{2, 3\}, \{1, 3\}\}, \tag{2.9}$$

then the interaction graph has the clique $\{1, 2, 3\}$, which is not an interaction. So our log-linear model is not a graphical interaction model. However, when the generating class is

$$\Gamma' = \{\{1, 2, 3\}\}, \tag{2.10}$$

then the interaction graph has the clique $\{1, 2, 3\}$ and it is an interaction in Γ' so our log-linear model is a graphical interaction model.

Note that model (2.9) corresponds to the Ising model on 3 vertices. When more than 3 vertices, define the Ising model, then the cliques are indeed the vertex-pairs, so those constitute the generating class at the same time.

Theorem 2.2 (see [18]). *The distribution* $\mathrm{p_V}$ *obeying the hierarchical log-linear model with generating class* Γ *defines an MRF if and only if the log-linear model is graphical.*

Decomposable log-linear models

In most of the applications we have a contingency table of large size. Even in case of binary variables, there are 2^n cells. However there are models where the ML-estimate of the cell probabilities under the model's assumptions can be given by explicit formulas. These models can be characterized by the special dependency structure of the variables when we build a graph or hypergraph on them. These are the decomposable models.

Definition 2.8. *The hierarchical log-linear model with generating class* Γ *is* ***decomposable*** *if its interaction graph is decomposable.*

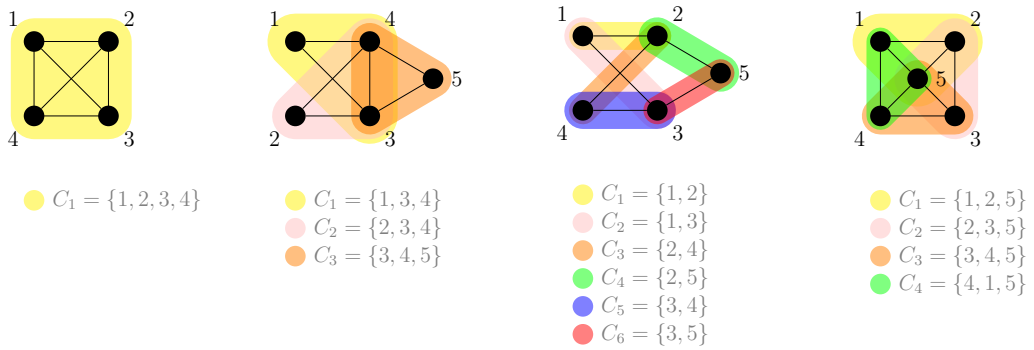
Definition 2.9. *The graph* G *is* ***decomposable*** *if it is either a complete graph or its vertices can be partitioned into disjoint subsets* $V = A \cup B \cup C$ *such that*

- $G(C)$ *(the subgraph induced by the vertices in* C*) is a complete subgraph;*
- C *separates* A *from* B *(in other words,* C *is a vertex cut-set between* A *and* B*: after removing the vertices of* C *the vertices of* A *and* B *will be in separate components);*
- *the subgraphs* $G(A \cup C)$ *and* $G(B \cup C)$ *are both decomposable.*

Proposition 2.1 (see [18]). *A log-linear model is graphical whenever it is decomposable.*

Again, in case of contingency tables the graphical log-linear models coincide with the MRFs. However, the decomposable models are proper subsets of these. In [5] the authors show examples of graphical interaction models that are not decomposable.

Note that some authors call the decomposable models Markov, as here the chain of the cliques behaves like a Markov chain. It is misleading since that special Markov chain property is stronger than the condition for being an MRF.



(a) (b) (c) (d)

Figure 2.3: Examples of graphical log-linear models and their maximal interactions

The concept of decomposability will be much more important in Sec. 3.1, where we show that this graph property is equivalent to chordality (see Def. 1.13), which is useful in many applications.

On Fig. 2.3 we can see some examples: (a) and (b) are decomposable, (c) and (d) are not. One may think that (d) is decomposable as well, but it is not: $\{1 - 2 - 3 - 4\}$ is a chordless 4-cycle in it.

2.3.3 Iterative proportional scaling

In hierarchical log-linear models, the mean value parameters, thus the cell probabilities are estimated based on the clique frequencies, and are obtainable by the Iterative Proportional Scaling (IPS) algorithm, see [22] and [13] or [4] for an information theoretic view. The goal of this algorithm is to make the clique probabilities equal to the corresponding relative frequencies, for all cliques.

Recall that

$$\{\, \mathrm{N}(\mathbf{x}_C) \;:\quad \mathbf{x}_C \in \mathcal{X}_C, \quad C \in \mathcal{C} \,\}$$

is a sufficient statistic for the canonical parameters of the log-linear model. Moreover, as we are in exponential family, the C-marginals of the ML-estimates ($\hat{\mu}()$'s) of the mean value parameters ($\mu()$'s) are equal to their relative frequencies

$$\hat{\mu}(\mathbf{x}_C) = \mathrm{N}(\mathbf{x}_C), \quad \mathbf{x} \in \mathcal{X}_C \;,\; C \in \mathcal{C} \;.$$

To solve the system of equations

$$\mu(\mathbf{x}_C) = \mathrm{N}(\mathbf{x}_C), \quad \mathbf{x}_C \in \mathcal{X}_C, \, C \in \mathcal{C},$$

we have to recursively adjust the above marginal counts going through the cliques in a cyclic iteration that finds the fixed point of the mapping $T = T_{C_1} \dots T_{C_k}$ (here

k is the cardinality of $\mathcal{C}$), where

$$T_{C_i}\mu^{(t)}(\mathbf{x}) = \mu^{(t)}(\mathbf{x})\frac{\mathrm{N}(\mathbf{x}_{C_i})}{\mu^{(t)}(\mathbf{x}_{C_i})}, \quad i = 1, \ldots, k \quad \forall \mathbf{x} \in \mathcal{X}_V \ .$$

So starting from some $\mu^{(0)}()$ defined for all $\mathbf{x} \in \mathcal{X}_V$, the iteration is

$$\mu^{(t+1)}(\mathbf{x}) = T\mu^{(t)}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}_V \ .$$

In [18] it is proved that if $\mathrm{N}(\mathbf{x}_C) > 0$ and $\mu^{(0)}(\mathbf{x}_C) > 0$, for all $\mathbf{x} \in \mathcal{X}_V$ and for all $C \in \mathcal{C}$, then the sequence $\mu^{(t)}(\mathbf{x})$ converges as $t \to \infty$, for all $\mathbf{x} \in \mathcal{X}_V$.

With the additional condition, that $\mu^{(0)}(\mathbf{x}_A) = \mathrm{N}(\mathbf{x}_A)$ can not hold for $A \notin \mathcal{C}$, the sequence $\mu^{(t)}(\mathbf{x})$ converges to the theoretically guaranteed unique ML-estimate of the mean value parameter $\mu(\mathbf{x})$:

$$\mu^{(t)}(\mathbf{x}) \to \hat{\mu}(\mathbf{x}) \quad \text{as} \quad t \to \infty, \quad \forall \mathbf{x} \in \mathcal{X}$$

or equivalently $\frac{\mu^{(t)}(\mathbf{x})}{m} \to \hat{p}(\mathbf{x})$, where m is the sample size.

The proof is based on information divergence minimization, see [18, 22]. The additional condition excludes the possibility that some extra subset of variables is added to the prescribed set of interactions (which are the cliques). In particular, the cell frequencies do not provide a good starting, as they belong to the saturated model. The suggested starting is the uniform distribution over the cells, i.e., $\mu^{(0)}(\mathbf{x}) = \frac{m}{|\mathcal{X}_V|}$, where of course $|\mathcal{X}_V|$ is the total number of the cells.

Note that the same idea is hidden behind the so-called covariance selection method in the Gaussian case, see Sec. 2.4.

In general, in hierarchical log-linear models, we cannot give the ML-estimate of the mean value parameters in explicit form, this is why the above infinite iteration is needed that converges to this estimate. However, when the log-linear model is decomposable, we have the ML-estimate in explicit form, and in accord with this, we can construct an iteration that converges in two runs. The iteration facilitates the quick computation of the clique marginals. This is the junction-tree algorithm of Sec. 3.3.

2.4 A continuous MRF: Gaussian graphical models

2.4.1 Partitioned covariance matrices and partial correlations

In this section we consider the multivariate normal distribution. Let $\mathbf{X} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be from now on an n-dimensional normal random vector with expectation (vector) $\boldsymbol{\mu}$ and positive definite (notation: > 0), symmetric $n \times n$ covariance matrix $\boldsymbol{\Sigma}$. Note

that this distribution belongs to the exponential family with canonical parameter $(\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})$. The also positive definite, symmetric matrix $\boldsymbol{K} := \boldsymbol{\Sigma}^{-1}$ of entries k^{ij} is called **concentration matrix**, and its zero entries indicate conditional independences between two components of $\mathbf{X}$, conditioned on the remaining components. This is supported by the following facts, where the initial setup is that we take the disjoint partitioning $A \cup B$ of the n variables, where $|A| = p$ and $|B| = q$.

Proposition 2.2 (see [1]). *Let the $n \times n$ covariance matrix $\boldsymbol{\Sigma} > 0$ be partitioned as*

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{pmatrix} ,$$

where $\boldsymbol{\Sigma}_{AA}$, $\boldsymbol{\Sigma}_{BB}$ are covariance matrices of $\mathbf{X}_A$ and $\mathbf{X}_B$, where $\boldsymbol{\Sigma}_{AB} = \boldsymbol{\Sigma}_{BA}^T$ is their cross-covariance matrix. Then the symmetric matrix $\boldsymbol{\Sigma}^{-1} > 0$ has the following partitioned form:

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Sigma}_{A|B}^{-1} & -\boldsymbol{\Sigma}_{A|B}^{-1}\boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1} \\ -\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{A|B}^{-1} & \boldsymbol{\Sigma}_{BB}^{-1} + \boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{A|B}^{-1}\boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1} \end{pmatrix} ,$$

where

$$\boldsymbol{\Sigma}_{A|B} := \boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA} .$$

Further, $\boldsymbol{\Sigma} > 0$ is equivalent to the fact that both $\boldsymbol{\Sigma}_{BB}$ and $\boldsymbol{\Sigma}_{A|B}$ are regular (invertible) matrices (actually they are positive definite).

Theorem 2.3. *Let $(\mathbf{X}_A^T, \mathbf{X}_B^T)^T \sim \mathcal{N}_{p+q}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a random vector, where the expectation $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are partitioned (with block sizes p and q) in the following way:*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{pmatrix} .$$

Then the random vector $\mathbf{X}_A$ conditioned on $\mathbf{X}_B = \mathbf{x}_B$ has the conditional distribution of $\mathcal{N}_p(\boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B) + \boldsymbol{\mu}_A, \boldsymbol{\Sigma}_{A|B})$.

Note that the conditional covariance matrix $\boldsymbol{\Sigma}_{A|B}$ does not depend on the value $\mathbf{x}_B$ of the condition $\mathbf{X}_B$. Further, for the conditional expectation, which is the expectation of the conditional distribution, we get that

$$\mathbb{E}(\mathbf{X}_A \mid \mathbf{X}_B = \mathbf{x}_B) = \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B) + \boldsymbol{\mu}_A .$$

Therefore,

$$\mathbb{E}(\mathbf{X}_A|\mathbf{X}_B) = \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\mathbf{X}_B - \boldsymbol{\mu}_B) + \boldsymbol{\mu}_A$$

which is a linear function of the coordinates of $\mathbf{X}_B$. In the $p = q = 1$ case, it is called **regression line**, while in the $p = 1$, $q > 1$ case, **regression plane**.

Summarizing, in case of the multidimensional normal distribution, the regression functions are linear functions of the variables in the condition, which fact has important consequences in the multivariate statistical analysis. Since $\boldsymbol{\mu}$ is just a shift, in the following parts we will assume $\boldsymbol{\mu} = \mathbf{0}$, i.e. the variables are mean centered.

Theorem 2.4. *Let $\mathbf{X} = (X_1, \ldots, X_n)^T \sim \mathcal{N}_n(\mathbf{0}, \boldsymbol{\Sigma})$ be a random vector, and let $V := \{1, \ldots, n\}$ denote the index set of the variables, $n \geq 3$. Assume that $\boldsymbol{\Sigma}$ is positive definite. Then*

$$r_{X_i X_j | \mathbf{X}_{V \setminus \{i,j\}}} = \frac{-k_{ij}}{\sqrt{k_{ii} k_{jj}}} \qquad i \neq j,$$

*where $r_{X_i X_j | \mathbf{X}_{V \setminus \{i,j\}}}$ denotes the **partial correlation coefficient** between X_i and X_j after eliminating the effect of the remaining variables $\mathbf{X}_{V \setminus \{i,j\}}$. Further,*

$$k_{ii} = \frac{1}{\mathrm{Var}(X_i \mid \mathbf{X}_{V \setminus \{i\}})} \qquad i = 1, \ldots, n \ ,$$

is the reciprocal of the conditional (residual) variance of X_i conditioned on the other variables $\mathbf{X}_{V \setminus \{i\}}$.

Definition 2.10. *Let $\mathbf{X} \sim \mathcal{N}_n(\mathbf{0}, \boldsymbol{\Sigma})$ be random vector with positive definite $\boldsymbol{\Sigma}$. Consider the regression plane*

$$\mathbb{E}(X_i \mid \mathbf{X}_{V \setminus \{i\}} = \mathbf{x}_{V \setminus \{i\}}) = \sum_{j \in V \setminus \{i\}} \beta_{ji \cdot V \setminus \{i\}} \cdot x_j, \quad j \in V \setminus \{i\} \ ,$$

*where x_j's are the coordinates of $\mathbf{x}_{V \setminus \{i\}}$. Then we call the coefficient $\beta_{ji \cdot V \setminus \{i\}}$ the **partial regression coefficient** of X_j when regressing X_i with $\mathbf{X}_{V \setminus \{i\}}$, $j \in V \setminus \{i\}$.*

Theorem 2.5. *Consider the above setting. The following is true:*

$$\beta_{ji \cdot V \setminus \{i\}} = -\frac{k_{ij}}{k_{ii}}, \quad j \in V \setminus \{i\} \ .$$

Corollary 2.1. *An important consequence of Thm. 2.4 and Thm. 2.5 is that*

$$\beta_{ji \cdot V \setminus \{i\}} = r_{X_i X_j | \mathbf{X}_{V \setminus \{i,j\}}} \sqrt{\frac{k_{jj}}{k_{ii}}} = r_{X_i X_j | \mathbf{X}_{V \setminus \{i,j\}}} \sqrt{\frac{\mathrm{Var}(X_i \mid \mathbf{X}_{V \setminus \{i\}})}{\mathrm{Var}(X_j \mid \mathbf{X}_{V \setminus \{j\}})}}, \quad \forall j \in V \setminus \{i\} \ .$$

The formula is analogous to the one used in unconditioned regression. Therefore if we do a regression of X_i with the other variables, only the X_j variables whose partial correlation with X_i (after eliminating the effect of the remaining variables) is not 0 enter into the regression.

Testing hypotheses about partial correlations.

For $i \neq j$ we want to test the hypothesis:

$$H_0 \,:\, r_{X_i X_j | \mathbf{X}_{V\setminus\{i,j\}}} = 0 \ .$$

I.e., that X_i and X_j are conditionally independent conditioned on the remaining variables. Equivalently, H_0 means that $\beta_{ij|V\setminus\{i\}} = 0$, $\beta_{ji|V\setminus\{j\}} = 0$, or simply $k_{ij} = k_{ji} = 0$. Note that $\mathbf{\Sigma} > 0$ is assumed.

To test H_0 in some form, several exact tests are known that are usually based on likelihood ratio tests. The following test uses the empirical partial correlation coefficient, denoted by $\hat{r}_{X_i X_j | \mathbf{X}_{V\setminus\{i,j\}}}$, and the following statistic is based on it:

$$B = 1 - (\hat{r}_{X_i X_j | \mathbf{X}_{V\setminus\{i,j\}}})^2 = \frac{|\mathbf{S}_{V\setminus\{i,j\}}| \cdot |\mathbf{S}_V|}{|\mathbf{S}_{V\setminus\{i\}}| \cdot |\mathbf{S}_{V\setminus\{j\}}|} \ ,$$

where $\mathbf{S}$ is the sample size times the empirical covariance matrix of the variables in the subscript (its entries are the product-moments).

It can be proven that under H_0 the test statistic

$$T = \sqrt{m-n} \cdot \sqrt{\frac{1}{B} - 1} = \sqrt{m-n} \cdot \frac{|\hat{r}_{X_i X_j | \mathbf{X}_{V\setminus\{i,j\}}}|}{\sqrt{1 - (\hat{r}_{X_i X_j | \mathbf{X}_{V\setminus\{i,j\}}})^2}}$$

is Student's t distributed with $m-n$ degrees of freedom. Therefore we reject H_0 for large values of T.

2.4.2 The model and the covariance selection

Let again $\mathbf{X}_V \sim \mathcal{N}_n(\boldsymbol{\mu}, \mathbf{\Sigma})$ be an n-dimensional normal random vector. We can form a graph G on the vertex/variable-set V, where V corresponds to the components of $\mathbf{X}_V$ and the edges are drawn according to the rule:

$$[i,j] \in E \quad \Leftrightarrow \quad k_{ij} \neq 0 \ , \qquad \forall\, i \neq j \ .$$

This is called **Gaussian graphical model**. In applications (when $\mathbf{\Sigma}^{-1}$ is unknown) we use the empirical partial correlation coefficients, and based on them, the above exact test of Sec. 2.4.1 to check whether they significantly differ from 0 or not. If we put zeros (based on the test results) into the no-edge ij positions of the inverse covariance matrix, we can fit a so-called **covariance selection model**. The restricted covariance matrix is denoted by $\mathbf{\Sigma}^*$. With the help of the concentration matrix $\boldsymbol{K} = \mathbf{\Sigma}^{-1}$ ($\{k_{ij}\}$) and the vector $\mathbf{h} = \boldsymbol{K}\boldsymbol{\mu}$ ($\{h_i\}$), the log-density of $\mathbf{X}$ has

the following form:

$$\ln f(\mathbf{x}) = c - \frac{1}{2}\sum_{i\in V} k_{ii}x_i^2 + \sum_{i\in V} h_i x_i - \sum_{i\neq j} k_{ij}x_i x_j \ ,$$

where c is a normalizing constant. Compared to the log-linear model of Sec. 2.3.2, the log-density is additively decomposed as:

- *quadratic main effects* with coefficients $-\frac{1}{2}k_{ii}$,
- *linear main effects* with coefficients h_i,
- *quadratic interactions* with coefficients $-k_{ij}$.

Notice that the interaction terms of the highest order involve pairs of variables, and there are no terms involving groups of variables with more than two elements. The hyperedges are usual edges. This is in contrast to the discrete case and it follows that within the normal distributions there are no hierarchical interaction models which are not graphical. So it is an MRF.

Covariance selection model

Given a G undirected graph and a sample (of more than n elements), we want to fit now a Gaussian distribution so that X_i is conditionally independent of X_j given the remaining variables, denoted by $X_i \perp\!\!\!\perp X_j \mid \mathbf{X}_{V\setminus\{i,j\}}$, whenever there is no edge between i and j in G. This is the Pairwise Markov property of Eq. 2.4, which is equivalent to the Local and Global Markov properties since we have a positive distribution.

That is, we want to estimate the mean value parameters ($\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$) from the i.i.d. sample $\mathbf{X}_1, \ldots, \mathbf{X}_m \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (again $m > n$), such that the concentration matrix ($\boldsymbol{K}$) has zero entries in the no-edge positions: $k_{ij} = 0$ when $[i,j] \notin E$.

This can be done by the covariance selection model. It can be proven (see Thm. 5.3 of [13]) that under this model the ML-estimate of the parameters are: $\hat{\boldsymbol{\mu}} = \overline{\mathbf{X}} = \frac{1}{m}\sum_{i=1}^m \mathbf{X}_i$ and that of the restricted covariance matrix $\boldsymbol{\Sigma}^* = \{\sigma_{ij}^*\}_1^n$ can be calculated as follows. We estimate the entries in the edge-positions as in the saturated model (no restrictions):

$$\hat{\sigma}_{ij}^* = \frac{1}{m}s_{ij} \quad \forall\, [i,j] \in E \ ,$$

where $\mathbf{S} = \{s_{ij}\}_1^n = \sum_{k=1}^m (\mathbf{X}_k - \overline{\mathbf{X}})(\mathbf{X}_k - \overline{\mathbf{X}})^T$. The other entries (in the no-edge positions) of $\boldsymbol{\Sigma}^*$ are free, but after taking $\boldsymbol{K}^* = \{k_{ij}^*\}_1^n = \boldsymbol{\Sigma}^{*-1}$ with these undetermined entries, we get the same number of equations for them from $k_{ij}^* = 0$ whenever $[i,j] \notin E$. To do so, there are numerical algorithms at our disposal, for

instance, the iterative proportional scaling (see [13]). Actually the equations can be stated for the cliques, and instead of the $m > n$ condition $m > c$ would suffice, where c is the cardinality of the largest clique.

Chapter 3

Chordality and decomposable graphical models

In this chapter we will focus on chordality and graphical models with decomposable graph representation. In Sec. 3.1 we examine equivalences of chordality. In Sec. 3.2 we show commonly used methods, usually preprocesses, used for such models. In Sec. 3.3 we give a detailed description of the junction-tree algorithm. The goal of this algorithm is to find marginal distributions for the variables in a graphical model, possibly after absorbing some evidence. This algorithm makes the marginalization more effective. We need two things to be able to construct a junction tree and use the algorithm:

- the probability distribution is a Gibbs-distribution with respect to the given graph G;
- the graph representation of the probability space (G) is triangulated, possibly after some modifications.

It basically applies the belief propagation, also known as sum-product message passing method on a modified graph structure called junction tree. It utilizes the fact that the belief propagation results in exact marginals for tree factor graphs, and with a proper scheduling, it terminates in two steps.

3.1 Triangulated graphs

Recall that a simple undirected graph $G = (V, E)$ is **triangulated** (or chordal) if every cycle with more than 3 vertices has a chord. Just by inspection it is hard to tell if a graph is triangulated or not. Luckily there are several properties which are equivalent to the graph being triangulated.

Proposition 3.1. *The following are all equivalent for a graph G:*

- *G is **triangulated**;*
- *G is **decomposable**;*
- *the vertices of G have a **perfect elimination ordering** (or perfect numbering);*
- *the cliques of G have a **junction tree** structure;*
- *the cliques of G enjoy the **running intersection property** (RIP);*
- *the cliques of G fulfill **Sundberg's criterion**;*
- *G is **recursively simplicial**;*
- *there is a numbering of the vertices in which order the adjacency matrix of G contains a **reducible zero pattern**.*

Before the proof we need to define the above terms. Recall that a **perfect elimination ordering** of the vertices in a graph is such that for each vertex v the neighbors of it that come later in the ordering form a complete subgraph. It is often called simply a perfect numbering. Now see the other terms.

Definition 3.1. *A graph G has a **junction tree** structure if the cliques of G can be represented with a tree graph, where every node corresponds to one clique of G and for any pair of cliques C_i, C_j, every clique on the (unique) path connecting them in this tree contains $C_i \cap C_j$.*

This structure is also called clique tree, join tree or tree decomposition.

Definition 3.2. *A graph G enjoys the **running intersection property** (RIP) if there is an ordering of the cliques of G $(C_1, \ldots, C_K)$ such that:*

- *$H_{j-1} = C_1 \cup \cdots \cup C_{j-1}$ and C_j is a decomposition of G for all j, i.e. $S_j = H_{j-1} \cap C_j$ is a separator,*
- *$R_j = C_j \setminus S_j$ is called the j'th residual,*
- *$S_1 = \emptyset$ and $R_1 = C_1$;*

Note that in some literature the above sequence of cliques is called the junction tree. Indeed, it gives a tree structure in the sense that we can represent this with a graph, where every vertex represents a clique and the edges represent the separators between them. In this case it is also worth noting that not every pairwise separator is represented, however some are represented with multiplicity.

Definition 3.3. *A graph G fulfills* ***Sundberg's criterion*** *if there is an ordering of the maximal cliques of G $(C_1, \ldots, C_K)$ such that:*

- *C_j and $H_{j+1} = C_{j+1} \cup \cdots \cup C_K$ is a decomposition of G for all j, i.e. $S_j = C_j \cap H_{j+1}$ is a separator,*
- *$S_k = \emptyset$ and $R_k = C_k$,*

Note that the above ordering [19] is a reversed RIP ordering.

Definition 3.4. *A vertex is* ***simplicial*** *in a graph if its neighbors form a complete subgraph. A graph is* ***recursively simplicial*** *if it contains a simplicial vertex and when it is removed the remaining subgraph is recursively simplicial.*

Note that if for a vertex i, $\mathrm{Adj}(i)$ is a complete subgraph then $\mathrm{Adj}(i) \cup \{i\}$ is a complete subgraph as well.

Definition 3.5. *The adjacency matrix of graph G contains a* ***reducible zero pattern*** *if there is a numbering of the vertices such that:*

- *if we denote with $I_0^G = \{(i,j) \mid i < j,\ (i,j) \in V \times V,\ A_{i,j}^G = 0\}$ the coordinate-pairs where the upper part of adjacency matrix (A^G) has zero elements, then for each $(i,j) \in I_0^G$ and every $h \in \{1, 2, \ldots, i-1\}$ it is true that $(h,i) \in I_0^G$ or $(h,j) \in I_0^G$ or both.*

Proof. We will not prove every direction.

triangulated $\Rightarrow$ decomposable:

We can prove by induction that every chordal graph with n vertices is decomposable. This is trivially true for $n = 1$. Suppose that it is true for any n, then by following argument it is true for a graph G with $n+1$ vertices:

1.: If G is complete, it is decomposable by definition, so suppose that G is not complete.

2.: Since G is not complete, V contains $a,\ b$ that are not neighbors. Let $S \subset V$ be a minimal set that separates a from b. Note that S can be empty. Let A be the subset of $V \setminus S$ connected to a by some path in $V \setminus S$, and let B be the remainder, $B = V \setminus S \setminus A$, then S separates A from B in G. Therefore we have V as the disjoint union $V = A \cup S \cup B$, where S separates A from B in G and $A,\ B$ are nonempty.

3.: We show that S is complete. Assume that S has cardinality at least 2, otherwise it is trivially complete. For any two distinct nodes $u,\ v$ in S, there are paths $(u, a_1, \ldots, a_k, v)$ and $(u, b_1, \ldots, b_l, v)$ with $a_i \in A$, $b_i \in B$ and $k, l \geq 1$. Since S is a minimal set that separates a from b , there must be a path from a to u and from a to v, since the absence of one of these paths would imply that S was not minimal. Now we have to see that u and v are neighbors. Take the path from u to v

through A with minimal length, and similarly the path from u to v through B also with minimal length. This pair of paths forms a cycle, which must have a chord. The chord has to be between u and v since the minimality of the paths implies that the chord cannot be between vertices that are both in A, nor can it be between vertices that are both in B . And because S separates A from B the chord cannot be between a vertex in A and one in B.

4.: The subgraphs induced by $A \cup S$ and $B \cup S$ are chordal. Since if one of these subgraphs contains a chordless cycle, then so does G.

5.: By the inductive hypothesis, these subgraphs are strictly smaller than G and hence decomposable.

decomposable $\Rightarrow$ recursively simplicial:

We can prove by induction that every decomposable graph with n vertices is recursively simplicial. This is trivially true for $n = 1$.

1.: Since G is decomposable it contains a simplicial vertex. For this, we can prove by induction the that any decomposable graph is either complete or has two nonadjacent simplicial vertices which is a stronger statement. This is trivially true for graphs with $|V| = 1$, and for the induction step, we notice that any decomposable G is either complete, or V can be decomposed as sets A, S, B. If $A \cup S$ is complete, then any $a \in A$ is simplicial. Otherwise the subgraph induced by $A \cup S$ has two non-adjacent simplicial nodes, by the inductive hypothesis. Since S is complete, one of these must be in A. Similarly, there is a simplicial $b \in B$.

2.: Since G is decomposable the subgraph corresponding to a subset of the vertices is decomposable. Again, we can prove this by induction. It is trivially true for graphs with $|V| = 1$. For the induction step, the result is trivially true if G is complete, otherwise we consider the usual decomposition of V into A, S, B, where S is complete and $A \cup S$ and $S \cup B$ are decomposable. By the inductive hypothesis, removing a node from S leaves $A \cup S$ and $S \cup B$ decomposable. Removing a node from A leaves $S \cup B$ unchanged, and either leaves A empty, in which case the remaining subgraph, $S \cup B$, is decomposable, or leaves $A \cup S$ decomposable by the inductive hypothesis. Thus, the subgraph that remains when we remove a simplicial vertex v from a decomposable G is also decomposable.

recursively simplicial $\Rightarrow \exists$ junction tree:

We can prove by induction that every recursively simplicial graph with n vertices has a junction tree. This is trivially true for $n = 1$. Take a simplicial vertex v from G , and let G' be the subgraph that remains when we remove v. By the inductive hypothesis, G' has a junction tree T', and this can be extended to give a junction tree for G. To see this, let C' be a clique in T' containing all neighbors of v in G . If C' is precisely the set of neighbors of v , then we can add v to C' to give a junction tree for G. It contains all cliques, and v is not in any other clique, so the junction

tree property is trivially satisfied. If not, that is, if C' contains the neighbors of v as a proper subset, then we add a new clique containing v and its neighbors to T', with an edge to C'. Since v is in no other clique and $C \setminus \{v\}$ is a subset of C', this is a junction tree for G.

$\exists$ junction tree $\Rightarrow$ triangulated:

We can prove by induction that the statement is true for junction trees with k nodes. If the clique tree has only one node, then G is complete, hence chordal. Assuming that the statement is true for some value of k, consider a graph with a junction tree T containing $k+1$ nodes. Fix a leaf C of T, and let C' be the neighbor of C in T, and let T' be the tree that remains when C is removed.

1.: If $C \subseteq C'$, then T' is a junction tree for G.

2.: On the other hand, if $C \cap C' \subseteq C$, removing the nonempty set $R = C \setminus C'$ from V leaves a subgraph G' that is chordal. To see this, notice that R has an empty intersection with every clique in T'. It is easy to see that T' is a junction tree for G' , and so G' is chordal.

3.: It follows that G contains no chordless cycles. Indeed, if a cycle is entirely in G', it is not chordless. If the cycle is entirely in the complete subgraph defined by C , it is not chordless. If the cycle intersects R , $C \cap C'$ and $V \setminus C$, then since the subgraph defined by $C \cap C'$ is complete, the cycle has a chord.

recursively simplicial $\Leftrightarrow$ $\exists$ perfect elimination ordering:

Define a simplicial vertex sequence as an ordering of the vertices of G that exhibits the recursively simpliciality of the graph. As we progressively remove the nodes in this order, the next node in the order is simplicial in the remaining subgraph. The same ordering is also a perfect elimination ordering trivially and vice versa.

Sundberg's criterion $\Leftrightarrow$ running intersection property:

As noted before the two ordering is trivially the reversed of one another.

$\exists$ perfect elimination ordering $\Rightarrow$ RZP in an ordering:

The condition of reducible zero pattern can be translated as follows: under a given ordering of the vertices if $i \nrightarrow j$ for some $i < j$, then $h \nrightarrow i$ or $h \nrightarrow j$ (or both) for all $h < i$. It is easy to see that from a perfect elimination ordering $1, 2, \ldots, n$ we can read out a RIP ordering of the cliques as:

Residuals	perfect elimination ordering
$R_1 = C_1$	$n, n-1, \ldots, n-\lvert C_1\rvert+1,$
$R_2 = C_2 \setminus S_2$	$n-\lvert C_1\rvert, \ldots, n-\lvert C_1\rvert-\lvert C_2 \setminus S_2\rvert+1,$
$\ldots$	$\ldots$
$R_k = C_k \setminus S_k$	$\lvert C_k \setminus S_k\rvert, \ldots, 1$

Now we check the RZP condition for the given perfect ordering by separating the cases based on the indicated RIP ordering.

If $i \in C_1$, then $\nexists j > i$ such that $i \nrightarrow j$. ✓

If $i \in C_l \setminus S_l$ for an arbitrary $l \in \{2, 3, \ldots, k\}$, then for a $j > i$ the following cases are possible:

- $j \in C_l \setminus S_l$, so $i \to j$; ✓
- $j \in C_m \setminus S_m$ for an $m < l$ and $i \to j$, so $j \in S_l$, because that separates $C_l \setminus S_l$ from the union of the previous cliques; ✓
- $j \in C_m \setminus S_m$ for an $m < l$ and $i \nrightarrow j$:
  - If $m = l - 1$, meaning $j \in C_{l-1} \setminus S_{l-1}$, then $j \notin S_l$, because $i \to s$ for all $s \in S_l$, so $h \nrightarrow j$ for all $h < i$, because S_l is a separator. ✓
  - If $m < l - 1$, then $h \nrightarrow j$ for all $h < i$, because S_l is separator and for all $s \in S_l$ $j > s > i$. ✓

Here we proved more. Not just that the existence of a perfect ordering guarantees RZP, but a perfect ordering is also a suitable ordering for a RZP. □

3.2 Methods to achieve chordality and to work with such graphs

There are steps which are necessary before we can use the forthcoming junction-tree algorithm. These steps are useful in case of other algorithms, which operate on undirected or chordal graphs, or use the cliques or junction tree structure:

1. Moralization: if the graph is directed;
2. Triangulation: to make the graph chordal;
3. Clique-identification: easy for chordal graphs;
4. Junction tree construction.

We can start for example with a BN or an MRF, but similar steps, maybe some additional ones, are necessary in other cases.

3.2.1 Moralization

Moralization is needed if initially we had a directed graphical representation, for example a Bayesian network. We will discuss this. In other directed cases we work similarly, but mostly lean on expert judgment.

The factors of a Bayesian network are marginal or conditional probabilities, so these are also potential functions in the sense described earlier on page 16. Therefore a directed factorization is also an undirected factorization (with $Z = 1$ normalization constant). Each vertex subset consists of a variable and its parents from the Bayesian network. We can transform a Bayesian network into a Markov random field by connecting (marrying) its parents for each variable and then dropping the edge directions. By the moralization there will be conditional independences in the (original) BN model that are not represented by the (new) MRF model.

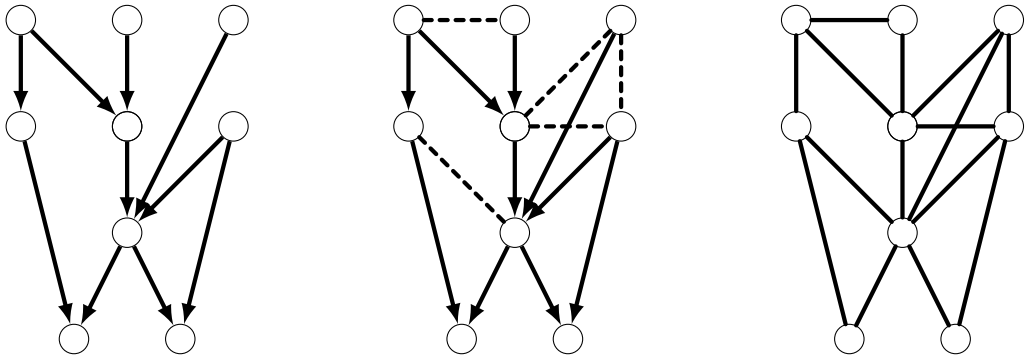


Figure 3.1: Moralization

3.2.2 Triangulation

As mentioned above some algorithms can be used on chordal graphs only. At this point, if we are lucky we already have a triangulated graph. Otherwise we can triangulate (fill-in) the graph, for which procedure there are several methods. We will describe one possible way. For more details on fill-in go back to page 8.

In [21] next to the Maximal Cardinality Search (discussed in Sec. 3.2.3) other issues are also discussed, for example a fill-in procedure for undirected graphs which are not triangulated. The fill-in procedure transforms the arbitrary undirected graph into a chordal one. Finding a fill-in is easy, finding an optimal or minimal fill-in however is a computationally hard problem.

Based on an arbitrary ordering of a graph we can get a fill-in as follows: starting from vertex 1, connect the neighbors of it (with each other) which are not already connected, take the next vertex and connect the higher numbered neighbors of it (with each other) which are not already connected, repeat this step until vertex n.

This method produces a triangulated graph for which the ordering of the vertices used during the triangulation is suitable to use in the junction tree construction since it is a perfect ordering in the new graph. Note however that this fill-in is really not optimal, and rejects possible numerous conditional independence assumptions.

Note that fill-in can also be defined for directed graphs, see e.g. [20]. The process of triangulation is called fill-in in [14] and [22] as well. For more details see Sec. 1.2 again.

3.2.3 Junction tree construction

A **cluster tree** of a graph G is a tree whose nodes represent some vertex-subsets of G with the property that if a vertex (of G) is in two distinct nodes of the tree, then the variable must be in every node (again these are vertex-subsets of G) on the path between the two nodes. Also, for every edge of G, there must be a node which contains both endpoints. This is in line with our earlier definition of the junction tree. By this we mean that the cluster tree whose nodes are the cliques of the (previously) triangulated moral graph will be our junction tree.

The junction tree construction is done based on a perfect ordering of the vertices. There are numerous algorithms to find a perfect ordering of the vertices in a graph, see [12].

To get the cliques of the triangulated graph and construct a junction tree based on a perfect ordering $1, 2, \ldots n-1, n$ we proceed as follows, based on [24]:

1. Take the vertex labeled 1 and its neighbors and let them be M_1.

2. Take the vertex labeled i and its neighbors labeled later ($j > i$) and let them be M_i for all $i = 2, 3, \ldots$.

3. Since the input was a perfect ordering, every M_i defined the above way will be a complete subgraph, so we throw away those which are not cliques, i.e. those which are contained by an "earlier" one.

4. We sort (relabel) the remained M_i's (now truly cliques) based on their index i decreasingly to get a RIP-ordering, or increasingly to get a Sundberg's ordering.

Let the ordering of the cliques $C_k, C_{k-1}, \ldots, C_1$ (of course $k < n$ since a chordal graph can contain only that many cliques) be the above described one, which fulfills the running intersection property. This ordering gives us a junction tree, see [13, 14]. The separators are given by intersections specified in the definition. Note again that not every pairwise clique-intersections are present as separators, however some of them are present with multiplicity.

Since this ordering is perfect, it is also true that for all non-adjacent $x, y \in V$ pair the minimal separator is a complete subgraph, see Thm. 1.1. Let's denote our cliques with $C_1, C_2, \ldots, C_k$.

Again, the perfect ordering of the vertices is translated to a λ ordering of the cliques. For this λ ordering

$$C_{\lambda(i)} \cap \Big(\bigcup_{j:\lambda(j)<\lambda(i)} C_{\lambda(j)} \Big) = C_{\lambda(i)} \cap C_{\lambda(j^*)} = S_{\lambda(i)}$$

for a $\lambda(j^*) < \lambda(i)$, for any i. A junction tree is easily constructed by attaching $C_\lambda(i)$ to any $C_\lambda(j^*)$ satisfying the above. Although j^* may not be uniquely determined, $S_{\lambda(i)}$ is. Indeed, the $S_{\lambda(i)}$ sets are minimal complete separators.

Notice that now we have an ordering of the vertices $(1, 2, \ldots n)$ where the partitioning of them (by the RIP ordering of the cliques) comes as:

$$C_k,\ C_{k-1} \setminus C_k,\ \ldots,\ C_2 \setminus C_3,\ C_1 \setminus C_2\ .$$

We can get a partitioning

$$C_k \setminus S_k,\ S_k,\ C_{k-1} \setminus C_k \setminus S_{k-1},\ S_{k-1},\ \ldots,\ S_2,\ C_1 \setminus C_2$$

by simply permuting the labels of the vertices party-by-part in the order $k,\ k-1,\ \ldots 1$; first in C_k and so on where it is necessary. This relabeling will not hurt the junction tree structure and it is an other perfect ordering of the vertices, see Fig. 3.2. It gives us a more causal way to look at the vertices, for example we can form a DAG in this modified perfect ordering on the skeleton of G, $j \to i$ if $i < j$. This way we can think of the labels as ages.

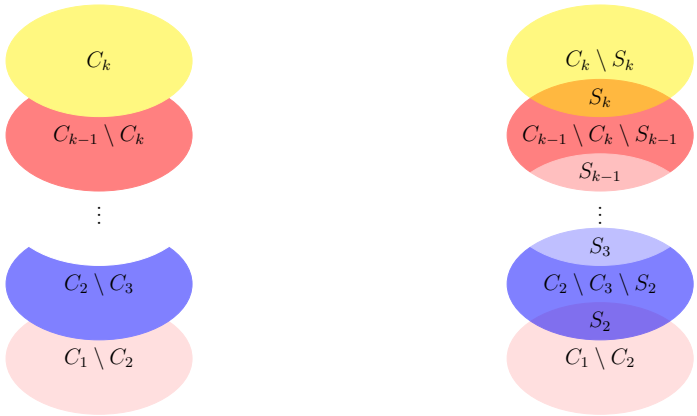


Figure 3.2: Junction tree reordering

Maximum Cardinality Search

It is crucial to find a perfect elimination ordering and there are many ways to find one. A method, called **Maximal cardinality search** (MCS) was first described by Tarjan and Yannakakis in [21]. The algorithm is as follows:

Number the vertices from n to 1 in decreasing order. As the next vertex to number, select the vertex adjacent to the largest number of previously numbered vertices, breaking ties arbitrarily.

Lemma 3.1. *Let α be an ordering of a chordal graph G. It is perfect if it has the following property:*

$$i <_\alpha j <_\alpha k,\ [i,k] \in E,\ [j,k] \notin E \Rightarrow \exists x \in V\ s.t.\ j <_\alpha x,\ [j,x] \in E,\ [i,x] \notin E$$

Theorem 3.1. *Any ordering generated by MCS has the property in the previous lemma, thus it is a perfect ordering if G is chordal.*

In other words, if we have a chordal graph, and we know that it is a chordal graph, then an ordering generated by the MCS is a perfect elimination (or zero-fill-in) ordering. The question is: We got an ordering α from the fairly simple MCS-algorithm, but is our graph chordal at all?

From Prop. 3.1 it is clear that chordality is equivalent to the existence of a perfect elimination/ zero fill-in/ monotone transitive ordering. So we just have to check whether our so obtained ordering α is perfect or not.

On smaller examples it is an easy task that can be achieved by simple eliminating the vertices in the given order by hand. On larger examples however it is time-consuming. Luckily the problem has many solutions implemented in linear (as in $|V| + |E|$) time. Here we recommend a method which was presented in the original paper for vertex elimination [20].

Definition 3.6. *Let $G^\star_\alpha = (V, E \cup F(G_\alpha))$ be an elimination graph as defined earlier. For any vertex v let $f(v)$ be the follower of v, which is the vertex of smallest number that is both adjacent to v in $G^\star_\alpha$ and has a number larger than that of v:*

$$f(v) := min(\mathrm{MAdj}(v)) \ .$$

We define $f^i(v)$ for $i \geq 0$ as $f^0(v) = v$, $f^{i+1}(v) = f(f^i(v))$.

Lemma 3.2. *If $[v,w] \in E \cup F(G_\alpha)$ and $v <_\alpha w$, then $\exists\ i \geq 1$ such that $f^i(v) = w$.*

Theorem 3.2. *$[v,w] \in E \cup F(G_\alpha)$ for some $v <_\alpha w$ pair if and only if $\exists\ x \in V$ s.t. $[x,w] \in E$ and $f^i(x) = v$ for some $i \geq 0$.*

This Theorem leads to an algorithm to find the fill-in:

- We process the vertices of $G_\alpha = (V, E, \alpha)$ from 1 to n.
- First for a vertex w let $A(w) := \{v \in V |\ [v, w] \in E \quad \text{and} \quad v <_\alpha w\}$ be the smaller numbered neighbors.
- Repeat the following step: Select $v \in A(w)$ s.t. $f(v)$ is already computed $(f(v) <_\alpha w)$ and $f(v) \notin A(w)$. Add this $f(v)$ to $A(w)$.
- Now on the reconstructed $A(w)$ let $f(v) := w$ for those $v \in A(w)$ which for we did not calculated $f(v)$ yet.

If we only want to test for zero fill-in, we can restate the algorithm as follows. Compute $f(v)$ for every vertex v. For every $[v, w] \in E$ s.t. $v <_\alpha w$ verify that either $(f(v), w) \in E$ or $f(v) = w$.

Generalization for hypergraphs

A hypergraph $H = (V, E)$ consists of a set of vertices V and a set of (hyper)edges E, each (hyper)edge is a subset of V. The (primal) graph G^H of a hypergraph H is the graph whose vertices are those of H and whose edges are the vertex pairs $[v, w]$ such that v and w are in a common edge of H. (See [21].)

Definition 3.7. *A hypergraph H is **conformal** if every complete subgraph of G^H is contained in a (hyper)edge of H, or equivalently if every clique of its primal graph is a hyperedge.*

Definition 3.8. *A hypergraph H is **acyclic** if H is conformal and G^H is triangulated (chordal).*

Theorem 3.3. *A hypergraph H is acyclic if and only if either of the following conditions holds:*

1. *All vertices of H can be deleted by repeatedly applying the two operations below:*
   (a) *erase a vertex that occurs in only one edge,*
   (b) *erase an edge that is contained in another edge.*
2. *There is a forest F (called join forest) such that its vertices are the edges of H and for every v vertex of H the subgraph of F is connected, which subgraph is induced by the vertices of F (edges of H) that contain v.*

The second condition is again the so-called running intersection property. It can be restated as: there is a forest F such that its vertices are the edges of H and if we take any vertex-pair of F in which both vertices contain a given vertex v of H, then there is a (unique) path in F between these pair on which every vertex contain v as well.

Maximum Cardinality Search on hypergraphs

Number the vertices from n to 1 in decreasing order. As the next vertex to number, select any unnumbered vertex in an edge of H containing as many numbered vertices as possible, breaking ties arbitrarily. (See [21].)

Theorem 3.4. *Let H be an acyclic hypergraph. Any ordering generated by a MCS on H can be generated by a MCS on G^H.*

During a MCS of a hypergraph (H-MCS), we call an edge *exhausted* if all vertices contained in it are already numbered and *nonexhausted* otherwise. If R is a nonexhausted edge containing as many numbered vertices as possible and we number a vertex in R, then if R is still nonexhausted it still contains as many numbered vertices as possible. Therefore after selecting a nonexhausted edge having as many numbered vertices as possible, we can number all of its unnumbered vertices consecutively before selecting another edge. This is a consequence and/or a rephrasing of the H-MCS. This way it is called the restricted form of maximum cardinality search (R-MCS).

The H-MCS facilitates testing G^H for chordality and H for conformity.A H-MCS gives an (α) ordering of the vertices of H (or G^H respectively to the above theorem) from n to 1 and in parallel a (β) ordering of the edges of H (we assign the number to an edge as soon as it becomes exhausted) from 1 to k (let k be the number of edges in H). We can extend this β ordering to the vertices as follows: $\beta(v) := min\{\beta(R) \mid R \text{ is an edge in } H,\ v \in R\}$

Note that $i <_\beta j$ implies $i >_\alpha j$. Finally we compute a (γ) ordering of the edges of H as

$$\gamma(R) := max\{\beta(v) | v \in R \text{ and } \beta(v) < \beta(R)\} \ .$$

(If $\beta(v) = \beta(R)$ for all $v \in R$ then $\gamma(R)$ is undefined.)

Theorem 3.5. *A hypergraph H is acyclic if and only if for each $i \in \{1, \dots k\}$ (k is the number of edges in H) and each edge R of H s.t. $\gamma(R) = i$ it is true that $R \cap \{v \in V |\ \beta(v) < i\} \subset \beta'(i)$.*

Maximal weight spanning tree

If the cliques of the graph G are already known, then another construction of a Junction-tree follows from the so-called cluster graph (see [22]) of G: Let the cliques be the nodes. Connect two nodes with an edge if they intersect. Take all pair-wise separators (intersections) between the cliques and assign the cardinality of these as weights to the edges. The so obtained weighted cluster graph usually contains cycles. We can find the **maximal weight spanning tree** of this cluster graph with usual

algorithms of Kruskal, Prim. Wainwright in [22] states that any maximal weight spanning tree (there can be more than one) of the cardinality weighted clique graph is a junction tree (clique tree) for the original graph.

This is a similar idea as when we have a joint distribution, and we want to find a tree-structured graph that defines an MRF over it. In this case one can use the Chow–Liu algorithm of [3]. Based on the empirical probabilities (estimated from the sample) of the vertices and vertex-pairs (edges), the likelihood of the spanning tree over the vertices is maximal if the mutual information between the vertices is maximal. Note that it is also the Kullback–Leibler distance between the joint distribution of the vertex-pairs (edge-distribution) and the one based on independent attachment of the vertices (with their probabilities). As the maximization is over spanning trees, we need the Kruskal and Prim algorithms to find the maximum, and going through all possible spanning trees is time-consuming. The algorithm theoretically applicable to junction trees (with cliques as vertices and separators as edges), but to run the algorithm for all possible clique structures would increase the computational time enormously.

3.3 Belief propagation on junction trees

In 1988, Lauritzen and Spiegelhalter in [14] proposed an alternative approach for computing marginals which (unlike Pearl's method in [16]) applies to any Bayesian network. It is the L–S version of the so-called junction tree algorithm. Sometimes the junction tree algorithm is called Lauritzen–Spigelhalter algorithm. Afterwards Jensen et al. proposed a modification of the Lauritzen–Spiegelhalter method, see [10, 11]. We call this modified version the HUGIN-algorithm since this was implemented in HUGIN, a software tool developed by the same group. The two versions are more or less the same. A detailed comparison can be found in [15].

We apply the algorithm on the nodes of the junction tree together with their separators. However the separators are explicitly used only in the HUGIN version and their initialized potentials are 1.

Let A and B be two consecutive cliques, and S be the separator between them.

Then denoting by * the newly updated potential, the algorithm is:

$$\begin{aligned}
&\text{message of A} && \psi_S^* = \sum_{X_{A\setminus S}} \psi_A && \psi_A^* := \psi_A \\
&\text{B receives the message} && \psi_B^* = \psi_B \cdot \frac{\psi_S^*}{\psi_S} \\
&\text{message of B} && \psi_S^{**} = \sum_{X_{B\setminus S}} \psi_B^* && \psi_B^{**} := \psi_B^* \\
&\text{A receives the message} && \psi_A^{**} = \psi_A^* \cdot \frac{\psi_S^{**}}{\psi_S^*}
\end{aligned}$$

It is however only two lines if we don't store the separators (in other words, we don't calculate with them explicitly) as in the original (L–S) version of the algorithm:

$$\begin{aligned}
&\text{B receives the message of A} && \psi_B^* = \psi_B \cdot \sum_{X_{A\setminus S}} \psi_A && \psi_A^* := \frac{\psi_A}{\sum_{X_{A\setminus S}} \psi_A} \\
&\text{A receives the message of B} && \psi_A^{**} = \psi_A^* \cdot \sum_{X_{B\setminus S}} \psi_B^* && \psi_B^{**} := \psi_B^*
\end{aligned}$$

These equations hold for any state-configurations $\mathbf{x}_A$, $\mathbf{x}_S$, $\mathbf{x}_B$ within the cliques.

This algorithm is actually the so-called **belief propagation algorithm**, or its other name is sum-product algorithm. The local consistency is guaranteed which means that after one back and forth step, we have

$$\begin{aligned}
\sum_{X_{A\setminus S}} \psi_A^{**} &= \sum_{X_{A\setminus S}} \psi_A^* \frac{\psi_S^{**}}{\psi_S^*} = \frac{\psi_S^{**}}{\psi_S^*} \sum_{X_{A\setminus S}} \psi_A^* \\
&= \frac{\psi_S^{**}}{\psi_S^*} \sum_{X_{A\setminus S}} \psi_A = \frac{\psi_S^{**}}{\psi_S^*} \psi_S^* = \psi_S^{**} = \sum_{X_{B\setminus S}} \psi_B^* = \sum_{X_{B\setminus S}} \psi_B^{**} \ .
\end{aligned}$$

So $\sum_{\mathbf{y}\in\mathcal{X}_{B\setminus S}} \psi_B^{**}(\mathbf{x}_S, \mathbf{y}) = \sum_{\mathbf{y}\in\mathcal{X}_{A\setminus S}} \psi_A^{**}(\mathbf{x}_S, \mathbf{y})$ holds, which means *local consistency*.

What happens when we have a tree of cliques instead of only a pair? We can achieve *global consistency*, meaning that all cliques containing variable X_i agree on its marginal $\mathrm{p}_{\mathrm{X_i}}$ by utilizing the junction tree structure and running the messaging passing scheme in an appropriate order.

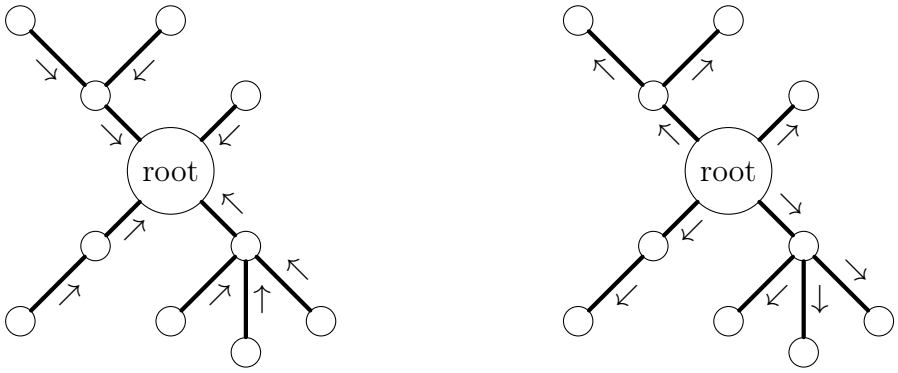
To find all clique potentials, we can run the algorithm for example in the RIP ordering $C_1, \ldots, C_k$ of the cliques. In the first backward run we start at C_k, and via the separators, end at C_1, called the *root*. The message passing is based on the parent-child connections in the junction tree, but the order of the updating (in the first run) is the reversed RIP ordering. Mostly C_1 is called the root, but the usage of this name is inconsistent in the literature. The potential of C_1 obtained in the first run is already the clique potential, so if we are interested only in this, we can

terminate the algorithm here.

To obtain all the clique potentials, we have to run the algorithm again, that is make a forward run in the RIP ordering.

Above we used the knowledge of an appropriate ordering of the cliques. However there is an other approach which takes into consideration the tree structure more soundly.

The protocol in one sentence: a clique can send a message to its neighbour only when it has received messages from all of its other neighbours. This protocol maintains consistency. We designate one (any) node of junction tree as root. First we pass messages inward to the root, and then backward out to the leaves. Making the updates by the RIP ordering looks more linear, but mainly because of the multiplicity of some separators, it actually works the same way. Fig. 3.3 shows an example, to have a better understanding of the two stages.



(message receiving) (message sending)

Figure 3.3: The two main stages of the belief propagation on a more complex junction tree with a designated root

At the end we have the marginals for every cliques, and for the separators, if we stored those too. In the L–S version we can compute the marginal for a single variable from any clique marginal that contains the variable. Since it is more efficient to compute the marginal from a smaller clique, we will do so from a smallest clique that contains the variable. In the HUGIN version the separator potentials are also stored, so we can search for a smallest separator which contains the variable as well. Of course, this strategy ignores the computational cost of identifying a smallest clique (and/or separator).

Evidences

Note that authors differ on what they call evidences. Lauritzen in [13] refers to the message receiving/sending process as evidence absorption/distribution, others however by evidence mean the a priori knowledge of the outcome of some variables

of the examined probability space. The common way to deal with these evidences is the following: at every node where we have observation of a variable (from data), we take appropriate slice of the potential.

For example we know that $X_i = x_i$, call it evidence e. In this case we take all cliques which contain X_i and modify their initial potentials to:

$$\phi_C^e(\mathbf{x}_{C\setminus\{i\}}) := \phi_C(\mathbf{x}_{C\setminus\{i\}}, \mathbf{x}_i) \quad \text{, where} \quad \phi_C^e : \mathcal{X}_{C\setminus\{i\}} \to \mathbb{R} \ .$$

This way we reduce the original graph. Usually the models are built in a way that evidences occur only at simplicial nodes. Therefore the graph will not fall apart or have a drastically new structure by this reduction.

3.3.1 A stylized example

Now comes an example, to see a hypothetical application of the algorithm. The following textbook example appears in many places, for more details see [13] or [14].

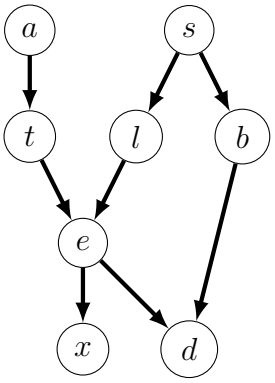
We assume the directed graph structure of Fig. 3.4 (an assumed casual network based on some expert knowledge) of the binary variables $\{a, s, t, l, e, b, x, d\}$. Their output is yes or no. It is a stylized example based on a really complex database. We also assume that this is a graphical representation of a Bayesian network, therefore the conditional independence structure of the example is encoded in this, see Sec. 2.1.

On the second picture we applied moralization and included an additional edge (the red one) to achieve a triangulated graph representation. A Maximal Cardinality Search gave us, starting the algorithm from vertex a, the ordering in the subscript. Based on this we found the cliques and ordered them, now they fulfill the running intersection property. This is a on the third graph.

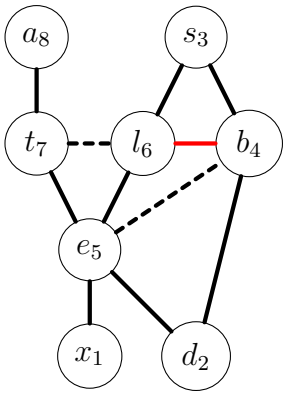
The other input for the algorithm is a conditional probability table, for example we know the values of the below functions:

$$\underbrace{\mathrm{p_a},\ \mathrm{p_{t|a}}}_{\phi_{C_1}},\ \underbrace{\mathrm{p_s},\ \mathrm{p_{l|s}},\ \mathrm{p_{b|s}}}_{\phi_{C_4}},\ \underbrace{\mathrm{p_{e|lt}}}_{\phi_{C_2}},\ \underbrace{\mathrm{p_{x|e}}}_{\phi_{C_6}},\ \underbrace{\mathrm{p_{d|eb}}}_{\phi_{C_5}} \ .$$

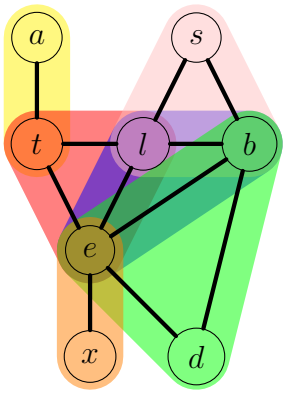
These are known (conditional) probabilities, or can be estimated empirically. Here we already assigned some potentials to multiplication of given (conditional) probabilities. To this table it was easy to assign most of the potentials. However we did not assigned anything to ϕ_{C_3}. In a case like this, it can be any non-zero constant function, e.g. one. Based on these we defined our potential functions which satisfy



(the initial casual network)



(moral graph with fill-in edge + possible MCS-ordering)



$C_1 = \{a, t\}$
$C_2 = \{t, l, e\}$
$C_3 = \{l, e, b\}$
$C_4 = \{l, b, s\}$
$C_5 = \{e, b, d\}$
$C_6 = \{e, x\}$

(the clique structure in the RIP order)

- a: visit to Asia? (y/n)
- s: smoking? (y/n)
- t: tuberculosis? (y/n)
- l: lung cancer? (y/n)
- e: tuberculosis or lung cancer? (y/n)
- b: bronchitis? (y/n)
- x: positive X-ray? (y/n)
- d: dyspnoea? (y/n)

Figure 3.4: The L-S example

the undirected factorization property of Def. 2.6:

$$\mathrm{p_V} = \frac{1}{Z}\prod_{i=1}^{6}\phi_{C_i}, \quad \text{where} \quad Z = \sum_{V}\prod_{i=1}^{6}\phi_{C_i}\ .$$

Now $Z = 1$ since we started with a directed factorization of the distribution, see Def. 2.2. Our goal is to modify these potentials to get the marginal probabilities. The belief propagation goes as follows if C_1 is the root:

- C_1 wants to send its message, but will not do it until it receives it from C_2;
- C_2 behaves the same way, it waits for C_3;
- C_3 has two other neighbors, so it waits for C_4 and C_5 as well;
- C_4 does not have other neighbors, so it can already send its message;

- C_5 however waits for C_6;
- C_6 is like C_4, so it can send its message right away.

The process works based on the junction tree structure, see Fig. 3.5 where it can be followed. Thus the two run of the message-passing goes as:

1. message receiving: $C_6 \to C_5 \to C_3,\ C_4 \to C_3,\ C_3 \to C_2 \to C_1$
2. message sending: $C_1 \to C_2 \to C_3,\ C_3 \to C_4,\ C_3 \to C_5 \to C_6$

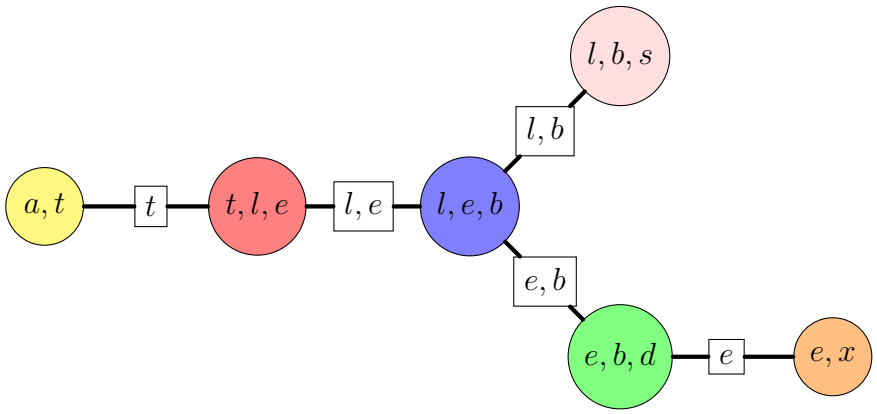


Figure 3.5: Junction tree structure based on the MCS-numbering

We used the HUGIN version of the algorithm. First we checked the results by taking the (conditional) probabilities as parametric functions, see Tab. 3.1. Note that here we extensively used the Bayes-theorem, and conditional independence assumptions which can be read from the original BN with e.g. the algorithm of Sec. 2.1.1. This way we got more compact forms for some potentials and at the end the results are clearer.

Tab. 3.3 at the end of this section, shows the numerical results. To keep the table readable, some trivial steps are omitted. We used an easier notation for the state spaces: $i\hat{j}$ means that $i = \text{yes}$ and $j = \text{no}$. Others accordingly. These numbers are based on the stylized conditional probability table of Tab. 3.2, which appeared in [14]. Here we used the notation: $\mathrm{p}_{\mathrm{i}|\hat{\mathrm{j}}} := \mathrm{Prob}\,(i = \text{yes} \mid j = \text{no})$.

A	ϕ_A	ϕ_A^*	ϕ_A^{**}
$S_1 = \emptyset$	1	1	1
$C_1 = \{a, t\}$	$\mathrm{p_a} \cdot \mathrm{p_{t\|a}} = \mathrm{p_{at}}$	$\mathrm{p_{at}} \cdot 1 = \mathrm{p_{at}}$	$\mathrm{p_{at}}$
$S_2 = \{t\}$	1	$\sum_{l,e} \mathrm{p_{el\|t}} = 1$	$\sum_a \mathrm{p_{at}} = \mathrm{p_t}$
$C_2 = \{t, l, e\}$	$\mathrm{p_{e\|lt}}$	$\mathrm{p_{e\|lt}} \cdot \mathrm{p_l} / 1 = \mathrm{p_{el\|t}}$	$\mathrm{p_{el\|t}} \cdot \mathrm{p_t} / 1 = \mathrm{p_{elt}}$
$S_3 = \{l, e\}$	1	$\sum_b \mathrm{p_{lb}} = \mathrm{p_l}$	$\sum_t \mathrm{p_{elt}} = \mathrm{p_{el}}$
$C_3 = \{l, e, b\}$	1	$(1 \cdot 1/1) \cdot \mathrm{p_{lb}} / 1 = \mathrm{p_{lb}}$	$\mathrm{p_{lb}} \cdot \mathrm{p_{el}} / \mathrm{p_l} = \mathrm{p_{leb}}$
$S_4 = \{l, b\}$	1	$\sum_s \mathrm{p_{lbs}} = \mathrm{p_{lb}}$	$\sum_e \mathrm{p_{leb}} = \mathrm{p_{lb}}$
$C_4 = \{l, b, s\}$	$\mathrm{p_s} \cdot \mathrm{p_{l\|s}} \cdot \mathrm{p_{b\|s}} = \mathrm{p_s} \cdot \mathrm{p_{lb\|s}} = \mathrm{p_{lbs}}$	$\mathrm{p_{lbs}}$	$\mathrm{p_{lbs}} \cdot \mathrm{p_{lb}} / \mathrm{p_{lb}} = \mathrm{p_{lbs}}$
$S_5 = \{e, b\}$	1	$\sum_d \mathrm{p_{d\|eb}} = 1$	$\sum_l \mathrm{p_{leb}} = \mathrm{p_{eb}}$
$C_5 = \{e, b, d\}$	$\mathrm{p_{d\|eb}}$	$\mathrm{p_{d\|eb}} \cdot 1/1 = \mathrm{p_{d\|eb}}$	$\mathrm{p_{d\|eb}} \cdot \mathrm{p_{eb}} / 1 = \mathrm{p_{deb}}$
$S_6 = \{e\}$	1	$\sum_x \mathrm{p_{x\|e}} = 1$	$\sum_{b,d} \mathrm{p_{deb}} = \mathrm{p_e}$
$C_6 = \{e, x\}$	$\mathrm{p_{x\|e}}$	$\mathrm{p_{x\|e}}$	$\mathrm{p_{x\|e}} \cdot \mathrm{p_e} / 1 = \mathrm{p_{xe}}$

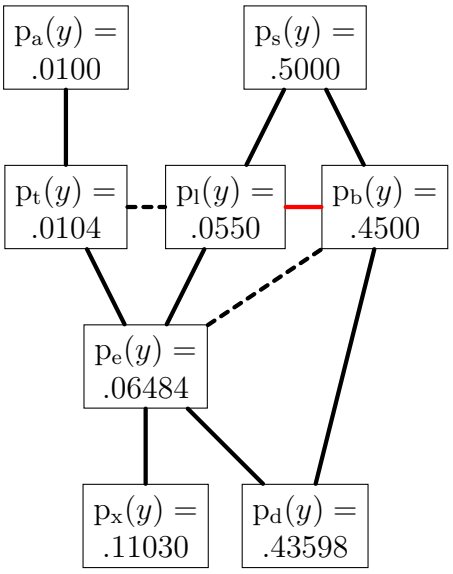
Table 3.1: Symbolic results

a:	$\mathrm{p_a}$	=.01	b:	$\mathrm{p_{b\|s}}$	=.60
t:	$\mathrm{p_{t\|a}}$	=.05		$\mathrm{p_{b\|\hat{s}}}$	=.30
	$\mathrm{p_{t\|\hat{a}}}$	=.01	s:	$\mathrm{p_s}$	=.50
l:	$\mathrm{p_{l\|s}}$	=.10	d:	$\mathrm{p_{d\|eb}}$	=.90
	$\mathrm{p_{l\|\hat{s}}}$	=.01		$\mathrm{p_{d\|e\hat{b}}}$	=.70
e:	$\mathrm{p_{e\|lt}}$	=1		$\mathrm{p_{d\|\hat{e}b}}$	=.80
	$\mathrm{p_{e\|l\hat{t}}}$	=1		$\mathrm{p_{d\|\hat{e}\hat{b}}}$	=.10
	$\mathrm{p_{e\|\hat{l}t}}$	=1	x:	$\mathrm{p_{x\|e}}$	=.98
	$\mathrm{p_{e\|\hat{l}\hat{t}}}$	=0		$\mathrm{p_{x\|\hat{e}}}$	=.05

Table 3.2: Fictional conditional probability table

After the algorithm finishes, we can easily calculate marginals for an arbitrary vertex. On Fig. 3.6 we can see the marginal probabilities for the variables. Again, these are binary variables, so we just denoted the probability of positive (yes) output. Next to the graph, for every variable, we denoted the smallest clique or separator which induce the marginal. Note that if we would have got the results from the L–S version of the algorithm, then we could not use the separators in this marginalization step, since those potential would not have been stored. This shows the computational advantage of the HUGIN version.

We also validated the results with an *R* script, where we used the *gRain* package of Søren Højsgaard, see [9].



- $C_1 \to p_a$ **
- $S_2 \to p_t$
- S_3 or $S_4 \to p_l$
- $S_6 \to p_e$
- S_4 or $S_5 \to p_b$
- $C_4 \to p_s$ **
- $C_5 \to p_d$
- $C_6 \to p_x$

**Note that these marginals already appeared in the conditional probability table.

Figure 3.6: Marginal probabilities of the variables

A	$\mathcal{X}_A$	ϕ_A	ϕ_A^*	ϕ_A^{**}
S_1	$\emptyset$	1	1	1
C_1	at	.0005	$.0005 \times 1 = .0005$	.0005
	$a\hat{t}$	.0095	$.0095 \times 1 = .0095$	.0095
	$\hat{a}t$	.0099	.0099	.0099
	$\hat{a}\hat{t}$	.9801	.9801	.9801
S_2	t	1	$.055 + 0 + .945 + 0 = 1$	$.0005 + .0099 = .0104$
	$\hat{t}$	1	$.055 + 0 + 0 + .945 = 1$	$.0095 + .9801 = .9896$
C_2	tle	1	$1 \times .055 = .055$	$.055 \times .0104/1 = .00057$
	$tl\hat{e}$	0	$0 \times .055 = 0$	$0 \times .0104/1 = 0$
	$t\hat{l}e$	1	$1 \times .945 = .945$	$.945 \times .0104/1 = .00983$
	$t\hat{l}\hat{e}$	0	0	0
	$\hat{t}le$	1	.055	$.055 \times .9896/1 = .05443$
	$\hat{t}l\hat{e}$	0	0	0
	$\hat{t}\hat{l}e$	0	0	0
	$\hat{t}\hat{l}\hat{e}$	1	.945	$.945 \times .9896/1 = .93517$
S_3	le	1	$.0315 + .0235 = .055$	$.00057 + .05443 = .055$
	$l\hat{e}$	1	$.0315 + .0235 = .055$	$0 + 0 = 0$
	$\hat{l}e$	1	$.4185 + .5265 = .945$	$.00983 + 0 = .00983$
	$\hat{l}\hat{e}$	1	$.4185 + .5265 = .945$	$0 + .93517 = .93517$
C_3	leb	1	$1 \times 1 \times .0315 = .0315$	$.0315 \times .055/.055 = .0315$
	$le\hat{b}$	1	$1 \times 1 \times .0235 = .0235$	$.0235 \times .055/.055 = .0235$
	$l\hat{e}b$	1	$1 \times 1 \times .0315 = .0315$	$.0315 \times 0/.055 = 0$

	$l\hat{e}\hat{b}$	1	$1 \times 1 \times .0235 = .0235$	$.0235 \times 0/.055 = 0$
	$\hat{l}eb$	1	$1 \times 1 \times .4185 = .4185$	$.4185 \times .00983/.945 = .00435$
	$\hat{l}e\hat{b}$	1	$1 \times 1 \times .5265 = .5265$	$.5265 \times .00983/.945 = .00548$
	$\hat{l}\hat{e}b$	1	$1 \times 1 \times .4185 = .4185$	$.4185 \times .93517/.945 = .41415$
	$\hat{l}\hat{e}\hat{b}$	1	$1 \times 1 \times .5265 = .5265$	$.5265 \times .93517/.945 = .52102$
S_4	lb	1	$.0300 + .0015 = .0315$	$.0315 + 0 = .0315$
	$l\hat{b}$	1	$.0200 + .0035 = .0235$	$.0235 + 0 = .0235$
	$\hat{l}b$	1	$.2700 + .1485 = .4185$	$.00435 + .41415 = .4185$
	$\hat{l}\hat{b}$	1	$.1800 + .3465 = .5265$	$.00548 + .52102 = .5265$
C_4	lbs	.0300	.0300	$.0300 \times .0315/.0315 = .0300$
	$lb\hat{s}$	.0015	.0015	$.0015 \times .0235/.0235 = .0015$
	$l\hat{b}s$	.0200	.0200	.0200
	$l\hat{b}\hat{s}$	.0035	.0035	.0035
$\vdots$				
$\vdots$				
	$\hat{l}bs$	.2700	.2700	.2700
	$\hat{l}b\hat{s}$	.1485	.1485	.1485
	$\hat{l}\hat{b}s$	.1800	.1800	.1800
	$\hat{l}\hat{b}\hat{s}$	.3465	.3465	.3465
S_5	eb	1	$.9 + .1 = 1$	$.0315 + .00435 = .03585$
	$e\hat{b}$	1	$.7 + .3 = 1$	$.0235 + .00548 = .02898$
	$\hat{e}b$	1	$.8 + .2 = 1$	$0 + .41415 = .41415$
	$\hat{e}\hat{b}$	1	$.1 + .9 = 1$	$0 + .52102 = .52102$
C_5	ebd	.9	$.9 \times 1/1 = .9$	$.9 \times .03585/1 = .03227$
	$eb\hat{d}$	.1	$.1 \times 1/1 = .1$	$.1 \times .03585/1 = .00359$
	$e\hat{b}d$	.7	.7	$.7 \times .02898/1 = .02029$
	$e\hat{b}\hat{d}$	.3	.3	$.3 \times .02898/1 = .00869$
	$\hat{e}bd$	.8	.8	.33132
	$\hat{e}b\hat{d}$	.2	.2	.08284
	$\hat{e}\hat{b}d$	.1	.1	.05210
	$\hat{e}\hat{b}\hat{d}$	.9	.9	.46892
S_6	e	1	$.98 + .02 = 1$	$.03227 + .00359 + .02029 + .00869 = .06484$
	$\hat{e}$	1	$.05 + .95 = 1$	$.33132 + .08284 + .05210 + .46892 = .93516$
C_6	ex	.98	.98	$.98 \times .06484/1 = .06354$
	$e\hat{x}$	.02	.02	$.02 \times .06484/1 = .00130$
	$\hat{e}x$	.05	.05	$.05 \times .93516/1 = .04676$

	$\hat{e}\hat{x}$	.95	.95	$.95 \times .93516/1 = .88840$

Table 3.3: Numerical results

3.4 Decomposable Gaussian graphical models

Decomposability has some advantages in Gaussian framework as well. If the Gaussian graphical model is decomposable (its concentration graph G is decomposable), then the cliques form a junction tree structure as before. Again denote with $\mathcal{C}$ the set of the cliques and with $\mathcal{S}$ the set of their separators in a junction tree of G.

In such case a factorization of the density is achievable:

$$f(\mathbf{x}) = \frac{\prod_{j=1}^{k} f_{C_j}(\mathbf{x}_{C_j})}{\prod_{j=2}^{k} f_{S_j}(\mathbf{x}_{S_j})} = \frac{\prod_{C\in\mathcal{C}} f_C(\mathbf{x}_C)}{\prod_{S\in\mathcal{S}} f_S(\mathbf{x}_S)^{\nu(S)}} , \qquad \mathbf{x} \in \mathbb{R}^n . \tag{3.1}$$

In the above equation the marginalization over a subset of variables is in line with the discrete case, just in this case instead of summation we integrate.

There are also exact tests for decomposable Gaussian models (see [13]). The ML-estimator of the concentration matrix $\boldsymbol{K}$ can be calculated based on the product moment estimators applied for subsets of the variables, corresponding to the cliques and separators. First introduce the simpler form for $\boldsymbol{K}$, see [13]:

$$\boldsymbol{K} = \boldsymbol{\Sigma}^{-1} = \sum_{C\in\mathcal{C}} [\boldsymbol{K}_C]^V - \sum_{S\in\mathcal{S}} [\boldsymbol{K}_S]^V = \sum_{C\in\mathcal{C}} [\boldsymbol{\Sigma}_C^{-1}]^V - \sum_{S\in\mathcal{S}} [\boldsymbol{\Sigma}_S^{-1}]^V ,$$

where $[\boldsymbol{K}_C]^V$ denotes the $n \times n$ matrix containing the entries of $\boldsymbol{K}$ in the $|C| \times |C|$ block corresponding to C, and otherwise zeros. Further,

$$|\boldsymbol{\Sigma}| = \frac{\prod_{C\in\mathcal{C}} |\boldsymbol{\Sigma}_C|}{\prod_{S\in\mathcal{S}} |\boldsymbol{\Sigma}_S|} .$$

Let m be the sample size for the underlying n-variate normal distribution, and assume that $m > n$. For any clique $C \in \mathcal{C}$ let $[\mathbf{S}_C]^V$ denote m times the empirical covariance matrix corresponding to the variables $\{X_i : i \in C\}$ complemented with zero entries to have an $n \times n$ (symmetric, positive semidefinite) matrix. Likewise for any separator $S \in \mathcal{S}$ in the junction tree, let $[\mathbf{S}_S]^V$ denote m times the empirical covariance matrix corresponding to the variables $\{X_i : i \in S\}$ complemented with zero entries to have an $n \times n$ (symmetric, positive semidefinite) matrix. Then the ML-estimator of the mean vector is the sample average (as usual), while the ML-

estimator of the concentration matrix is

$$\hat{\boldsymbol{K}} = m \left\{ \sum_{C \in \mathcal{C}} [\mathbf{S}_C^{-1}]^V - \sum_{S \in \mathcal{S}} [\mathbf{S}_S^{-1}]^V \right\}, \quad \text{and} \quad |\hat{\boldsymbol{K}}| = m^n \cdot \frac{\prod_{S \in \mathcal{S}} |\mathbf{S}_S|}{\prod_{C \in \mathcal{C}} |\mathbf{S}_C|} \ .$$

Here the structure of $\boldsymbol{K}$ imitates the junction tree structure, through reducible zero patterns, see Def. 3.5. Since decomposable models provide the Markov property through a chain, another factorization also holds:

$$f(\mathbf{x}) = \prod_{i=1}^{k} f_{C_i}(\mathbf{x}_{R_i} \mid \mathbf{x}_{S_i}) \tag{3.2}$$

in the RIP ordering of the cliques, residuals, and separators. According to [6, 19], the same can be done for all members of the *exponential* family.

Conclusion

Now we have a really complex picture about graphical models. We started with assigning graph vertices to random variables of a probability space, and connected them based on conditional independence assumptions. By introducing more and more models we gained a broad knowledge about the Bayesian networks, Markov random fields and Gaussian graphical models. The detailed study of decomposable models and the belief propagation on these can help us when we start to examine even more complex models.

By noticing the many overlaps among the fairly different structures, we can have the impression that it is possible to build more general models. There are already models which unite directed and undirected representations. One of these are the so-called regression graph models, for details see [27]. It is possible to build graphical models around conditional Gaussian distributions. In these spaces Gaussian and discrete variables appear mixed. Lauritzen in [13] writes about these as well.

Our long-term goal is to build new models and algorithms for more general mixed models by utilizing the junction tree structure, and by applying non-parametric statistical methods. We hope that we can contribute to the theory and application of graphical models.

Bibliography

[1] Bolla, M. and Krámli, A. *Statisztikai következtetések elmélete*. Typotex, 2005.

[2] Chandran, L.S. et al. "Generating and characterizing the perfect elimination orderings of a chordal graph". In: *Theoretical Computer Science* 307.2 (2003), pp. 303–317.

[3] Chow, C. K. and Liu, C. N. "Approximating discrete probability distributions with dependece trees". In: *IEEE Trans. Inf. Theory* IT-14 (1968), pp. 452–467.

[4] Csiszár, I. and Shields, P. "Information Theory and Statistics: A Tutorial". In: *Foundations and Trends in Communications and Information Theory* 1.4 (2004).

[5] Darroch, J. N., Lauritzen, S. L., and Speed, T. P. "Markov Fields and Log-Linear Interaction Models for Contingency tables". In: *The Annals of Statistics* 8.3 (1980), pp. 522–539.

[6] Dempster, A. P. "Covariance selection". In: *Biometrics* 28 (1972), pp. 157–175.

[7] Geiger, D. and Pearl, J. "On the Logic of Causal Models". In: *Uncertainty in Artificial Intelligence*. Ed. by Shachter, R. D. et al. Vol. 9. Machine Intelligence and Pattern Recognition. North-Holland, 1990, pp. 3–14.

[8] Hammersley, J. M. and Clifford, P. E. "Markov random fields on finite graphs and lattices". In: Unpublished manuscript (1971).

[9] Højsgaard, Søren. "Graphical Independence Networks with the gRain Package for R". In: *Journal of Statistical Software* 46.10 (2012), pp. 1–26.

[10] Jensen, F. V., Lauritzen, S., and Olesen, K. G. "Bayesian Updating in Causal Probabilistic Networks by Local Computations". In: *Computational Statistics Quarterly* 5 (Jan. 1990), pp. 269–282.

[11] Jensen, F. V., Olesen, K. G., and Andersen, S. K. "An algebra of bayesian belief universes for knowledge-based systems". In: *Networks* 20.5 (Aug. 1990), pp. 637–659.

[12] Koller, D. and Friedman, N. *Probabilistic Graphical Models. Principles and Techniques*. MIT Press, 2009.

[13] Lauritzen, S. L. *Graphical Models*. Oxford University Press, 1995.

[14] Lauritzen, S. L. and Spiegelhalter, D. "Local computations with probabilities on graphical structures and their application in expert systems". In: *J. R. Statist. Soc. B* 50 (1988), pp. 157–224.

[15] Lepar, V. and Shenoy, P. P. "A Comparison of Lauritzen-Spiegelhalter, Hugin, and Shenoy-Shafer Architectures for Computing Marginals of Probability Distributions". In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*. Morgan Kaufmann, 1998, pp. 328–337.

[16] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[17] Rose, D. J. "Triangulated graphs and the elimination process". In: *Journal of Mathematical Analysis and Applications* 32.3 (1970), pp. 597–609.

[18] Rudas, T. "Kontingencia táblák loglineáris elemzése". In: *Többváltozós statisztikai analízis*. Ed. by Móri, F. T. and Székely, J. G. Budapest: Műszaki Könyvkiadó, 1986, pp. 191–207.

[19] Sundberg, R. "Some results about decomposable (or Markov-type) models for multidimensional contingency tables: distribution of marginals and partitioning of tests". In: *Scandinavian Journal of Statistics* 2 (1975), pp. 71–79.

[20] Tarjan, R. E., Rose, D. J., and Lueker, G. S. "Algorithmic aspects of vertex elimination on graphs". In: *SIAM Journal of Computing* 5 (1976), pp. 266–283.

[21] Tarjan, R. E. and Yannakakis, M. "Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduce acyclic hypergraphs". In: *Siam J. Computing* 13 (1984), pp. 566–579.

[22] Wainwright, M. J. "Graphical Models and Message-Passing Algorithms: Some Introductory Lectures". In: *Mathematical Foundations of Complex Networked Iformation Systems, Lecture Notes in Mathematics*. Ed. by Fagnani, F. et al. Springer, 2015, p. 2141.

[23] Wainwright, M. and Jordan, M. I. "Graphical models, exponential families, and variational inference". In: *Foundations and Trends in Machine Learning* 1.1-2 (2008), pp. 1–305.

[24] Wermuth, N. "Linear recursive equations, covariance selection, and path analysis". In: *J. Amer. Stat. Assoc.* 75 (1980), pp. 963–972.

[25] Wermuth, N. "Traceable regressions". In: *International Statistical Review* 80.3 (2012), pp. 415–438.

[26] Wermuth, N. "Graphical Markov models, unifying results and their interpretation". In: *Wiley Stats*. Ed. by Balakrishnan, N. et al. Wiley, 2015.

[27] Wermuth, N. and Cox, D. R. "Graphical Markov models: Overview". In: *International Encyclopedia of the Social and Behavioral Sciences, 2nd ed.* Ed. by Wright, J. Oxford, 2015, pp. 341–350.