Funded by the
European Union

The Tensor Network Approach towards Efficient and Explainable AI

Alex Goessmann

Technical report on the ENEXA project at DATEV eG

March 17, 2025

Abstract: While tensors appear naturally in artificial intelligence as factored representations of systems, their decompositions into networks improve the efficiency and explainability of several approaches. Since the curse of dimensionality prevents feasible generic representations and reasoning, logical and probabilistic reasoning focuses on tradeoffs between efficiency and generality. In this work we present these tradeoffs based on the tensor network formalism and formulate feasible reasoning algorithms involving tensor network contractions. We review the classical logical and probabilistic approaches to reasoning in the first part and develop applications in neuro-symbolic AI in the second part. In the third part we investigate in more detail schemes to exploit tensor network contractions for calculus.

Contents

Introduction

Artificial intelligence is a long-standing dream, which has in recent years received enourmous attention, driven by breakthroughs in large language models. Among the key priorities towards an economic and trustworthy usage remain the creation of efficient and the explainable models.

Instead of post-hoc explainability of a models inference given specific data, our aim in this work is the intrinsic human understandability of a model. We are motivated by the theory of logic, which formalization of human thoughts serves as an interface between mechanized reasoning on a machine and human understandability. Having established this advanced form of explainability enables novel forms of human interactions with a model based on verbalizations, manipulations and guarantees on the models inference output.

The desire of an efficient model originates more from an economic perspective on the realizability of a model and its power consumption. Tensors appear naturally as representations of a system with multiple variables, both in logical and probabilistic approaches towards artificial intelligence. However, already for moderate numbers of variables, the curse of dimensionality prevents a typical machines memory to store a generic representation. The careful design of representation formats is therefore a necessary task to avoid the exponential increase of storage demands and balance the expressivity and the efficiency of representation formats.

We in this work exploit the formalism of tensor networks in the creation of efficient representation schemes. The chosen tensor network formats are motivated from explainable learning architectures and provide a synergy between the aims of efficiency and explainability. Tensor networks appear as the natural numerical structures in probabilistic graphical models and logical knowledge bases. After presenting the probabilistic and logical approaches based on the tensor network formalism we develop novel applications schemes towards neuro-symbolic artificial intelligence.

1 Background

Before presenting an overview over the contents, we further motivate this work based on the broach approaches towards artificial intelligence and more recent developments.

1.1 Classical Approached towards AI

We start with ontological commitments in the description of a system and follow the book [RN21] distinguishing atomic, factored and structured representations. While in atomic representation, the states of a systems are enumerated and represented in a single variable, factored representations describe a systems state based on a collection of variables. In the tensor formalism, each state of a system corresponds with a coordinate of a representing tensor. The order of the tensor coincides therefore with the number of variables in a system. In an atomic representation, where there is a single coordinate, each state corresponds with a coordinate of the representing vector being a tensor of order one. Having a factored representation with two variables requires order two tensors or matrices, where a coordinate is specified by a row and a column index. Given larger numbers of coordinates now extends this representation picture to tensors of larger orders, which have more abstract axes besides rows and columns. The generalization of the atomic representation to a factored system thus corresponds with the generalization of vectors towards matrices and tensors of larger orders. Along this line, we can always transform a factored representation of a system to an atomic one, just by enumerating the states of the factored system and interpreting them by a single variable. This amounts to the flattening of a representing tensor to a

vector. However, by doing so, we would loose much of the structure of the representation, which we would like to exploit in reasoning processes.

A more generic representation of systems are structured representation. Structured representations involve objects of differing numbers and relations between them. As a consequence the numbers of variables can differ depending on the state of a system. This poses a challenge to the tensor representation, since a fixed number of variables is required to motivate a tensor space of representations. There are approaches to circumvent these difficulty by the development of template models such as Markov Logic Networks [RD06], which are instantiated on systems with differing number of objects. We will discuss those in Chapter 9.

In this work we treat discrete systems, where the number of states is finite. One can understand them as a discretization of continuous variables and many results will generalize by the converse limit to the situation of continuous variables.

Besides ontological commitments in the choice of a representation scheme, modelling a system also requires epistemologic commitments, by defining what properties are to be reasoned about. In logical approaches the properties of states are boolean values representing whether a state is consistent with known constraints. Probabilistic approaches assign to the coordinates of the tensors numbers in $[0, 1]$ encoding the probability of a state. Compared with logical approached to reasoning, probabilistic approaches thus bear a more expressive modelling.

1.2 Logic and Explainability in AI

Inductive Logic Programming:

- ILP is a classical task [MD94]
- Amie [Gal+13] is a method of learning Horn clauses using a refinement operator.
- Class Expression Learning [Leh+11] is a more recent approach to assist in the design of reasoning capabilities in Knowledge Graphs. However, problems arise from the expressivity of description logics and the efficient choice of formulas from exponentially large hypothesis sets.
- CEL has therefore recently received further popularity in combination with reinforcement learning [DN21] and neural networks [Kou+22; Kou+23], which are methods searching efficienctly in exponentially large spaces of formulas.

Statistical Relational AI: [GT19]

- Classical combination of logical and probabilistic approaches to reasoning

Knowledge Graphs [Hog+21]

- The advent of large Knowledge Graphs enables explainable reasoning methods on structured data.
- Knowledge Graphs are stored in a sparse format, i.e. only true atoms instead of all + truth label.

1.3 Tensor Networks in AI

Tensor Network formats

- HT Format [HK09]
- CP Format

Tensor Networks as Regressors

- Dynamical Systems learning [Gel+19; Goe+20]

- Supervised learning CITE: Stoudenmire etc.

Tensor Representation of Logics

- Tensor Networks have been applied in the automatization of logic reasoning [SIS17; Sat17] apply Matrix multiplication in reasoning.

- [Nic+16] review over relational machine learning and latent features via matrix embeddings.

Tensor Representation of Knowledge Graphs

- Effective representation of queries

- Usage of tensor networks in embeddings [Yan+15] and using complex extensions [TN17; Tro17]

Tensor Representation of Graphical Models

- Duality of Graphical Models and Tensor Networks: [RS19]

- Expressivity studies [Gla+19]

1.4 Infrastructure of AI

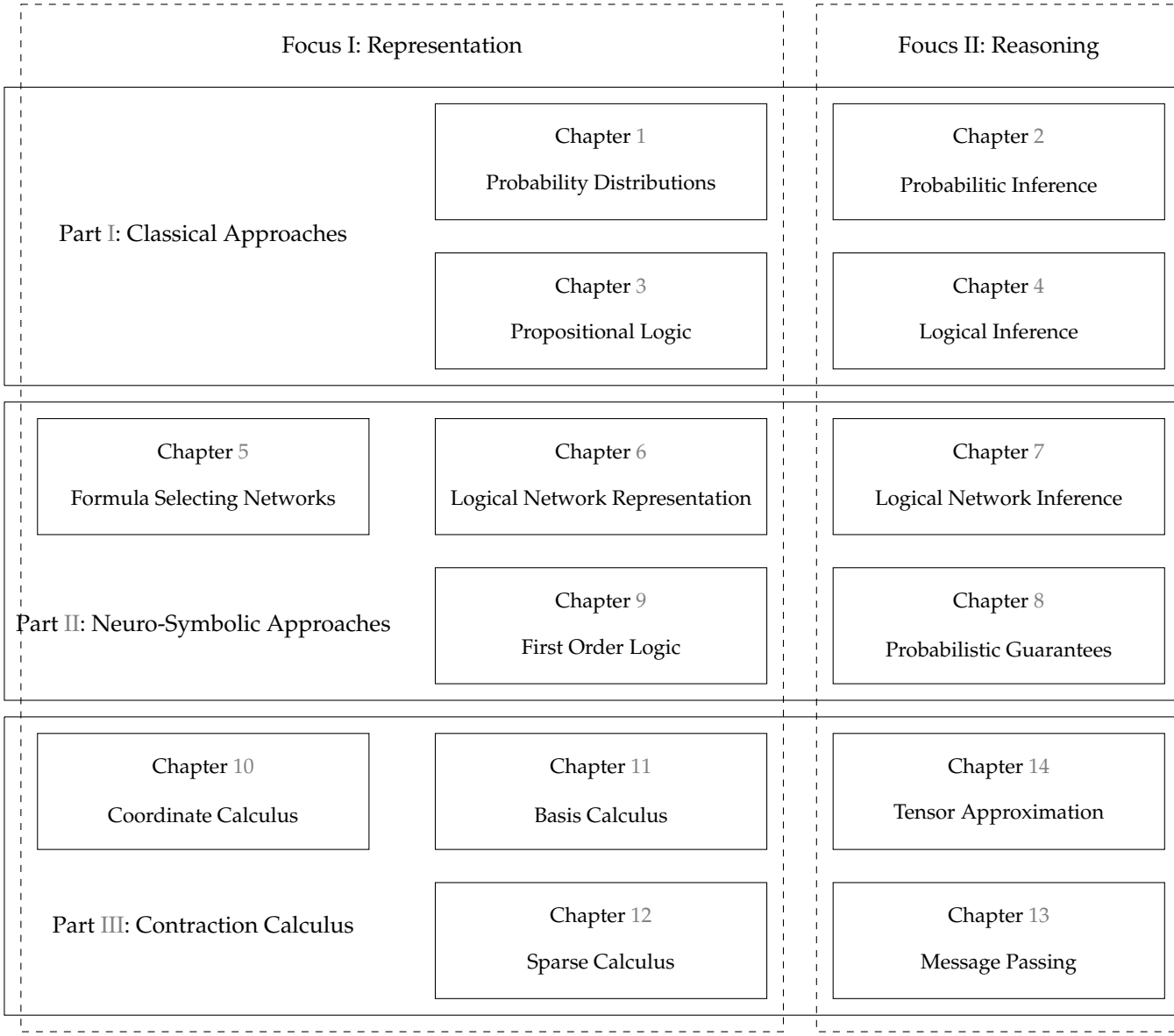
The formalism of tensors and their network decompositions and contractions bears the potential of parallel computations exploited in the AI-dedicated soft and hardware.

- Hardware: TPUs beyond GPUs

- Software: Tensors as basic data structure in TensorFlow, pyTorch etc., storing neural activations and model weights.

2 Structure of the work

The chapters are structured into three parts, and two focuses, as sketched by:



Part I: Classical Approaches

The probabilistic and logical approaches towards artificial intelligence are reviewed in the tensor network formalism.

Tensors appear naturally in

- Logics: Boolean tensors indicating models (propositional case) and interpretation tensors in first order logics
- Probability theory: Truth tables, which are tensors of probabilities for joint distibutions of categorical variables.

Tensor network decompositions as representation schemes appear in

- Logics: Conjunctions of formulas are Hadamard products of the tensor representation of formulas (Coordinate Calculus/ Effective Calculus)

- Probability theory: Graphical models are tensor networks of the factors. Further sparsity schemes apply, when placing restrictions on the structure of each factor.
- Data bases: Relations encoded by lists as storage of nonvanishing coordinates of a relation encoding

Tensor network contractions as reasoning schemes appear in

- Logics: Model counts, used for satisfiablility decisions and entailment
- Probability theory: Marginal probability distributions, extended to conditional probability distributions through normations

Part II: Neuro-Symbolic Approaches

Motivated by the classical approaches we apply the tensor network formalism towards learning and infering neuro-symbolic models.

Neurosymbolic AI

- Required for more advanced AI [Hoc22]
- Add the paradigm of neural computing to logical reasoning
- Potential benefits from Statistical Relational AI [Mar+24]
- Tensor based approaches [CYM20]
- [Bad+22] representation of logic using tensor networks and automated differentiation to optimize.

Tensor Approaches to Neuro-Symbolic AI

- TensorLog [CYM20]
- [Bad+22] representation of logic using tensor networks and automated differentiation to optimize.

In Deep Neural Networks, functions between the input layer and the output layer are decomposed into neurons. Typical neurons are linear transforms with an activation function.

Sparsity means restriction to functions, which are decomposable into a small number of neurons. Approximations of generic functions (see the universal approximation theorems) would require large amounts of neurons. When restricting to functions based on a fixed architecture, we restrict to a certain set of functions called the inductive bias of the architecture.

Part III: Contraction Calculus

The applied schemes of calculus using tensor network contractions are investigated in more detail.

Focus I: Representation

Here we motivate and investigate the efficient representation of tensors based on tensor network decompositions.

Foucs II: Reasoning

We develop schemes to efficiently perform inductive and deductive reasoning based on information stored in decomposed tensor.

Notation and Basic Concepts

We here provide the fundamental definitions of tensors, which are essentiell for the content in Part I and Part II. In Part III we will further investigate the properties of tensors focusing on their contractions.

1 Categorical Variables and Representations

We will in this work investigate systems, which are described by a set of properties, each called a categorical variable. This is called an ontological commitment, since it defines what properties a system has.

Definition 0.1 An atomic representation of a system is described by a categorical variables X taking values x in a finite set

$$[m] \coloneqq \{0, \dots, m-1\}$$

of cardinality m.

We will in this work always notate categorical variables by large literals and indices by small literals, possible with other letters such as X, L, O, J and corresponding values x, l, o, j.

Definition 0.2 A factored representation of a system is a set of categorical variables X_k, where $k \in [d]$, taking values in $[m_k]$.

2 Tensors

Tensors are multiway arrays and a generalization of vectors and matrices to higher orders. We will first provide a formal definition as real maps from index sets enumerating the coordinates of vectors, matrices and larger order tensors.

Definition 0.3 (Tensor) Let there be numbers $m_k \in \mathbb{N}$ for $k \in [d]$ and categorical variables X_k taking their values in $[m_k]$. We call maps

$$T\left[X_0, \dots, X_{d-1}\right] : \bigtimes_{k \in [d]} [m_k] \to \mathbb{R}$$

tensor of order d and leg dimensions $m_0, \dots, m_{d-1}$. Evaluations of these maps at indices $x_0, \dots, x_{d-1}$ are denoted by

$$T\left[X_0 = x_0, \dots, X_{d-1} = x_{d-1}\right] = T\left[X_0, \dots, X_{d-1}\right](x_0, \dots, x_{d-1}) \ .$$

Tensors $T\left[X_0, \dots, X_{d-1}\right]$ are elements of the space

$$\bigotimes_{k \in [d]} \mathbb{R}^{m_k}$$

which is, with the operations of coordinatewise summation and scalar multiplication, a linear

space called a tensor space.

We here introduced tensors in a non-canonical way based on categorical variables assigned to its axis. While coming as syntactic sugar at this point, this will allow us to define contractions without further specification of axes, based on comparisons of shared categorical variables. Especially, this eases the implementation of tensor network contractions without the need to further specify a graph (see Appendix A).

We abbreviate lists $X_0, \ldots, X_{d-1}$ of categorical variables by $X_{[d]}$, that is denote $T[X_0, \ldots, X_{d-1}]$ by $T\left[X_{[d]}\right]$. Occasionally, when the categorical variables of a tensor are clear from the context, we will omit the notation of the variables.

Example 0.4 (Trivial Tensor) The trivial tensor is defined as the map

$$\mathbb{I}\left[X_{[d]}\right] : \bigtimes_{k\in[d]} [m_k] \to \{1\} \subset \mathbb{R}$$

with all coordinates being 1, that is for all $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]} [m_k]$

$$\mathbb{I}\left[X_{[d]} = x_{[d]}\right] = 1\,.$$

$\Diamond$

3 One-hot encodings

We are now ready to provide the link between tensors and states of systems with factored representations. To this end, we define the one-hot encoding of a state, which is a bijection between the states and the basis elements of a tensor space.

Definition 0.5 (One-hot encodings to Atomic Representations) Given an atomic system described by the categorical variable X, we define for each $x \in [m]$ the basis vector $e_x[X]$ by

$$e_x[X = \tilde{x}] = \begin{cases} 1 & \text{if} \quad x = \tilde{x} \\ 0 & \text{else}\,. \end{cases}$$

The one-hot encoding of states $x \in [m]$ of the atomic system described by the categorical variable X is the map

$$e : [m] \to \mathbb{R}^m$$

which maps $x \in [m]$ to the basis vectors $e_x[X]$.

The basis vectors $e_x[X]$ are tensors of order 1 and leg dimension m of the structure

$$e_x[X] = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T\,,$$

where the 1 is at the xth coordinate of the vector.

We have so far described one-hot representations of the states of a single categorical variable, which would suffice to encode the state of an atomic system. In a factored system on the other side, we are dealing with multiple categorical variables.

Definition 0.6 (One-hot encodings to Factored Representations) Let there be a factored

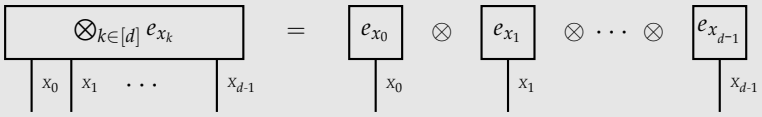
system defined by a tuple $(X_0, \dots, X_{d-1})$ of variables taking values in $\bigtimes_{k\in[d]}[m_k]$. The one-hot encoding of its states is the tensor product of the one-hot encoding to each categorical variables, that is the map

$$e : \bigtimes_{k\in[d]} [m_k] \to \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$$

defined by mapping $x_0, \dots, x_{d-1} = x_{[d]}$ to

$$e_{x_{[d]}} \left[X_{[d]}\right] =: \bigotimes_{k\in[d]} e_{x_k} [X_k] \ .$$

We will call one-hot representations *tensor representations* and depict them as



In Chapter 10 we will investigate the image of e in more detail and show that it is an orthonormal basis of the tensor space $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$.

Remark 0.7 (Flattening of Tensors) The use the tensor product to represent states of factored systems can be motivated by the reduction to atomic systems by enumeration of the states. We have this property reflected in the state encoding of factored systems, since the tensor space $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ is isomorphic to the vector spaces $\mathbb{R}^{\prod_{k\in[d]} m_k}$. This operation is called flattening (or unfolding) of tensors with many axes to tensors of less axes. ◊

4 Contractions

Contractions are the central manipulation operation on sets of tensors. To introduce them, we will develop a graphical illustration of sets of tensors, which we also call tensor networks. In Part III we will further investigate the utility of contractions in representing specific calculations, which demand different encoding schemes.

4.1 Graphical Illustrations

Sets of tensor with categorical variables assigned to each legs implicitly carry a notion of a hypergraph. This perspective is especially useful, when some categorical variables are assigned to axis of multiple tensors, as it will often be the case in the applications considered in this work. Each variable can then be labeled by a node and each tensor as a hyperedge containing the nodes to its axis variables. Let us first formally introduce hypergraphs, which are generalizations of graphs allowing edges to be arbitrary nonempty subsets of the nodes, whereas canonical graphs demand a cardinality of two.

Definition 0.8 A hypergraph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$, where each hyperedge $e \in \mathcal{E}$ is a subset of the nodes $\mathcal{V}$. A directed hypergraph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that each hyperedge $e \in \mathcal{E}$ is the tuple of two disjoint sets $e^{\text{in}}, e^{\text{out}} \subset \mathcal{V}$, that is

$$e = (e^{\text{in}}, e^{\text{out}}) \, .$$

We will use the standard visualization by factor graphs as a diagrammatic illustration of sets of tensors, where tensors are represented by block nodes and each axis assigned with by a categorical variable X_k represented by a node, see Figure 1a). Different simplifications of these factor graph depictions have been evolved in different research fields. In the tradition of graphical models,

which started with the work [Pea88], the categorical variables are highlighted and the tensor blocks just depicted by hyperedges. To depict dependencies with causal interpretations, the edges are further decorated by directions in the depiction of Bayesian networks, see for example [Pea09].

In the tensor network community on the other hand, a simplification scheme highlighting the tensors as blocks and omitting the depiction of categorical variables has been evolved. The variables, or sometimes their index or dimension, are then directly assigned to the lines depicting the axes of the tensor blocks.

Both depiction schemes are simplifications of factor graphs, by highlighting the categorical variables in the depiction in Figure 1b) and the tensors in the depiction in Figure 1c). We in this work will prefer the simplification of the tensor network community, depicted in Figure 1b).

In another interpretation (see [RS19]), both simplification schemes are itself interpret as hypergraphs, which are dual to each other.

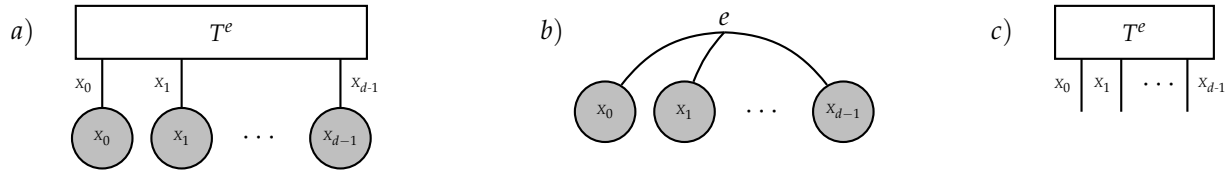
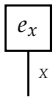


Figure 1: Depiction of Tensors a) As a factor in a factor graph, depicted by a block, and connected to categorical variables assigned to nodes. b) Highlighting only the variable dependencies by a hyperedge connecting the variables X_k to each axis $k \in [d]$. c) Highlighting the tensor by a blockwise notation with axes denoted by open legs represented by the variables X_k.

To depict vector calculus and its generalizations, we will apply the graphical notation (mainly version b) introduced in Chapter 2. Along this line, we represent vectors and their generalization to tensors by blocks with legs representing its indices. The basis vectors being one-hot encodings of states are in this scheme represented by



where $\tilde{x}$ is an indexed represented by an open leg. Assigning x to this index will retrieve the xth coordinate (with value 1), whereas all other assignments will retrieve the coordinate values 0.

Drawing on the interpretation of tensors by hyeredges we can continue with the definition of tensor networks.

Definition 0.9 Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a hypergraph with nodes decorated by categorical variables X_v with dimensions

$$m_v \in \mathbb{N}$$

and hyperedges $e \in \mathcal{E}$ decorated by core tensors

$$T^e\left[X_e\right] \in \bigotimes_{v \in e} \mathbb{R}^{m_v},$$

where we denote by X_e the set of categorical variables X_v with $v \in e$. Then we call the set

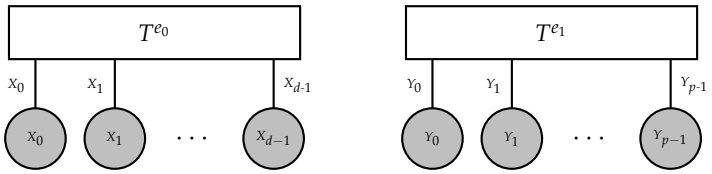
$$\mathcal{T}^{\mathcal{G}}\left[X_{\mathcal{V}}\right] = \left\{T^e\left[X_e\right] : e \in \mathcal{E}\right\}$$

the Tensor Network of the decorated hypergraph $\mathcal{G}$.

Figure 2: Example of a tensor network. a) Hypergraph with edges $e_0 = \{X_0, X_1, X_2\}$, $e_1 = \{X_1, X_2\}$ and $e_2 = \{X_2, X_3\}$ decorated by tensor cores. b) Dual tensor network, depicting a contraction with leaving all variables open.

4.2 Tensor Product

Let us now exploit the developed graphical representations to define contractions of tensor networks. The simplest contraction is the tensor product, which maps a pair of two tensors with distinct variables onto a third tensor and has an interpretation by coordinatewise products. Such a contraction corresponds with a tensor network of two tensors with disjoint variables, depicted as:



Definition 0.10 (Tensor Product) Let there be two tensor

$$T^{e_0}\left[X_{[d]}\right] : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R} \quad \text{and} \quad T^{e_1}\left[Y_{[p]}\right] : \bigtimes_{l\in[r]} [m_l] \to \mathbb{R}$$

with different categorical variables assigned to its axes. Then there tensor product is the map

$$\left\langle T^{e_0}\left[X_{[d]}\right], T^{e_1}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}, Y_{[p]}\right] : \left(\bigtimes_{k\in[d]} [m_k]\right) \times \left(\bigtimes_{l\in[r]} [m_l]\right) \to \mathbb{R}$$

defined for $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]} [m_k]$ and $y_0, \ldots, y_{r-1} \in \bigtimes_{l\in[r]} [m_l]$ as

$$\begin{aligned}&\left\langle T, \tilde{T}\right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}, Y_0 = y_0 \ldots, Y_{p-1} = y_{p-1}\right] \\ &\quad := T^{e_0}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] \cdot T^{e_1}\left[Y_0 = y_0 \ldots, Y_{p-1} = y_{p-1}\right] .\end{aligned}$$

Other popular standard notations of tensor products (see [KB09; Hac12; Cic+15])

$$\left(T \otimes \tilde{T}\right) = \left(T \circ \tilde{T}\right) = \left\langle T^{e_0}\left[X_{[d]}\right], T^{e_1}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}, Y_{[p]}\right] .$$

We will avoid these notations in this work in favor of a consistent notation capable of depicting generic tensor network contractions.

When the tensor $T^{e_1}\left[Y_{[p]}\right]$ coincides with the trivial tensor $\mathbb{I}\left[Y_{[p]}\right]$ (see Example 0.4), we further

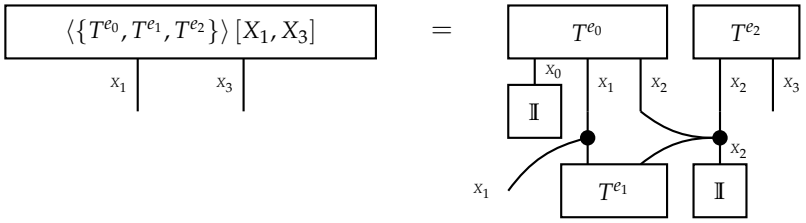


Figure 3: Example of a tensor network contraction of all but the variables X_1, X_3. Contraction of variables can always be depicted by closing the open legs with trivial tensors $\mathbb{I}$ performing index sums.

make a notation convention to omit that tensor, that is

$$\left\langle T^{e_0}\left[X_{[d]}\right], \mathbb{I}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}, Y_{[p]}\right] = \left\langle T^{e_0}\left[X_{[d]}\right]\right\rangle \left[X_{[d]}, Y_{[p]}\right] .$$

4.3 Generic Contractions

Contractions of Tensor Networks $\mathcal{T}^{\mathcal{G}}$ are operations to retrieve single tensors by summing products of tensors in a network over common indices. We will define contractions formally by specifying just the indices not to be summed over.

When some of the variables are not appearing as leg variables, we define the contraction as being a tensor product with the trivial tensor $\mathbb{I}$ carrying the legs of the missing variables.

Definition 0.11 Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network on a decorated hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For any subset $\tilde{\mathcal{V}} \subset \mathcal{V}$ we define the contraction to be the tensor

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] \in \bigotimes_{v \in \tilde{\mathcal{V}}} \mathbb{R}^{m_v}$$

defined coordinatewise by the sum

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}} \in \times_{v \in \mathcal{V}/\tilde{\mathcal{V}}}[m_v]} \left(\prod_{e \in \mathcal{E}} T^e \left[X_e = x_e\right] \right) .$$

We call $X_{\tilde{\mathcal{V}}}$ the open variables of the contraction.

To ease notation, we sometimes omit the set notation by brackets $\{\cdot\}$ and specify the tensors to be contracted with the delimiter "," (see e.g. Example 0.13).

Remark 0.12 (Alternative Notations) Contractions can also denoted by the Einstein summations of the indices along connected edges, understood as scalar product in each subspace. This is as in Def. 0.11, just omitting the sums. We found it useful in this work to do the diagrammatic representation instead, since it offers a better possibility to depict hierarchical arrangements of shared variables. ♢

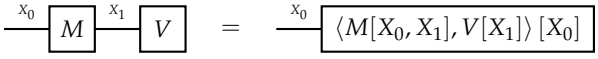
Further notations without usage of axis variables are mode products (see [KB09; Hac12; Cic+15]), often denoted by the operation $\times_n$. With our more generic variable-based notations, we can capture these more specific contractions by coloring the tensor axes, that is assignment of axis variables.

To further gain familiarity with the generic contractions, we show the connection to two more popular examples.

Example 0.13 Matrix Vector Products The matrix vector product is a special case of tensor contractions, where a matrix $M[X_0, X_1]$ shares a categorical variable with a vector $V[X_1]$. When leaving the variable unique to the matrix open we get the matrix vector product as

$$\langle M[X_0, X_1], V[X_1] \rangle [X_0 = x_0] = \sum_{x_1 \in [m_1]} M[X_0 = x_0, X_1 = x_1] \cdot V[X_1 = x_1] .$$

Exploiting the diagramatic tensor network visualization we depict matrix vector by:



◊

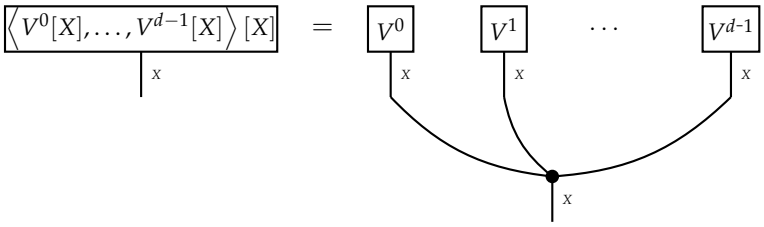
Example 0.14 Hadamard products of vectors A node appearing in arbitrary many hyperedges denotes a Hadamard product of the axis of the respective decorating tensors. To give an example, let $V^k[X] \in \mathbb{R}^m$ be vectors for $k \in [d]$. Their hadamard product is the vector

$$\left\langle \{V^k[X] : k \in [d]\} \right\rangle [X] \in \mathbb{R}^m$$

defined by

$$\left\langle \{V^k[X] : k \in [d]\} \right\rangle [X = x] = \prod_{k \in [d]} V^k[X = x] .$$

In a contraction diagram the Hadamard product is depicted by:



◊

4.4 Decompositions

Tensors can be represented by tensor network decompositions, when the contraction of the network retrieves the tensor.

Definition 0.15 A Tensor Network Decomposition of a tensor $T[X_{\mathcal{V}}]$ is a Tensor Network $\mathcal{T}^{\mathcal{G}}$ such that

$$T[X_{\mathcal{V}}] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}] .$$

We call the hypergraph $\mathcal{G}$ the format of the decomposition.

5 Properties of Tensors

We will often encounter situations, where the coordinates of tensors are in $\{0, 1\} = [2]$.

Definition 0.16 We call a tensor $T\left[X_{[d]}\right]$ boolean, when $\mathrm{im}(T) \subset [2]$, i.e. all coordinates are either 0 or 1.

Directionality represents constraints on the structure of tensors: Summing over outgoing trivializes the tensor.

Definition 0.17 A Tensor

$$T\left[X_{\mathcal{V}}\right] \in \bigotimes_{v\in\mathcal{V}} \mathbb{R}^{m_v}$$

is said to be directed with incoming variables $\mathcal{V}^{\text{in}}$ and outgoing variables $\mathcal{V}^{\text{out}}$, where $\mathcal{V} = \mathcal{V}^{\text{in}} \dot\cup \mathcal{V}^{\text{out}}$, when

$$\langle T\rangle\left[X_{\mathcal{V}^{\text{out}}}\right] = \mathbb{I}\left[X_{\mathcal{V}^{\text{in}}}\right]$$

where $\mathbb{I}\left[X_{\mathcal{V}^{\text{in}}}\right]$ denoted the trivial tensor in $\bigotimes_{v\in\mathcal{V}^{\text{in}}} \mathbb{R}^{m_v}$ which coordinates are all 1.

While by default all legs are outgoing, we can change the direction by normation.

Definition 0.18 A tensor $T\left[X_{\mathcal{V}}\right]$ is said to be normable on $\mathcal{V}^{\text{in}} \subset \mathcal{V}$, if for any $x_{\mathcal{V}^{\text{in}}} \in \bigtimes_{v\in\mathcal{V}^{\text{in}}}[m_v]$ we have

$$\left\langle T\left[X_{\mathcal{V}}\right], e_{x_{\mathcal{V}^{\text{in}}}}\left[X_{\mathcal{V}^{\text{in}}}\right]\right\rangle[\varnothing] > 0\,.$$

The normation of a on $\mathcal{V}^{\text{in}} \subset \mathcal{V}$ normable tensor is the tensor

$$\langle T\left[X_{\mathcal{V}}\right]\rangle\left[X_{\mathcal{V}^{\text{out}}}|X_{\mathcal{V}^{\text{in}}}\right] = \sum_{x_{\mathcal{V}^{\text{in}}}\in\bigtimes_{v\in\mathcal{V}^{\text{in}}}[m_v]} e_{x_{\mathcal{V}^{\text{in}}}}\left[X_{\mathcal{V}^{\text{in}}}\right] \otimes \frac{\left\langle T\left[X_{\mathcal{V}}\right], e_{x_{\mathcal{V}^{\text{in}}}}\left[X_{\mathcal{V}^{\text{in}}}\right]\right\rangle\left[X_{\mathcal{V}^{\text{out}}}\right]}{\left\langle T\left[X_{\mathcal{V}}\right], e_{x_{\mathcal{V}^{\text{in}}}}\left[X_{\mathcal{V}^{\text{in}}}\right]\right\rangle[\varnothing]}$$

where $\mathcal{V}^{\text{out}} = \mathcal{V}/\mathcal{V}^{\text{in}}$.

We will investigate the contractions of directed tensors in Part III, where we show in Theorem 10.11 that normations are directed tensors.

In our graphical tensor notation, we depict directed tensors by directed hyperedges (a), which are decorated by directed tensors (b), for example:

6 Encoding schemes for functions

Tensors are defined here as real-valued functions on the state set of a system described by categorical variables. We provide further schemes to represent functions in order to perform sparse calculus and to handle more generic functions.

6.1 Relational encodings

Let us now show how we can encode maps between factored systems. The scheme is described in more generality and detail (encoding of subsets and relations) in Chapter 11, see Def. 11.6.

Definition 0.19 (Relation encoding of maps between Factored Systems) Let f be a function

$$f : \bigtimes_{k\in[d]}[m_k] \to \bigtimes_{l\in[r]}[m_l]$$

which maps the states of a factored system to variables $X_0, \ldots, X_{d-1}$ to the states of another

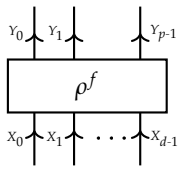
factored system with variables $Y_0, \ldots, Y_{p\text{-}1}$. Then the tensor representation of f is a tensor

$$\rho^f \left[X_0, \ldots, X_{d-1}, Y_0, \ldots, Y_{p\text{-}1} \right] \in \left(\bigotimes_{l \in [p]} \mathbb{R}^{m_l} \right) \otimes \left(\bigotimes_{k \in [d]} \mathbb{R}^{m_k} \right)$$

defined by

$$\rho^f \left[Y_0, \ldots, Y_{p\text{-}1}, X_0, \ldots, X_{d-1} \right] = \sum_{x_0, \ldots, x_{d-1} \in \times_{k \in [d]} [m_k]} e_{f(x_0, \ldots, x_{d-1})} \left[Y_0, \ldots, Y_{p\text{-}1} \right] \otimes e_{x_0, \ldots, x_{d-1}} \left[X_0, \ldots, X_{d-1} \right] .$$

We depict relational encodings by directed tensors:



6.2 Tensor-valued functions

Definition 0.20 (Selection encoding of Maps between Factored Systems) Given a tensor space $\bigotimes_{s \in [n]} \mathbb{R}^{p_s}$ described by categorical variables $L_0, \ldots, L_{n-1}$ and a tensor-valued function

$$f : \bigtimes_{k \in [d]} [m_k] \to \bigotimes_{s \in [n]} \mathbb{R}^{p_s}$$

the selection encoding of f is a tensor

$$\gamma^f \left[X_{[d]}, L_{[n]} \right] \in \left(\bigotimes_{k \in [d]} \mathbb{R}^{m_k} \right) \otimes \left(\bigotimes_{s \in [n]} \mathbb{R}^{p_s} \right)$$

defined by the basis decomposition

$$\gamma^f \left[X_{[d]}, L_{[n]} \right] = \sum_{x_{[d]} \in \times_{k \in [d]} [m_k]} e_{x_0, \ldots, x_{d-1}} \left[X_{[d]} \right] \otimes f(x_0, \ldots, x_{d-1}) [L_{[n]}] .$$

We call these tensor representation of maps selection encodings, since the coordinate of a function f to be processed is selected by another argument to γ^f.

We will provide more detail to the tensor representation of functions in Part III, where we distinguish between embeddings for basis and coordinate calculus.

Contraction equations

We here provide a summary for the application of contractions and normation in the probabilistic and logical reasoning, which will be introduced in Part I. In Chapter 1 we introduce:

- Marginal probabilities (Def. 1.4, The. 1.5)

$$\mathbb{P}\left[X_0\right] = \langle \mathbb{P} \rangle \left[X_0\right]$$

- Conditional probabilities (Def. 1.6, The. 1.7)

$$\mathbb{P}\left[X_0 | X_1\right] = \langle \mathbb{P} \rangle \left[X_0 | X_1\right]$$

- The probability distribution of a Markov Network is (Def. 1.19)

$$\mathbb{P}^{\mathcal{T}^{\mathcal{G}}} = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\mathcal{V} | \varnothing\right]$$

  The partition function of a Markov Networks

$$\mathcal{Z}\left(\mathcal{T}^{\mathcal{G}}\right) = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\varnothing\right]$$

  Bayesian Networks (Def. 1.24), when hypergraph directed and acyclic, such that the decorating tensors are accordingly directed.

Further the following properties are defined by contraction equations:

- X_0 and X_1 are independent when (Def. 1.11, The. 1.12)

$$\langle \mathbb{P} \rangle \left[X_0, X_1\right] = \langle \mathbb{P} \rangle \left[X_0\right] \otimes \langle \mathbb{P} \rangle \left[X_1\right]$$

- X_0 and X_1 are called independent conditioned on X_2 when (Def. 1.13, The. 1.14)

$$\langle \mathbb{P} \rangle \left[X_0, X_1 | X_2\right] = \langle \mathbb{P} \rangle \left[X_0 | X_2\right] \otimes \langle \mathbb{P} \rangle \left[X_1 | X_2\right]$$

In Chapter 3 we introduce:

- Propositional formulas by boolean tensors
- Syntactical representation of formulas corresponding with tensor networks of boolean tensors

Part I

Classical Approaches

1

Probability Distributions

In this chapter we will establish relations between the formalism of tensor networks and basic concepts of probability theory. We will first understand distributions as tensors and connect their marginalizations and conditionings to the tensor operations of contractions and normations. Then we discuss independence assumptions as examples of contraction equations, which lead to tensor network decompositions known as graphical models. We then treat more generic exponential families and investigate their representation as tensor networks.

1.1 Tensor Representation of Distributions

After having discussed how to represent states of factored systems by one-hot encodings, let us now take advantage of these representation by associating properties with these states. Let there be uncertainties of the assignments x_k to the categorical variables X_k of a factored system. We then understand X_k as random variables, which have a joint distribution defined by the uncertainties of the state assignments. To capture these uncertainties we now make use of the one-hot representation of factored systems in Chapter **??**.

Definition 1.1 (Probability Tensor) Let there be a factored system defined by a categorical variable X_k for each $k \in [d]$ taking values in $[m_k]$. A probability distribution over the states of a system in factored representation is a tensor

$$\mathbb{P}\left[X_0, \ldots, X_{d-1}\right] : \bigtimes_{k\in[d]} [m_k] \to [0,1] \subset \mathbb{R}$$

such that

$$\sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[m_k]} \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = 1\,.$$

We notice that there are two conditions for a tensor to be probability tensor. First, the tensor needs to have non-negative coordinates and second, the coordinates need to sum to 1.

The probability tensor to the distribution is an object

$$\mathbb{P}\left[X_0, \ldots, X_{d-1}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$$

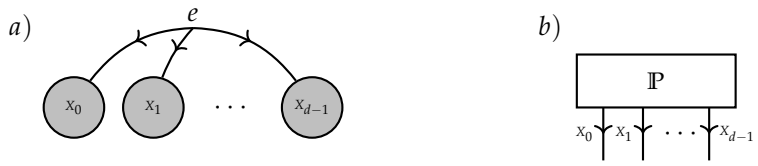
which is the sum over the one-hot encodings (see Lem. 10.1)

$$\mathbb{P}\left[X_0, \ldots, X_{d-1}\right] = \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[m_k]} \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] \cdot e_{x_0,\ldots,x_{d-1}}\left[X_0, \ldots, X_{d-1}\right]\,.$$

The normation condition of probability tensors can be expressed by the contraction equation $1 = \langle \mathbb{P} \rangle [\varnothing]$ since

$$1 = \sum_{x_0,\ldots,x_{d-1}} \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = \sum_{x_0,\ldots,x_{d-1}} \left\langle \mathbb{P}, e_{x_0,\ldots,x_{d-1}} \right\rangle [\varnothing] = \langle \mathbb{P} \rangle [\varnothing]\,.$$

Probability tensors are depicted as



1.1.1 Base measures

From a measure theoretic perspective, probabilities are measurable functions called probability densities, which integrals are 1. In our case of finite dimensional state spaces of factored systems, we implicitly used the trivial tensor $\mathbb{I}\left[X_{[d]}\right]$ as a base measure, which measures subsets of states by their cardinality and is therefore refered to as state counting base measure. The distribution tensors $\mathbb{P}\left[X_{[d]}\right]$ can then be understood as probability densities with respect to this state counting base measure. We in this work will also consider more general base measures $\nu\left[X_{[d]}\right]$, which we restrict to be boolean, that is $\nu\left[X_{[d]} = x_{[d]}\right] \in \{0,1\}$ for all states $x_{[d]}$. When understanding $\mathbb{P}$ as a probability density with respect to ν, any probabilistic interpretation will be through the contraction $\langle \mathbb{P}, \nu\rangle\left[X_{[d]}\right]$ and the normation condition reads as

$$\langle \mathbb{P}, \nu\rangle\left[\varnothing\right] = 1\,.$$

Since we restrict to boolean base measures, the contraction effectively manipulates the tensor $\mathbb{P}$ by setting the coordinates $\mathbb{P}\left[X_{[d]} = x_{[d]}\right]$ to zero, when $\nu\left[X_{[d]} = x_{[d]}\right] = 0$. Therefore, multiple tensors $\mathbb{P}$ will have the same proabilistic interpretation, when $\nu\left[X_{[d]}\right] \neq \mathbb{I}\left[X_{[d]}\right]$. To avoid this ambiguity, we introduce the notation of representability with respect to a base measure ν, by demanding that such coordinates are zero.

Definition 1.2 We say that a probability distribution $\mathbb{P}$ is representable with respect to a boolean base measure ν, if for all $x_{[d]}$ with $\nu\left[X_{[d]} = x_{[d]}\right] = 0$ we have $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0$.

When a probability distribution $\mathbb{P}$ is representable with respect to a boolean base measure ν, we have the invariance

$$\mathbb{P}\left[X_{[d]}\right] = \langle \mathbb{P}, \nu\rangle\left[X_{[d]}\right]$$

and can therefore safely ignore the base measures.

Starting with Chapter 3 we will further investigate boolean tensors and relate them with propositional formulas. In Chapter 4 we will connect the representation and positivity with respect to boolean base measures with the formalism of entailment.

We now investigate, which base measures ν can be chosen for a probability distribution $\mathbb{P}$, such that $\mathbb{P}$ is representable by ν. Here we want to finde a ν, which is in a sense to be defined minimal amount the base measures, such that $\mathbb{P}$ is representable with respect to them. For this minimality criterion we will develop in Chapter 4 orders based on entailment and show the minimality in The. 4.12. Here, we just introduce the minimality criterion as positivity of a distribution with respect to a base measure.

Definition 1.3 We say that a probability distribution $\mathbb{P}$ is positive with respect to a boolean base measure ν, if the distribution is representable by ν (i.e. $\langle \mathbb{P}, \nu\rangle\left[\varnothing\right] = 1$) and for all $x_{[d]}$

with $\nu\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = 1$ we have $\mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] > 0$.

1.2 Marginal Distribution

Contractions of probability distributions are related to marginalizations as we introduce next.

Definition 1.4 (Marginal Probability) Given a distribution $\mathbb{P}\left[X_0, X_1\right]$ of the categorical variables X_0 and X_1 the marginal distribution of the categorical variable X_0 is defined for each x_0 as the tensor

$$\mathbb{P}\left[X_0\right] : [m_0] \to \mathbb{R}$$

defined for $x_0 \in [m_0]$ by

$$\mathbb{P}\left[X_0 = x_0\right] = \sum_{x_1 \in [m_1]} \mathbb{P}\left[X_0 = x_0, X_1 = x_1\right] .$$

Def. 1.4 generalizes to marginalizations of sets of variables, since we can always group a set of categorical variables and understand them as a single one.

Theorem 1.5 *For any distribution $\mathbb{P}\left[X_0, X_1\right]$ the marginal distribution of the variable X is the contraction*

$$\mathbb{P}\left[X_0\right] = \langle \mathbb{P} \rangle \left[X_0\right] .$$

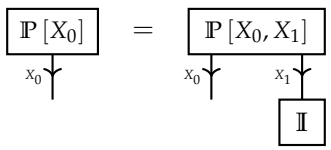
Further, any marginal distribution is a probability distribution.

Proof. We have $\mathbb{P}\left[X_0\right] = \langle \{\mathbb{P}\} \rangle \left[X_0\right]$ by definition. To show that $\mathbb{P}\left[X_0\right]$ is a probability distribution, we need to show that $\langle \mathbb{P}\left[X_0\right] \rangle \left[\varnothing\right] = 1$. But this follows from the normation of $\mathbb{P}$ and the commutativity of contractions (see The. 13.1 in Chapter 13) as

$$\langle \mathbb{P}\left[X_0\right] \rangle \left[\varnothing\right] = \langle \langle \mathbb{P} \rangle \left[X_0\right] \rangle \left[\varnothing\right] = \langle \mathbb{P} \rangle \left[\varnothing\right] = 1 .$$

■

We depict the sum over the possible values of X_1 by contraction of the probability tensor with the trivial tensors $\mathbb{I}$ as



Let us notice, that marginal distributions are probability tensors for themself, which we again denote by a directed leg.

1.3 Conditional Probabilities

Normations of probability distributions result in conditional distributions as we define next.

Definition 1.6 (Conditional Probability) Let $\mathbb{P}\left[X_0, X_1\right]$ be a distribution of the categorical variables X_0 and X_1, such that $\mathbb{P}$ is normable on $\{X_1\}$. Then the distribution of X_0 condi-

tioned on X_1 is defined by

$$\mathbb{P}\left[X_0 = x_0 | X_1 = x_1\right] = \frac{\mathbb{P}\left[X_0 = x_0, X_1 = x_1\right]}{\mathbb{P}\left[X_1 = x_1\right]}\,.$$

We show in the next theorem, that conditional distributions are calculated by normations.

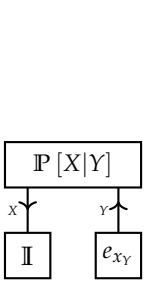
Theorem 1.7 *The tensor $\mathbb{P}\left[X_0|X_1\right]$ is the normation of $\mathbb{P}\left[X_0, X_1\right]$ on X_1 (see Def. 0.18), that is*

$$\mathbb{P}\left[X_0|X_1\right] = \langle \mathbb{P} \rangle \left[X_0|X_1\right]\,.$$

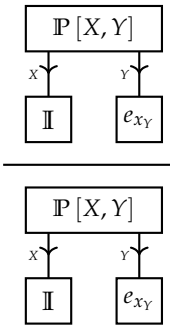
Further, for any $x_1 \in [m_1]$ the tensor $\mathbb{P}\left[X_0|X_1 = x_1\right]$ is a probability tensor.

Proof. The first claim follows from a comparison of Def. 1.6 and 0.18. The second claim follows from the first and The. 10.11. Alternatively, the second claim can be showed using the diagrammatic notation as

$\sum_{x_{X_0}} \mathbb{P}\left[X = x_{X_0} | Y = x_{X_1}\right]$ =



=



$= 1\,.$

■

The. 1.5 and 1.7 show that the formalism of contractions and normations is applied in basic operations of probabilistic reasoning.

We can further show, that exactly the directed tensors with non-negative coordinates are conditional probability tensors.

Theorem 1.8 *Any tensor with non-negative coordinates is a conditional distribution tensor, if and only if it is directed with the condition variables ingoing and the other outgoing.*

Proof. "⇒": By The. 1.7 a conditional probability tensor $\mathbb{P}\left[X_0|X_1\right]$ is the normation of a tensor and by The. 10.11 a directed tensor. Since probability tensors have only non-negative coordinates, their contractions with one-hot encodings also have only non-negative coordinates and also their normations.

"⇐": Conversely, let $T\left[X_{\mathcal{V}}\right]$ be a directed tensor with $\mathcal{V}^{\text{in}}$ incoming and $\mathcal{V}^{\text{out}}$ outgoing and non-negative coordinates. Then

$$\mathbb{P}\left[X_{\mathcal{V}}\right] = \frac{1}{\prod_{v \in \mathcal{V}^{\text{in}}} m_v} \cdot T\left[X_{\mathcal{V}}\right]$$

is a probability tensor, since

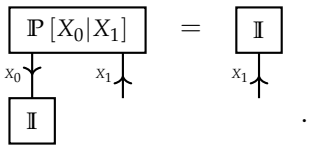
$$\sum_{x_{\mathcal{V}^{\text{in}}}} \sum_{x_{\mathcal{V}^{\text{out}}}} \mathbb{P}\left[X_{\mathcal{V}} = x_{\mathcal{V}}\right] = \sum_{x_{\mathcal{V}^{\text{in}}}} \sum_{x_{\mathcal{V}^{\text{out}}}} \frac{1}{\prod_{v \in \mathcal{V}^{\text{in}}} m_v} \cdot T\left[X_{\mathcal{V}} = x_{\mathcal{V}}\right] = \sum_{x_{\mathcal{V}^{\text{in}}}} \frac{1}{\prod_{v \in \mathcal{V}^{\text{in}}} m_v} = 1\,.$$

The conditional probability $\mathbb{P}\left[X_{\mathcal{Y}^{\text{out}}} | X_{\mathcal{Y}^{\text{in}}}\right]$ coincides with T, since

$$\begin{aligned}\mathbb{P}\left[X_{\mathcal{Y}^{\text{out}}} | X_{\mathcal{Y}^{\text{in}}} = x_{\mathcal{Y}^{\text{in}}}\right] =& \frac{\mathbb{P}\left[X_{\mathcal{Y}^{\text{out}}}, X_{\mathcal{Y}^{\text{in}}} = x_{\mathcal{Y}^{\text{in}}}\right]}{\sum_{x_{\mathcal{Y}^{\text{out}}}} \mathbb{P}\left[X_{\mathcal{Y}^{\text{out}}} = x_{\mathcal{Y}^{\text{out}}}, X_{\mathcal{Y}^{\text{in}}} = x_{\mathcal{Y}^{\text{in}}}\right]} \\ =& \frac{T\left[X_{\mathcal{Y}^{\text{out}}}, X_{\mathcal{Y}^{\text{in}}} = x_{\mathcal{Y}^{\text{in}}}\right]}{\sum_{x_{\mathcal{Y}^{\text{out}}}} T\left[X_{\mathcal{Y}^{\text{out}}} = x_{\mathcal{Y}^{\text{out}}}, X_{\mathcal{Y}^{\text{in}}} = x_{\mathcal{Y}^{\text{in}}}\right]} = T\left[X_{\mathcal{Y}^{\text{out}}}, X_{\mathcal{Y}^{\text{in}}} = x_{\mathcal{Y}^{\text{in}}}\right] ,\end{aligned}$$

where in the last equation we used that the denominator is by definition trivial since T is normed. ∎

Since conditional probabilities are directed tensors we therefore depict them by



.

The. 1.8 specifies a broad class of tensors to represent conditional probabilities. In combination with The. 11.8, which states that relational encodings are directed, we get that any relational encoding of a function is a conditional probability tensor.

1.4 Bayes Theorem and the Chain Rule

So far, we have connected concepts of probability theory such as marginal and conditional probabilities with contractions and normations of tensors. We will now proceed to show that basic theorems of probability theory translate into more general contraction equations.

Theorem 1.9 (Bayes Theorem) *For any probability distribution $\mathbb{P}\left[X_0, X_1\right]$ with positive $\mathbb{P}\left[X_1\right]$ we have*

$$\mathbb{P}\left[X_0, X_1\right] = \left\langle \mathbb{P}\left[X_0 | X_1\right], \mathbb{P}\left[X_1\right] \right\rangle \left[X_0, X_1\right] .$$

Proof. Directly from the more generic contraction equation The. 10.12, since by assumption of positivity of $\mathbb{P}\left[X_1\right]$, the tensor network $\mathbb{P}$ is normable with respect to X_1. ∎

Probability distributions can be decomposed into conditional probabilities, as we demonstrate in the next theorem.

Theorem 1.10 (Chain Rule) *For any joint probability distribution $\mathbb{P}$ of the variables $\mathbb{P}\left[X_0, \ldots, X_{d-1}\right]$ we have*

$$\mathbb{P} = \left\langle \mathbb{P}\left[X_k, \ldots, X_{d-1} | X_0, \ldots, X_{k-1}\right] \,:\, k \in [d] \right\rangle \left[X_0, \ldots, X_{d-1}\right]$$

where for $k = 0$ we denote by $\mathbb{P}\left[X_0 | X_0, \ldots, X_{-1}\right]$ the marginal distribution $\mathbb{P}\left[X_0\right]$.

Proof. This follows from The. 10.13. ∎

1.5 Independent Variables

Independence leads to severe simplifications of conditional probabilities and is thus the key assumption to gain sparse decompositions. We will demonstrate this here applying the chain rule.

Definition 1.11 (Independence) Given a joint distribution of variables X_0 and X_1, we say that X_0 is independent from X_1 if for any values x_0, x_1 we have

$$\mathbb{P}\left[X_0 = x_0, X_1 = x_1\right] = \mathbb{P}\left[X_0 = x_0\right] \cdot \mathbb{P}\left[X_0 = x_0\right] .$$

We give a criterion on independence based on a contraction equation of the probability distribution in the next theorem.

Theorem 1.12 *Given a probability distribution $\mathbb{P}$, X_0 is independent from X_1, if and only if*

$$\mathbb{P}\left[X_0, X_1\right] = \langle \langle \mathbb{P} \rangle \left[X_0\right], \langle \mathbb{P} \rangle \left[X_1\right] \rangle \left[X_0, X_1\right] .$$

Proof. By The. 1.5 we know that marginal probabilities are equivalent to contracted probability distributions, i.e. $\mathbb{P}\left[X_0\right] = \langle \{\mathbb{P}\} \rangle \left[X_0\right]$. By orthogonality of one-hot encodings we have that

$$\forall x_0, x_1 : \quad \mathbb{P}\left[X_0 = x_0, X_1 = x_1\right] = \mathbb{P}\left[X_0 = x_0\right] \cdot \mathbb{P}\left[X_0 = x_0\right]$$

is equivalent to

$$\sum_{x_0} \sum_{x_1} \mathbb{P}\left[X_0 = x_0, X_1 = x_1\right] \cdot e_{x_0}\left[X_0\right] e_{x_1}\left[X_1\right] = \sum_{x_0} \sum_{x_1} \mathbb{P}\left[X_0 = x_0\right] \cdot \mathbb{P}\left[X_0 = x_0\right] \cdot e_{x_0}\left[X_0\right] e_{x_1}\left[X_1\right] .$$

We reorder the summations and arrive at

$$\sum_{x_0, x_1} \mathbb{P}\left[X_0 = x_0, X_1 = x_1\right] \cdot e_{x_0, x_1}\left[X_0, X_1\right] = \left(\sum_{x_0} \mathbb{P}\left[X_0 = x_0\right] e_{x_0}\left[X_0\right] \right) \cdot \left(\sum_{x_1} \mathbb{P}\left[X_0 = x_0\right] \cdot e_{x_1}\left[X_1\right] \right)$$

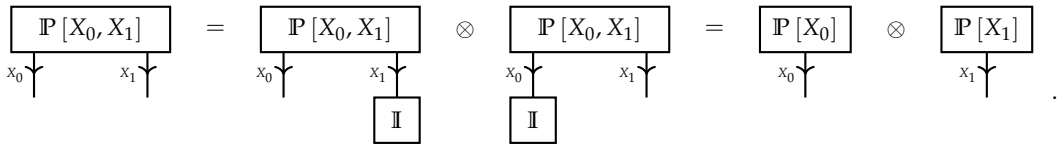
which is by Lem. 10.1 equal to the claim

$$\mathbb{P}\left[X_0, X_1\right] = \langle \langle \mathbb{P} \rangle \left[X_0\right], \langle \mathbb{P} \rangle \left[X_1\right] \rangle \left[X_0, X_1\right] .$$

■

Independent variables result in decompositions of $\mathbb{P}$ in a tensor product of marginal probability tensors. Having pairwise independent variables reduces the degrees of freedom from exponentially many in the number of atoms to linear.

In the tensor network decomposition we depict this by



Independence is a very strong assumption, which is often too restrictive. Conditional independence instead is a less demanding assumption, when certain conditional distribution variables are independent. This leads to tensor network decompositions with a more realistic assumption.

Definition 1.13 (Conditional Independence) Given a joint distribution of variables X_0, X_1 and X_2, we say X_0 is independent from X_1 conditioned on X_2 if for any incides x_0, x_1 and x_2

$$\mathbb{P}\left[X_0 = x_0, X_1 = x_1 | X_2 = x_2\right] = \mathbb{P}\left[X_0 = x_0 | X_2 = x_2\right] \cdot \mathbb{P}\left[X_1 = x_1 | X_2 = x_2\right] .$$

Conditional independence is a relation between conditional probabilities and is therefore equivalent to a normation equation stated next.

Theorem 1.14 (Conditional Independence as a Contraction Equation) *Given a distribution $\mathbb{P}$ of variables X_0, X_1 and X_2, the variable X_0 is independent from X_1 if and only if the contraction equation*

$$\mathbb{P}\left[X_0, X_1 | X_2\right] = \langle \mathbb{P}\left[X_0|X_2\right], \mathbb{P}\left[X_1|X_2\right] \rangle \left[X_0, X_1, X_2\right]$$

holds.

Proof. Directly by The. 1.7 used on the conditional probabilities in Def. 1.13. ∎

We can exploit conditional independence to find tensor network decompositions of probability tensors, as we show in the next theorem.

Corollary 1.15 *If and only if X_0 is independent from X_1 conditioned on X_2 the probability distribution $\mathbb{P}$ satisfies*

$$\mathbb{P}\left[X_0, X_1, X_2\right] = \langle \{\mathbb{P}\left[X_0|X_2\right], \mathbb{P}\left[X_1|X_2\right], \mathbb{P}\left[X_2\right]\} \rangle \left[X_0, X_1, X_2\right] .$$

Proof. Follows from The. 1.14 and The. 1.9. ∎

Corollary 1.16 *Whenever X_0 is independent of X_1 given X_2, we have*

$$\mathbb{P}\left[X_0|X_1, X_2\right] = \mathbb{P}\left[X_0|X_2\right] .$$

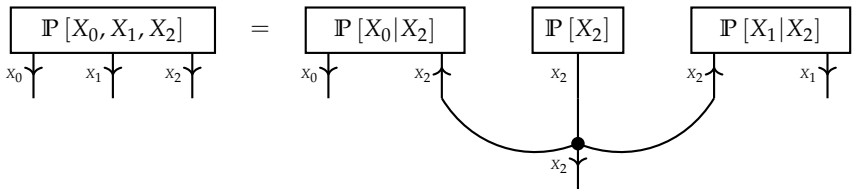


Figure 1.1: Diagrammatic visualization of the contraction equation in Corollary 1.15. Conditional independence of X_0 and X_1 given X_2 holds if the contraction on the right ride is equal to the probability tensor on the left side.

Theorem 1.17 (Markov Chain) *Let there be a set of variables X_t where $t \in [T]$. When X_t is independent of $X_{0:t-2}$ conditioned on X_{t-1} (the Markov Property), then*

$$\mathbb{P} = \langle \{\mathbb{P}\left[X_t|X_{t-1}\right] \,:\, t \in [T]\} \rangle \left[X_0, \ldots, X_{T-1}\right]$$

We depict this decomposition in Figure 1.2.

Proof. By the chain rule (The. 1.10) we have

$$\mathbb{P}\left[X_0, \ldots, X_{T-1}\right] = \langle \{\mathbb{P}\left[X_t|X_{0:t}\right] : t \in [T]\} \rangle \left[X_{[T]}\right]$$

Using the conditional independence of X_t and $X_{0:t-2}$ conditioned on X_{t-1} we further have by Corollary 1.16

$$\mathbb{P}\left[X_t|X_{0:t} = x_{0:t}\right] = \mathbb{P}\left[X_t|X_{t-1} = x_{t-1}\right] .$$

Composing both equalities shows the claim. ∎

Here we denoted by $X_{0:t}$ the tuple $X_0, ..., X_t$.

Remark 1.18 Let us notice that the dimensionality dropped drastically through applying the independence assumption. The tensor space in the naive representation of any probability distribution has

$$\prod_{t\in[T]} m_t$$

coordinates, while the Markov Chain is represented by

$$\sum_{t\in[T]} m_t \cdot m_{t-1} \,.$$

Replacing exponential scaling with the number of variables to linear scaling is the advantage of tensor network decompositions. ◊

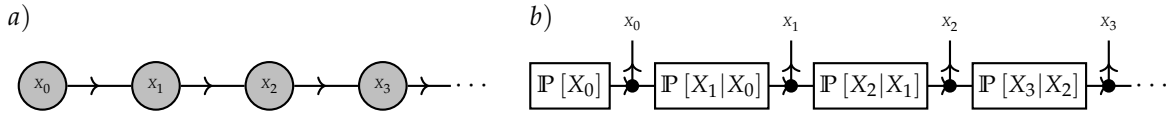


Figure 1.2: Depiction of a Markov Chain. a) Dependency Graph (of the corresponding chain Graphical Model). b) Dual Tensor Network representing the conditional probability factors.

1.6 Graphical Models

We have already depicted conditional dependency assumptions made for Markov Chains in Figure 1.2 and discussed the implied decomposition of the dual tensor networks. Graphical models provide a more general framework for conditional dependency assumptions and provide a generic approach to exploit independences in finding tensor network decompositions of $\mathbb{P}$.

Following the tensor network formalism we in this section introduce graphical models based on hypergraphs. Whether the hypergraph is directed or not distinguished between Bayesian Networks and Markov Networks.

1.6.1 Markov Networks

While typically Markov Networks are defined on graphs, we define them here on hypergraphs to establish a direct connection to tensor networks defined on the same hypergraph. Along that line, Markov Networks are tensor networks with non-negative tensors (see Def. 0.9), which are interpreted as probability distributions after normation.

Definition 1.19 (Markov Network) Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network of non-negative tensors on a hypergraph $\mathcal{G}$. Then the Markov Network to $\mathcal{T}^{\mathcal{G}}$ is the probability distribution of X_v defined by the tensor

$$\mathbb{P}^{\mathcal{G}}[X_{\mathcal{V}}] = \frac{\langle\{T^e : e \in \mathcal{E}\}\rangle[X_{\mathcal{V}}]}{\langle\{T^e : e \in \mathcal{E}\}\rangle[\varnothing]} = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}|\varnothing] \,.$$

We call the denominator

$$\mathcal{Z}\left(\mathcal{T}^{\mathcal{G}}\right) = \langle\{T^e : e \in \mathcal{E}\}\rangle[\varnothing]$$

the partition function of the Markov Network.

Often, we are only interested in the distribution of a subset of variables, which are called the observable variables, and call the other variables hidden variables. The marginalization of a Markov Network to $\mathcal{T}^{\mathcal{G}}$ on the variables $X_{\tilde{\mathcal{V}}}$ is

$$\mathbb{P}^{\mathcal{G}}\left[X_{\tilde{\mathcal{V}}}\right] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\tilde{\mathcal{V}}} | \varnothing\right] .$$

This can be derived from The. 13.1, which established an equivalence of contractions with sequences of consecutive contractions.

Further, the distribution of $X_{\tilde{\mathcal{V}}}$ conditioned on $X_{\bar{\mathcal{V}}}$, where $\tilde{\mathcal{V}}, \bar{\mathcal{V}}$ are disjoint subsets of $\mathcal{V}$, is

$$\mathbb{P}^{\mathcal{G}}\left[X_{\tilde{\mathcal{V}}} | X_{\bar{\mathcal{V}}}\right] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\tilde{\mathcal{V}}} | X_{\bar{\mathcal{V}}}\right] .$$

Definition 1.20 (Separation of Hypergraph) A path in a hypergraph is a sequence of nodes v_k for $k \in [d]$, such that for any $k \in [d-1]$ we find a hyperedge $e \in \mathcal{E}$ such that $(v_k, v_{k+1}) \subset e$. Given disjoint subsets A, B, C of nodes in a hypergraph $\mathcal{G}$ we say that C separates A and B with respect to $\mathcal{G}$, when any path starting at a node in A and ending in a node in B contains a node in C.

To characterize Markov Networks in terms of conditional independencies we need to further define the property of clique-capturing. This property of clique-capturing established a correspondence of hyperedges with maximal cliques in an alternative graph-based definition of Markov Networks [KF09].

Definition 1.21 (Clique-Capturing Hypergraph) We call a hypergraph $\mathcal{G}$ clique-capturing, when each subset $\tilde{\mathcal{V}} \subset \mathcal{V}$ is contained in a hyperedge, if for any $a, b \in \tilde{\mathcal{V}}$ there is a hyperedge $e \in \mathcal{E}$ with $a, b \in \tilde{\mathcal{V}}$.

Let us now show a characterization of Markov Networks in terms of conditional independencies, which is analogous to The. 1.25.

Theorem 1.22 (Hammersley-Clifford) *Given a clique-capturing hypergraph $\mathcal{G}$, the set of positive Markov Networks on the hypergraph coincides with the set of positive probability distributions, such that each for each disjoint subsets of variables A, B, C we have X_A is independent of X_B conditioned on X_C, when C separates A and B in the hypergraph.*

Proof. Let there be a hypergraph $\mathcal{G}$, a Markov Network $\mathcal{T}^{\mathcal{G}}$ on $\mathcal{G}$ and nodes $A, B, C \subset \mathcal{V}$, such that C separates A from B. Let us denote by $\mathcal{V}_0$ the nodes with paths to A, which do not contain a node in C, and by $\mathcal{V}_1$ the nodes with paths to B, which do not contain a node in C. Further, we denote by $\mathcal{E}_0$ the hyperedges which contain a node in $\mathcal{V}_0$ and by $\mathcal{E}_1$ the hyperedges which contain a node in $\mathcal{V}_1$. By assumption of separability, both sets $\mathcal{E}_0$ and $\mathcal{E}_1$ are disjoint and no node in A is in a hyperedge in $\mathcal{E}_1$, respectively no node in B is in a hyperedge in $\mathcal{E}_0$, . We then have

$$\begin{aligned}\left\langle \{T^e : e \in \mathcal{E}\} \right\rangle \left[X_A, X_B | X_C = x_C\right] &= \left\langle \{T^e : e \in \mathcal{E}\} \cup \{e_{x_C}\} \right\rangle \left[X_A, X_B | \varnothing\right] \\ &= \left\langle \{T^e : e \in \mathcal{E}_0\} \cup \{e_{x_C}\} \right\rangle \left[X_A | \varnothing\right] \otimes \left\langle \{T^e : e \in \mathcal{E}_1\} \cup \{e_{x_C}\} \right\rangle \left[X_B | \varnothing\right] .\end{aligned}$$

By The. 1.14, it now follows that X_A is independent of X_B conditioned on X_C. The converse direction, i.e. that positive distributions respecting the conditional indpendence assumptions are representable as Markov Networks, is known as the Hammersley Clifford Theorem, which we will proof later in Sect. 10.4. ∎

From the proof of The. 1.22 Markov Networks with zero coordinates still satisfy the conditional independence assumption. However, the reverse is not true, that is there are distributions with

vanishing coordinates, which satisfy the conditional independence assumptions, but cannot be represented as a Markov Network (see Example 4.4 in [KF09]).

1.6.2 Bayesian Networks

Compared to Markov Networks, Bayesian Networks impose further conditions on tensor networks representing a distribution. They assume a directed hypergraph and each tensor decorating the edges to be normed according to the direction. We will observe, that if the hypergraph is in addition acyclic, then each tensor core coincides with the conditional distribution of the underlying Markov Network. To introduce Bayesian Networks, we extend Def. 0.8 by introducing the property of acyclicity for hypergraphs.

Definition 1.23 A directed path is a sequence $v_0, \dots v_r$ such that for any $l \in [r]$ there is an hyperedge $e = (e^{\text{in}}, e^{\text{out}}) \in \mathcal{E}$ such that $v_l \in e^{\text{in}}$ and $v_{l+1} \in e^{\text{out}}$. We call the hypergraph $\mathcal{G}$ acyclic, if there is no path with $r > 0$ such that $v_0 = v_r$. Given a directed hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we define for any node $v \in \mathcal{V}$ its parents by

$$\text{Pa}(v) = \{\tilde{v} : \exists e = (e^{\text{in}}, e^{\text{out}}) \in \mathcal{E} : \tilde{v} \in e^{\text{in}}, v \in e^{\text{out}}\}$$

and its non-descendants $\text{NonDes}(v)$ as the set of nodes $\tilde{v}$, such that there is no directed path from v to $\tilde{v}$.

Based on these additional graphical properties, we now define Bayesian Networks.

Definition 1.24 (Bayesian Network) Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic hypergraph with edges of the form

$$\mathcal{E} = \{(\text{Pa}(v), \{v\}) : v \in \mathcal{V}\}.$$

A *Bayesian Network* is a decoration of each edge $(\text{Pa}(v), \{v\})$ by a conditional probability distribution

$$\mathbb{P}\left[X_v | X_{\text{Pa}(v)}\right]$$

which represents the probability distribution

$$\mathbb{P}[X_{\mathcal{V}}] = \left\langle \{\mathbb{P}\left[X_v | X_{\text{Pa}(v)}\right] : v \in \mathcal{V}\} \right\rangle [X_{\mathcal{V}}] .$$

By definition each tensor decorating a hyperedge is directed with $X_{\text{Pa}(v)}$ incoming and X_v outgoing. Thus, the directionality of the hypergraph is reflected in each tensor decorating a directed hyperedge. This allows us to verify with The. 10.14 that their contraction defines a probability distribution.

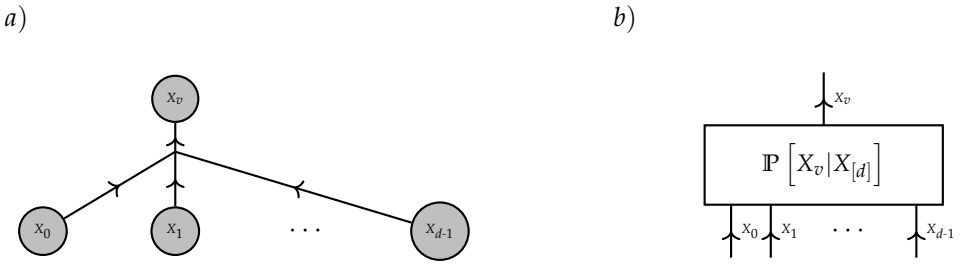


Figure 1.3: Example of a Factor of a Bayesian Network to the node X_v with parents $X_0, \dots, X_{d-1}$, as subgraph a) and dual tensor core b).

Marginalization of a Bayesian Network are still Bayesian Networks on a graph where the edges directing to variables, which are not marginalized over, are replaced by directed edges to the children. Conditioned Bayesian Network do not have a simple Bayesian Network representation, which is why we will treat them as Markov Networks to be introduced next.

Theorem 1.25 (Independence Characterization of Bayesian Networks) *A probability distribution $\mathbb{P}[X_{\mathcal{V}}]$ has a representation by a Bayesian Network on a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if and only if for any $v \in \mathcal{V}$ the variables X_v are independent on* $\mathrm{NonDes}(v)$ *conditioned on* $\mathrm{Pa}(v)$.

Proof. We choose a topological order $\prec$ on the nodes of $\mathcal{G}$, which exists since $\mathcal{G}$ is acyclic.

"$\Rightarrow$": Let us assume, that the conditional independencies are satisfied and apply the chain rule with respect to that ordering to get

$$\mathbb{P}[X_{\mathcal{V}}] = \langle \mathbb{P}[X_v | X_{\tilde{v}} : \tilde{v} \prec v] \rangle [X_{\mathcal{V}}] \ .$$

Since $\prec$ is a topological ordering we have

$$\mathrm{Pa}(v) \subset \{\tilde{v} : \tilde{v} \prec v\}$$

We apply the assumed conditional independence with Corollary 1.16 and get

$$\mathbb{P}[X_{\mathcal{V}}] = \left\langle \mathbb{P}\left[X_v | X_{\mathrm{Pa}(v)}\right] \right\rangle [X_{\mathcal{V}}] \ .$$

"$\Leftarrow$": To show the converse direction, let there be a Bayesian Network $\mathbb{P}[X_{\mathcal{V}}]$ on $\mathcal{G}$. To show for any node v, that X_v is independent of $\mathrm{NonDes}(v)$ conditioned on $\mathrm{Pa}(v)$, we reorder the tensors in the contraction

$$\begin{aligned}
&\mathbb{P}\left[X_v, X_{\mathrm{NonDes}(v)} | X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)}\right] \\
&\quad = \left\langle \{\mathbb{P}\left[X_{\tilde{v}} | X_{\mathrm{Pa}(\tilde{v})}\right] \ : \ \tilde{v} \in \mathcal{V}\} \right\rangle \left[X_v, X_{\mathrm{NonDes}(v)} | X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)}\right] \\
&\quad = \left\langle \{\mathbb{P}\left[X_{\tilde{v}} | X_{\mathrm{Pa}(\tilde{v})}\right] \ : \ \tilde{v} \in \mathcal{V}\} \cup \{e_{x_{\mathrm{Pa}(v)}}\} \right\rangle \left[X_v, X_{\mathrm{NonDes}(v)} | \varnothing\right] \\
&\quad = \left\langle \{\mathbb{P}\left[X_{\tilde{v}} | X_{\mathrm{Pa}(\tilde{v})}\right] \ : \ \tilde{v} \in \mathrm{NonDes}(v)\} \cup \{e_{x_{\mathrm{Pa}(v)}}, \mathbb{P}\left[X_v | X_{\mathrm{Pa}(v)}\right]\} \right\rangle \left[X_v, X_{\mathrm{NonDes}(v)} | \varnothing\right] \\
&\quad = \left\langle \{\mathbb{P}\left[X_{\tilde{v}} | X_{\mathrm{Pa}(\tilde{v})}\right] \ : \ \tilde{v} \in \mathrm{NonDes}(v)\} \cup \{e_{x_{\mathrm{Pa}(v)}}\} \right\rangle \left[X_{\mathrm{NonDes}(v)} | \varnothing\right] \\
&\qquad \cdot \left\langle \{\mathbb{P}\left[X_v | X_{\mathrm{Pa}(v)}\right], e_{x_{\mathrm{Pa}(v)}}\} \right\rangle [X_v | \varnothing] \\
&\quad = \left\langle \{\mathbb{P}\left[X_{\mathrm{NonDes}(v)} | X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)}\right], \mathbb{P}\left[X_v | X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)}\right]\} \right\rangle \left[X_v, X_{\mathrm{NonDes}(v)}\right]
\end{aligned}$$

Here we have dropped in the third equation all tensors to the descendants, since their marginalization is trivial (which can be shown by a leaf-stripping argument). In the fourth equation we made use of the fact, that any directed path between the non-descendants and the node is through the parents of the node. By The. 1.14, it now follows that X_v is independent of $\mathrm{NonDes}(v)$ conditioned on $\mathrm{Pa}(v)$. ∎

1.6.3 Bayesian Networks as Markov Networks

Markov Networks are more flexible compared with Bayesian Networks, since any Bayesian Network is a Markov Network by ignoring the directionality of the hypergraph and understanding the conditional distributions as generic tensor cores. In the next theorem we provide the conditions for the interpretation of a Markov Network as a Bayesian Network.

Theorem 1.26 *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network on a directed acyclic hypergraph, such that the edges are of the structure*

$$\mathcal{E} = \{(\mathrm{Pa}(v), \{v\}) \,:\, v \in \mathcal{V}\}$$

and each tensor T^e respects the directionality of the graph, that is each $T^{(\mathrm{Pa}(v),\{v\})}$ is directed with the variables to $\mathrm{Pa}(v)$ incoming and v outgoing. Then $\mathcal{Z}\left(\mathcal{T}^{\mathcal{G}}\right) = 1$ and for each $v \in \mathcal{V}$ we have

$$T^{(\mathrm{Pa}(v),\{v\})} = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_v | X_{\mathrm{Pa}(v)} \right] .$$

In particular, $\mathcal{T}^{\mathcal{G}}$ is a Bayesian Network.

Proof. We show the claim by induction over the cardinality of $\mathcal{V}$.

$|\mathcal{V}| = 1$: In this case we find a unique node $v \in \mathcal{V}$ and have $\mathcal{E} = \{(\varnothing, \{v\})\}$. The tensor $T^{(\varnothing,\{v\})}$ is then normed with no incoming variables and we thus have

$$\mathcal{Z}\left(\mathcal{T}^{\mathcal{G}}\right) = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing] = \left\langle T^{(\varnothing,\{v\})} \right\rangle [\varnothing] = 1$$

and

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_v | \varnothing] = T^{(\varnothing,\{v\})} .$$

$|\mathcal{V}| - 1 \to |\mathcal{V}|$: Let there now be a directed hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let us now assume, that the theorem holds for any tensor networks with node cardinality $|\mathcal{V}| - 1$. Since the hypergraph is acyclic, we find a root $v \in \mathcal{V}$ such that $v \notin \mathrm{Pa}(\tilde{v})$ for $\tilde{v} \in \mathcal{V}$. We denote $\mathcal{T}^{\tilde{\mathcal{G}}}$ the tensor network on the hypergraph $\tilde{\mathcal{G}} = \{\mathcal{V}/\{v\}, \mathcal{E}/\{(\mathrm{Pa}(v), \{v\})\}\}$ with decorations inherited from $\mathcal{T}^{\mathcal{G}}$. With Theorem 13.1, the directionality of $T^{(\mathrm{Pa}(v),\{v\})}$ and the induction assumption on $\mathcal{T}^{\tilde{\mathcal{G}}}$ we have

$$\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \cup \left\{ T^{(\mathrm{Pa}(v),\{v\})} \right\} \right\rangle [\varnothing] = \left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \cup \left\{ \left\langle T^{(\mathrm{Pa}(v),\{v\})} \right\rangle \left[X_{\mathrm{Pa}(v)} \right] \right\} \right\rangle [\varnothing] = \left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \cup \left\{ \mathbb{I} \left[X_{\mathrm{Pa}(v)} \right] \right\} \right\rangle [\varnothing] = 1$$

and thus a trivial partition function. Since v does not appear in $\tilde{\mathcal{G}}$, we have for any index $x_{\mathrm{Pa}(\mathcal{V})}$

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_v, X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)} \right] = \left\langle T^{(\mathrm{Pa}(v),\{v\})} \right\rangle \left[X_v, X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)} \right] \cdot \left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle \left[X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)} \right]$$

and thus, since $T^{(\mathrm{Pa}(v),\{v\})}$ is directed, that

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_v | X_{\mathrm{Pa}(v)} \right] = T^{(\mathrm{Pa}(v),\{v\})} .$$

■

Theorem 1.26 states that Bayesian Networks are a subset of Markov Networks. While Markov Network allow generic tensor cores, Bayesian Networks impose a local directionality condition on each tensor core by demanding it to be a conditional probability tensor. In our diagrammatic notation, the local normation of Bayesian Networks is highlighted by the directionality of the hypergraph. Generic Markov Networks are on undirected hypergraphs, where in general no local directionality condition is assumed. As a consequence, tasks such as the determination of the partition functions or calculation of conditional distributions involve global contractions.

1.6.4 Example: Hidden Markov Models

We here extend the example of Markov Chains from Theorem 1.17 to a limited observation of the variables by observations. Let there be the variables X_t (states) and E_t (observations) with a discrete and finite time $t \in [T]$.

The conditional assumptions are

- X_{t+1} is independent of $X_{0:t-1}$ and $E_{0:t}$ conditioned on X_t
- E_t is independent of all other variables conditioned on X_t

Then the probability tensor has the decomposition

$$\begin{aligned}\mathbb{P}\left[X_{0:T}, E_{0:T}\right] &= \prod_{t\in[T]} \left(\mathbb{P}\left[X_t|X_{0:t-1}, E_{0:t-1}\right] \cdot \mathbb{P}\left[E_t|X_{0:t}, E_{0:t-1}\right]\right) \\ &= \mathbb{P}\left[X_0\right] \cdot \mathbb{P}\left[E_0|X_0\right] \cdot \prod_{t\in[T],t>0} \left(\mathbb{P}\left[X_t|X_{t-1}\right] \cdot \mathbb{P}\left[E_t|X_t\right]\right)\end{aligned}$$

Here we used the Chain Rule decomposition of Theorem 1.10 in the first equation and the conditional independence assumptions in the second.

We notice, that this is a Bayesian Netowork on a directed acyclic hypergraph $\mathcal{G}$ consistent in nodes to each state and each observation and directed hyperedges

- $(\{X_t\}, \{X_{t+1}\})$ for $t \in [T-1]$
- $(\{X_t\}, \{E_t\})$ for $t \in [T]$

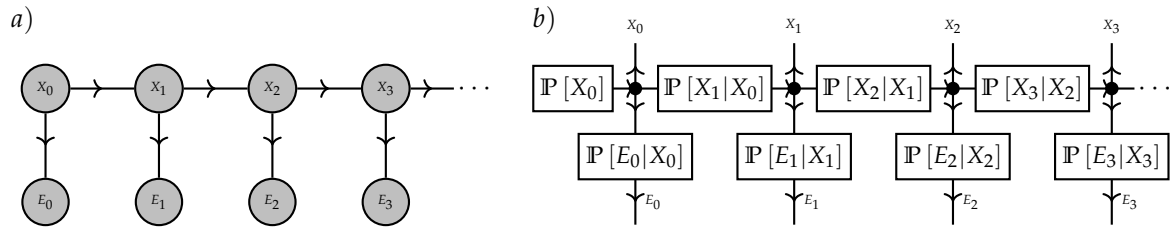


Figure 1.4: Depiction of a Hidden Markov Model. a) Dependency Graph (of the corresponding chain Graphical Model). b) Dual Tensor Network representing the conditional probability factors.

1.7 Exponential Families

Exponential families are collections of probability distributions, where each coordinate is determined by a base measure and a set ϕ of features as

$$\mathbb{P}\left[X_{[d]} = x_{[d]}\right] \propto \nu(x) \cdot \exp\left[\sum_{l\in[p]} \phi_l\left[X_{[d]} = x_{[d]}\right] \cdot \theta\left[L = l\right]\right].$$

We use the selection encoding to represent the weighted summation over the statistics, that is the tensor

$$\gamma^{\phi}\left[X_{[d]}, L\right] : \bigtimes_{k\in[d]} [m_k] \times [p] \to \mathbb{R}$$

with

$$\gamma^{\phi}\left[X_{[d]} = x_{[d]}, L = l\right] = \phi_l\left[X_{[d]} = x_{[d]}\right].$$

We then understand θ as a vector to the categorical variable L and use Theorem 11.20 to get

$$\sum_{l\in[p]} \theta\left[L = l\right] \cdot \phi_l\left[X_{[d]}\right] = \left\langle \gamma^{\phi}\left[X_{[d]}, L\right], \theta\left[L\right]\right\rangle \left[X_{[d]}\right].$$

Definition 1.27 Given a sufficient statistics

$$\phi : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R}^p$$

and a boolean base measure

$$\nu : \bigtimes_{k\in[d]} [m_k] \to [2]$$

with $\langle\nu\rangle[\varnothing] \neq 0$, the set $\Gamma^{\phi,\nu} = \{\mathbb{P}^{(\phi,\theta,\nu)} : \theta[L] \in \mathbb{R}^p\}$ of probability distributions

$$\mathbb{P}^{(\phi,\theta,\nu)}\left[X_{[d]}\right] = \left\langle \exp\left[\left\langle \gamma^\phi, \theta\right\rangle \left[X_{[d]}\right], \nu\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$$

is called the exponential family to ϕ. We further define for each member with parameters θ the associated energy tensor

$$E^{(\phi,\theta,\nu)}[X_{[d]}] = \left\langle \gamma^\phi, \theta\right\rangle \left[X_{[d]}\right]$$

and the cumulant function

$$A^{(\phi,\nu)}(\theta) = \ln\left[\left\langle \nu, \exp\left[\left\langle \gamma^\phi, \theta\right\rangle \left[X_{[d]}\right]\right]\right\rangle [\varnothing]\right] .$$

Since we restrict the discussion to finite state spaces, the distribution $\mathbb{P}^{(\phi,\theta,\nu)}$ is well-defined for any $\theta \in \mathbb{R}^p$. For infinite state space there are sufficient statistics and parameters, such that the partition function $\left\langle \nu, \exp\left[\left\langle \gamma^\phi, \theta\right\rangle \left[X_{[d]}\right]\right]\right\rangle [\varnothing]$ diverges and the normation $\mathbb{P}^{(\phi,\theta,\nu)}$ is not well-defined. In that cases, the canonical parameters need to be chosen from a subset where the partition function is finite.

As before, we restrict for boolean base measures, which satisfy $\langle\nu\rangle[\varnothing] \neq 0$ We notice, that by positivity of the exponential function, any distribution in an exponential family $\Gamma^{\phi,\nu}$ is positive with respect to ν (see Def. 1.3). In Chapter 6 we will investigate distributions, where the base measures and the sufficient statistics share a common computation framework.

Lemma 1.28 *For any member of an exponential family $\Gamma^{\phi,\nu}$ we have*

$$\mathbb{P}^{(\phi,\theta,\nu)}\left[X_{[d]}\right] = \left\langle \exp\left[E^{(\phi,\theta,\nu)}[X_{[d]}] - A^{(\phi,\nu)}(\theta)\cdot \mathbb{I}\left[X_{[d]}\right]\right], \nu^{X_{[d]}}\right\rangle \left[X_{[d]}\right] .$$

Proof. By definition we have

$$\begin{aligned}
\mathbb{P}^{(\phi,\theta,\nu)}\left[X_{[d]}\right] &= \left\langle \exp\left[\left\langle \gamma^\phi, \theta\right\rangle \left[X_{[d]}\right]\right], \nu\left[X_{[d]}\right]\right\rangle \left[X_{[d]}|\varnothing\right] \\
&= \frac{\left\langle \exp\left[\left\langle \gamma^\phi, \theta\right\rangle \left[X_{[d]}\right], \nu\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}\right]}{\left\langle \exp\left[\left\langle \gamma^\phi, \theta\right\rangle \left[X_{[d]}\right]\right], \nu\left[X_{[d]}\right]\right\rangle [\varnothing]} \\
&= \frac{\left\langle \exp\left[E^{(\phi,\theta,\nu)}[X_{[d]}]\right], \nu\left[X_{[d]}\right]\right\rangle \left[X_{[d]}\right]}{\exp\left[A^{(\phi,\nu)}(\theta)\right]} \\
&= \left\langle \exp\left[E^{(\phi,\theta,\nu)}[X_{[d]}] - A^{(\phi,\nu)}(\theta)\cdot \mathbb{I}\left[X_{[d]}\right]\right], \nu^{X_{[d]}}\right\rangle \left[X_{[d]}\right] .
\end{aligned}$$

■

Definition 1.29 (Minimal) We say that a statistic ϕ is minimal with respect to a boolean base measure ν, if there is no pair of a nonvanishing vector $V[L]$ and a scalar $\lambda \in \mathbb{R}$ with

$$\left\langle \gamma^{\phi}\left[X_{[d]}, L\right], V[L], \nu\left[X_{[d]}\right]\right\rangle\left[X_{[d]}\right] = \lambda \cdot \nu\left[X_{[d]}\right] .$$

1.7.1 Tensor Network Representation

We can use the relational encoding formalism to represent members of exponential families by a single contraction, as we show next. The central insight here is a relational encoding of the sufficient statistics, which enables representation by tensor network decomposition, when the sufficient statistic is decomposable.

Theorem 1.30 (Generic Representation of Exponential Families) *Given any base measure ν and a sufficient statistic ϕ we enumerate for each coordinate $l \in [p]$ the image* $\mathrm{im}(\phi_l)$ *by a variable Y_{ϕ_l} taking values in* $[|\mathrm{im}(\phi_l)|]$*, given an interpretation map*

$$I_l : [|\mathrm{im}(\phi_l)|] \to \mathrm{im}(\phi_l) .$$

For any parameter vector $\theta[L] : [p] \to \mathbb{R}$ we build the activation cores

$$W^l[Y_{\phi_l} = x_{\phi_l}] = \exp\left[\theta\left[L = l\right] \cdot I_l(Y_{\phi_l})\right]$$

and have

$$\mathbb{P}^{(\phi,\theta,\nu)} = \left\langle \{\nu, \rho^{\phi}\} \cup \{W^l \,:\, l \in [p]\}\right\rangle\left[X_{[d]}|\varnothing\right] .$$

Proof. We use an extended image of ϕ by

$$\mathrm{im}(\phi) = \bigtimes_{l\in[p]} \mathrm{im}(\phi_l) .$$

Theorem 11.11 implies

$$\exp\left[\left\langle \gamma^{\phi}, \theta\right\rangle\left[X_{[d]}\right]\right] = \left\langle \rho^{\phi}, \exp\left[\langle\cdot, \theta\rangle\right]|_{\mathrm{im}(\phi)}\right\rangle\left[X_{[d]}\right] .$$

The claim follows from the equation

$$\exp\left[\langle\cdot, \theta\rangle\right]|_{\mathrm{im}(\phi)}\left[X_{\phi_0}, \ldots, X_{\phi_{p-1}}\right] = \bigotimes_{l\in[p]} \exp\left[\cdot\theta\left[L = l\right]\right]|_{\mathrm{im}(\phi_l)}\left[X_{\phi_l}\right] = \bigotimes_{l\in[p]} W^l[X_{\phi_l}] .$$

■

We notice, that the relational encoding is the contraction of the relational encoding of its coordinate maps as

$$\rho^{\phi}\left[X_{[d]}, Y_{\phi_{[p]}}\right] = \left\langle \rho^{\phi_0}, \ldots, \rho^{\phi_{p-1}}\right\rangle\left[X_{[d]}, Y_{\phi_{[p]}}\right] .$$

We will show this property in Theorem 12.20. One strategy to create ρ^{ϕ} is thus the creation of the encoding of all its coordinate maps. When the coordinate maps are sharing common components, a sparser representation can be derived through encodings of the components shared among the coordinate map encodings.

A tensor network representation of an exponential family is thus a Markov Network consistent of two types of cores. Computation cores are relational encodings of statistics ρ^{ϕ_l}. Our intuition is that they compute the hidden variable X_{ϕ_l}, based on Basis Calculus (see Chapter 11). Activation cores W^l exploit the computed variable and provide, when contracted with the relational

encoding, a factor

$$\left\langle \rho^{\phi_l}, W^l[X_l] \right\rangle \left[X_{[d]}\right]$$

to the Markov Network reduced to the visible coordinates $X_{[d]}$. The activation cores are trivial, i.e. $W^l[Y_{\phi_l}] = \mathbb{I}\left[Y_{\phi_l}\right]$, when $\theta\left[L = l\right] = 0$. In that case

$$\left\langle \rho^{\phi_l}, W^l[X_l] \right\rangle \left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$$

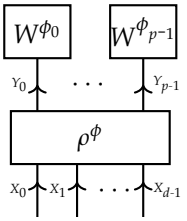
and both the activation core and the corresponding computation core can be dropped from the network without changing its distribution.

By The. 1.30 any member of an exponential family is represented by the normed contraction of a collection of unary activation cores contracted with the computation network $\rho^{\phi}\left[Y_{[p]}, X_{[d]}\right]$. We understand these activation cores as a member of a simple Markov Network distributing the head variables $Y_{[p]}$. This Markov Network has a graph, where the edges contain single variables, that is $\mathcal{G}^{\mathrm{EL}} = ([p], \{\{l\} : l \in [p]\})$. We generalize these classed of probability distributions now to generic tensor network formats representing activation cores.

Definition 1.31 Given a statistic ϕ, and a hypergraph $\mathcal{G} = ([p], \mathcal{E})$ with nodes associated to the coordinates of the statistic, we define the set of realizable distributions by

$$\Lambda^{\phi,\mathcal{G}} = \left\{ \left\langle \{\rho^{\phi}\left[Y_{[p]}, X_{[d]}\right]\} \cup \{T^e\left[Y_e\right]\} \right\rangle \left[X_{[d]}|\varnothing\right] \,:\, T^e\left[Y_e\right] \in \bigotimes_{l \in e} \mathbb{R}^{n_l} \right\}.$$

For unary activation cores, that is for the graph $\mathcal{G}^{\mathrm{EL}}$, we find for any member of $\Lambda^{\phi,\mathrm{EL}}$



For CP, these are mixtures. More general formats are mixtures with further similarity constraints among the mixed distributions. The larger the edges, the more generic: We will show in the next chapter, that with respect to the maximal graph the format contains any distribution.

Corollary 1.32 (Corollary of The. 1.30) *For any base measure ν and statistic ϕ we have*

$$\Gamma^{\phi,\nu} \subset \Lambda^{\phi,\mathcal{G}}.$$

A natural question is to further ask, whether $\Gamma^{\phi,\nu}$ is a proper subset of $\Lambda^{\phi,\mathcal{G}}$. This is the case for most pairs ϕ, ν, since members of exponential families are positive with respect to their base measure, while in $\Lambda^{\phi,\mathcal{G}}$ we allow also for activation cores with vanishing coordinates, which in general do not produce positive distributions. We will follow these intuitions in the discussion of logical reasoning, starting with Chapter 4, and will use the formats $\Lambda^{\phi,\mathcal{G}}$ as hybrid formats storing probability distributions and logical knowledge bases.

Remark 1.33 (Comparison of relation and selection encodings) Relation encodings are in general of higher dimensions than selection encodings. In can thus be intractible to instantiate the probability distribution as a tensor networks, while the energy tensor can still be efficiently represented based on selection encodings. In this case, energy-based reasoning algorithms are tractible while more direct methods are intractible. ♢

1.7.2 Mean Parameters

Mean parameters are an alternative way to represent members of exponential families.

Definition 1.34 Let there be an exponential family defined by a statistic ϕ and a boolean base measure. We call the tensor

$$\mu = \left\langle \mathbb{P}, \gamma^{\phi} \right\rangle [L]$$

the mean parameter tensor to a distribution $\mathbb{P}$ of an exponential family. The set

$$\mathcal{M}_{\phi,\nu} = \left\{ \left\langle \mathbb{P}, \gamma^{\phi}, \nu \right\rangle [L] \,:\, 0 \prec \mathbb{P},\, \langle \mathbb{P}, \nu \rangle \left[X_{[d]} \right] = \nu \left[X_{[d]} \right] \right\},$$

where Γ denotes the set of all probability distributions, is called the convex polytope of realizable mean parameters. The map

$$F^{(\phi,\nu)} : \mathbb{R}^p \to \mathbb{R}^p$$

with $F^{(\phi,\nu)}(\theta) = \left\langle \mathbb{P}^{(\phi,\theta,\nu)}, \gamma^{\phi} \right\rangle [L]$ is called the forward map of the exponential family and any map

$$B^{(\phi,\nu)} : \operatorname{im}\left(F^{(\phi,\nu)} \right) \to \mathbb{R}^p$$

with $\mathbb{P}^{(\phi, B^{(\phi,\nu)}(F^{(\phi,\nu)}(\theta)),\nu)} = \mathbb{P}^{(\phi,\theta,\nu)}$ for any $\theta \in \mathbb{R}^p$ a backward map.

While introduced here as a property of a distribution, the mean parameters will be central to the discussion of probabilistic inference in Chapter 2. We now provide a simple characterization of the sets of mean parameters based on slices of the selection encoding of the statistic.

Theorem 1.35 *For any statistic ϕ the polytope of mean parameters is the convex hull of the slices of γ^{ϕ} with fixed indices to $X_{[d]}$, that is*

$$\mathcal{M}_{\phi,\nu} = \operatorname{conv}\left(\gamma^{\phi}\left[X_{[d]} = x_{[d]}, L \right] \,:\, x_{[d]} \in \bigtimes_{k \in [d]} [m_k],\, \nu\left[X_{[d]} = x_{[d]} \right] = 1 \right).$$

Proof. This follows from the fact, that the set of probability distributions is the convex hull of the one-hot encodings and the convex hull of mean parameters is a linear transform of that set. ∎

The convex polytope $\mathcal{M}_{\phi,\nu}$ can further be characterized as an intersection of half-spaces, as we state next.

Theorem 1.36 *For any statistic ϕ and base measure ν there exists a finite collection $((a_i[L], b_i) \,:\, i \in [n])$ where $a_i[L]$ a vector and $b_i \in \mathbb{R}$ for all $i \in [n]$ such that*

$$\mathcal{M}_{\phi,\nu} = \left\{ \mu[L] \,:\, \forall_{i \in [n]}\, \langle \mu[L], a_i[L] \rangle [\varnothing] \leq b_i \right\}.$$

Proof. This is a standard result of combinatorical optimization, see e.g. [Zie13]. ∎

The determination of the the vectors $((a_i[L], b_i) \,:\, i \in [n])$ and is one reason for the intractability of probabilistic inference (see e.g. [WJ08]).

Definition 1.37 Given a mean parameter polytope $\mathcal{M}_{\phi,\nu}$ in the half space representation of

Theorem 1.36, and any subset $\mathcal{I} \subset [n]$ we say that the set

$$Q^{\mathcal{I}}_{\phi,\nu} = \left\{ \mu\left[L\right] \in \mathcal{M}_{\phi,\nu} \,:\, \forall_{i\in\mathcal{I}} \left\langle \mu\left[L\right], a_i\left[L\right]\right\rangle\left[\varnothing\right] \leq b_i \right\}$$

is the face to the constraints $\mathcal{I}$.

Theorem 1.38 *For any non-empty face $Q^{\mathcal{I}}_{\phi,\nu}$ to a subset $\mathcal{I} \subset [n]$ there is a vector $\theta\left[L\right]$, which we call a normal of the face, such that*

$$Q^{\mathcal{I}}_{\phi,\nu} = \mathrm{argmax}_{\mu\in\mathcal{M}_{\phi,\nu}} \left\langle \theta\left[L\right], \mu\left[L\right]\right\rangle\left[\varnothing\right] .$$

For any collection of positive λ_i, where $i \in \mathcal{I}$, the vector

$$\theta\left[L\right] = \sum_{i\in\mathcal{I}} \lambda_i \cdot a_i\left[L\right]$$

is a normal for $Q^{\mathcal{I}}_{\phi,\nu}$.

Proof. The first claim follows trivially from the second. To show the second claim, let there be for $i \in \mathcal{I}$ arbitrary positive scalars λ_i. We use that the face is non-empty, and thus there is a $\mu\left[L\right]$ with

$$\left\langle \mu\left[L\right], a_i\left[L\right]\right\rangle\left[\varnothing\right] = b_i$$

for all $i \in \mathcal{I}$. Since for any $\mu \in \mathcal{M}_{\phi,\nu}$

$$\left\langle \mu\left[L\right], a_i\left[L\right]\right\rangle\left[\varnothing\right] \leq b_i$$

it follows that

$$\max_{\mu\in\mathcal{M}_{\phi,\nu}} \left\langle \theta\left[L\right], \mu\left[L\right]\right\rangle\left[\varnothing\right] = \sum_{i\in\mathcal{I}} \lambda_i \cdot b_i \ .$$

The maximum is attained at a $\mu\left[L\right]$, if and only if the equations $\left\langle \mu\left[L\right], a_i\left[L\right]\right\rangle\left[\varnothing\right] = b_i$ are satisfied for $i \in \mathcal{I}$. This is equalt to $\mu \in Q^{\mathcal{I}}_{\phi,\nu}$. ∎

In a slide abuse of notation, we denote in this case $Q^{\theta}_{\phi,\nu} = Q^{\mathcal{I}}_{\phi,\nu}$.

1.7.3 Examples

Example 1.39 (The minterm exponential family) When taking as sufficient statistic the identity $\delta\left[X_{[d]}, L_{[d]}\right]$, we can represent any positive distribution $\mathbb{P}$ as a member of the exponential family, namely when choosing the canonical parameter

$$\theta = \ln\left[\mathbb{P}\right] .$$

The associated mean parameter is then, after relabeling the variables $X_{[d]}$ by $L_{[d]}$,

$$\mu = \mathbb{P}$$

and $\mathcal{M}_{\delta,\mathbb{I}}$ coincides with the set of probability tensors. For reasons to be explained in the Chapter 3 we refer to this family as the minterm exponential family. ♢

Given a hypergraph with fixed node decoration, the different decorations of the hyperedges by tensors can be represented by an exponential family, as we show next.

Theorem 1.40 (Exponential Representation of Markov Networks) *For any hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we define a sufficient statistics*

$$\phi = \bigtimes_{e \in \mathcal{E}} \phi_e$$

where

$$\phi_e(x_{\mathcal{V}}) = x_e \,.$$

Given any Markov Network $\{T^e : e \in \mathcal{E}\}$ on $\mathcal{G}$ with positive tensors T^e we define

$$\theta = \bigtimes_{e \in \mathcal{E}} \theta_e$$

where

$$\theta_e = \ln [T^e]$$

and ln *acts coordinatewise. Then, the Markov Network is in the member of the exponential family with trivial base measure, sufficient statistic ϕ and parameters θ.*

Proof. We have for any $x_{\mathcal{V}}$

$$\prod_{e \in \mathcal{E}} T^e [X_e = x_e] = \exp\left[\sum_{e \in \mathcal{E}} \theta_e [X_e = x_e]\right] = \exp\left[\sum_{e \in \mathcal{E}} \langle \theta_e [X_e], \phi_e(x_{\mathcal{V}}) \rangle [\varnothing]\right] .$$

Using that

$$\langle \phi, \theta \rangle [X_{\mathcal{V}}] = \sum_{e \in \mathcal{E}} \langle \phi_e, \theta_e \rangle [X_{\mathcal{V}}]$$

we get

$$\langle \{T^e : e \in \mathcal{E}\} \rangle [X_{\mathcal{V}}] = \exp [\langle \theta, \phi \rangle [X_{\mathcal{V}}]] .$$

This implies

$$\langle \{T^e : e \in \mathcal{E}\} \rangle [X_{\mathcal{V}} | \varnothing] = \langle \exp [\langle \theta, \phi \rangle [X_{\mathcal{V}}]] \rangle [X_{\mathcal{V}} | \varnothing] .$$

■

The mean parameter of the Markov Network exponential family is the cartesian product of the marginals $\mu_e [X_e]$ are often refered to as beliefs in the literature. They are outer bounded by local consistency polytope, which leads to the motivation of message passing algorithms!

1.8 Empirical Distributions

Let us now apply the formalism of probability distributions in tensor network representations to encode data.

Definition 1.41 Given a dataset $\left((x_0^j, \ldots, x_{d-1}^j) : j \in [m]\right)$ of samples of the factored system we define the sample selector map

$$D : [m] \to \bigtimes_{k \in [d]} [m_k]$$

elementwise by

$$D(j) = (x_0^j, \ldots, x_{d-1}^j) .$$

The empirical distribution to the sample selector map D is the probability distribution

$$\mathbb{P}^D\left[X_{[d]}\right] := \left\langle \rho^D \right\rangle \left[X_{[d]} | \varnothing\right] ,$$

where by ρ^D we denote the relational encoding (see Def. 0.19) of the sample selector map, and the distributed variables $X_{[d]}$ are the head variables of the relational encoding.

The relational encoding of the sample selector map is the sum

$$\rho^D\left[J, X_{[d]}\right] = \sum_{j\in[m]} e_j\,[J] \otimes e_{x_0^j,\ldots,x_{d-1}^j}\left[X_{[d]}\right] .$$

Each coordinate $x_{[d]}$ of the empirical distribution can be calculated by

$$\mathbb{P}^D\left[X_{[d]} = x_{[d]}\right] = \frac{1}{\langle \rho^D \rangle\,[\varnothing]}\left(\sum_{j\in[m]} e_{x_0^j,\ldots,x_{d-1}^j}\left[X_{[d]} = x_{[d]}\right]\right) = \frac{\left|j \in [m] \,:\, (x_0^j,\ldots,x_{d-1}^j) = (x_0,\ldots,x_{d-1})\right|}{|j\in[m]|}$$

and is thus interpreted as the frequency of the corresponding world in the data.

The relational encoding of the sample selector map is a sum of one-hot encodings of the data indices and the corresponding sample states. Such sums of basis tensors will be further investigated in Sect. 12.1.2 as basis CP decompositions. We now exploit this structure to find efficient tensor network decompositions (see Figure 1.5) based on matrices encoding its variables.

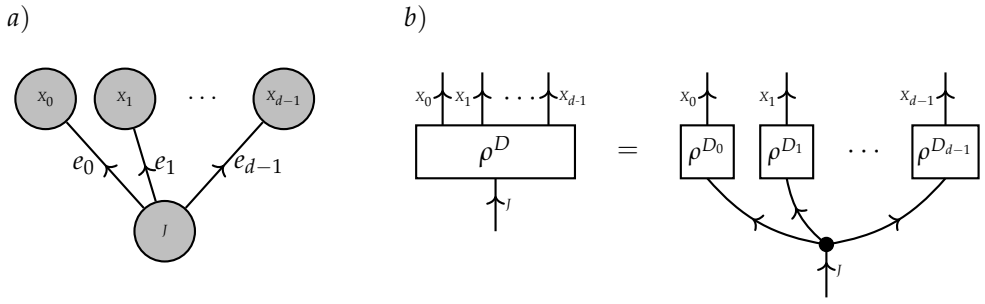


Figure 1.5: Representation of a dataset $\left((x_0^j,\ldots,x_{d-1}^j) \,:\, j\in[m]\right)$. a) Interpretation as a data selection variable J selecting states for the variables $X_{[d]}$. b) Corresponding decomposition of the relational encoding ρ^D into a tensor network in the basis CP Format (see Sect. 12.1.2), where $T^{e_k} = \rho^{D_k}$.

Theorem 1.42 *Given a data map $D : [m] \to \bigtimes_{k\in[d]}[m_k]$ we define for $k \in [d]$ its coordinate maps*

$$D_k : [m] \to [m_k]$$

by

$$D_k(j) = x_k^j .$$

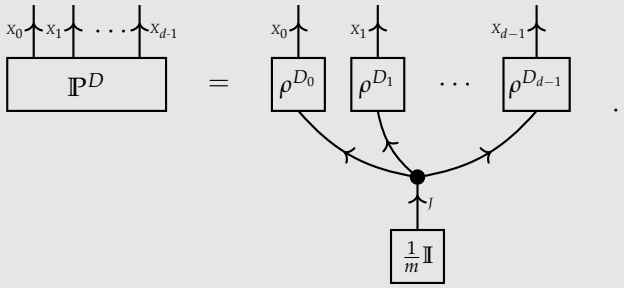
We then have

$$\rho^D\left[J, X_{[d]}\right] = \left\langle \{\rho^{D^k}\,[J, X_k] : k \in [d]\} \right\rangle \left[J, X_{[d]}\right]$$

and

$$\mathbb{P}^D\left[X_{[d]}\right] = \left\langle \rho^D\left[J, X_{[d]}\right], \frac{1}{m}\mathbb{I}[J]\right\rangle\left[X_{[d]}\right] = \left\langle \rho^{D_0}[J, X_0], \ldots, \rho^{D_{d-1}}[X_{d-1}], \frac{1}{m}\mathbb{I}[J]\right\rangle\left[X_{[d]}\right].$$

In a contraction diagram this reads



Proof. The first claim is a special case of Theorem 12.20, to be shown in Chapter 11. To show the second claim we notice

$$\left\langle \rho^D\right\rangle[\varnothing] = \sum_{j\in[m]}\left\langle \rho^D\left[J = j, X_{[d]}\right]\right\rangle[\varnothing] = m\,.$$

With the first claim it now follows that

$$\mathbb{P}^D\left[X_{[d]}\right] = \left\langle \rho^D\right\rangle\left[X_{[d]}|\varnothing\right] = \frac{\left\langle \rho^D\right\rangle\left[X_{[d]}\right]}{\left\langle \rho^D\right\rangle[\varnothing]} = \left\langle \{\rho^{D^k}[J, X_k] : k\in[d]\}\cup\{\frac{1}{m}\mathbb{I}[J]\}\right\rangle\left[J, X_{[d]}\right].$$

■

The cores ρ^{D_k} are matrices storing the value of the categorical variable X_k in the sample world indexed by j.

From the proof of Theorem 1.42 we notice that the scalar $\frac{1}{m}$ could be assigned with any core in a representation of $\mathbb{P}^D$, and the core $\mathbb{I}[J]$ is thus redundant in the contraction representation. However, creating the core $\frac{1}{m}\mathbb{I}[J]$ provides us with a simple interpretation of the empirical distribution. We can understand $\frac{1}{m}\mathbb{I}[J]$ as the uniform probability distribution over the samples, which is by the map D forwarded to a distribution over $\bigtimes_{k\in[d]}[m_k]$. The one-hot encoding of each sample is itself a probability distribution, which is understood as conditioned on the respective state of the sample selection variable J. The conditional distribution ρ^D therefore forwards the uniform distribution of the samples to a distribution of the variables $X_{[d]}$. In the perspective of a Bayesian Network (see Figure 1.5, the variable J served as single parent for each categorical variable X_k.

1.9 Discussion and Outlook

Remark 1.43 (Alternative definitions of graphical models) In the literature, tensor networks are often called dual to the hypergraphs defining graphical models (see e.g. [RS19]). The duality becomes clear, when one interpretes the tensors as cores and their common variables as edges. We in this work avoid this ambiguity by directly defining tensor networks as decoration of hyperedges by tensors.

Often, the tensors decorating hyperedges are called factors and their logarithm features [KF09].

Further, we directly use hypergraphs instead of the more canonical association of factors with cliques of a graph. This avoids the discussion of non-maximal cliques as decorated with trivial tensors. Such hypergraphs follow the same line of though compared with factor graphs, which

are bipartite graphs with nodes either corresponding with single variables or with a collection of them affected by a factor. ◊

2

Probabilitic Inference

We have investigated means to store the knowledge about a system and now turn to the retrieval of information, a process called inference.

Contraction of the relational encoding of a function with a Markov Network gives the statistics over the values of the functions. When contracting the function directly, we get the expectation.

One can increase the efficiency of inference algorithms by using approximative contractions. Here, message passing schemes can be applied as to be introduced in Chapter 13.

2.1 Queries

In the previous chapter, we have derived efficient representation schemes of probability distributions based on tensor network decompositions. We have argued that one should avoid naive instantiation of these distributions based on an storage of each coordinates. In the task of reasoning, we want to retrieve information encoded in the probability distribution. To derive an efficient approach one therefore needs to avoid instantiating the distribution in a coordiantewise manner in an intermediate step. We thus formalize a basic reasoning scheme by contractions of the decomposed distributions with query tensors.

2.1.1 Querying by functions

We can formalize queries by retrieving expectations of functions given a distribution specified by probability tensors. We exploit basis calculus in defining categorical variables X_f to tensors f, which are enumerating the set $\mathrm{im}(f)$. More details on this scheme are provided in Chapter 11, see Def. 11.6 therein.

Definition 2.1 The marginal query of a probability distribution $\mathbb{P}\left[X_{[d]}\right]$ by a tensor

$$f : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R}$$

is the vector $\mathbb{P}\left[X_f\right] \in \mathbb{R}^{|\mathrm{im}(f)|}$ defined as the contraction

$$\mathbb{P}\left[X_f\right] = \left\langle \mathbb{P}\left[X_{[d]}\right], \rho^f\left[X_{[d]}, X_f\right] \right\rangle \left[X_f\right] .$$

The expectation query of $\mathbb{P}$ by f is

$$\mathbb{E}\left[f\right] = \langle f, \mathbb{P}\rangle\left[\varnothing\right] .$$

Given another tensor $g : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}$ the conditional query of the probability distribution $\mathbb{P}\left[X_{[d]}\right]$ by the tensor f conditioned on the tensor g is the matrix $\mathbb{P}\left[X_f|X_g\right] \in$

$\mathbb{R}^{|\mathrm{im}(f)|} \otimes \mathbb{R}^{|\mathrm{im}(g)|}$ defined as the normation

$$\mathbb{P}\left[X_f | X_g\right] = \left\langle \{\mathbb{P}\left[X_{[d]}\right], \rho^f\left[X_{[d]}, X_f\right], \rho^g\left[X_{[d]}, X_g\right]\}\right\rangle \left[X_f | X_g\right] .$$

Expectation queries are contractions of marginal queries with identities, that is

$$\mathbb{E}\left[f\right] = \left\langle \mathbb{P}\left[X_f\right] \mathrm{Id}|_{\mathrm{im}(f)} X_f \right\rangle [\varnothing] .$$

This will be shown in more detail in Chapter 11 in Corollary 11.12.

Conditional probabilities are queries, where the tensors f and g are identity mappings in the respective variable state spaces. Conversely, we can understand the conditional query $\mathbb{P}\left[f|g\right]$ as the conditional probability of f conditioned on g, of the underlying Markov Network with cores $\{\mathbb{P}, \rho^f, \rho^g\}$ and variables X_f, X_g besides the variables distributed by $\mathbb{P}$.

We further denote event queries by

$$\mathbb{E}\left[f = z\right] = \left\langle \mathbb{P}, \rho^f, e_z \right\rangle [\varnothing]$$

where by e_z be denote the one hot encoding of the state z with respect to some enumeration. Let us note that they are further contraction of the queries in Def. 2.1 since by The. 13.1

$$\begin{aligned}\mathbb{E}\left[f = z\right] &= \left\langle \left\langle \mathbb{P}, \rho^f \right\rangle \left[X_f\right], e_z \right\rangle [\varnothing] \\ &= \left\langle \mathbb{P}\left[f\right], e_z \right\rangle [\varnothing] .\end{aligned}$$

2.1.2 MAP Queries

Find the maximal variable of a tensor is a problem, which can be approached by sampling methods as we discuss here.

Definition 2.2 Given a tensor T the MAP query is the problem

$$\mathrm{argmax}_{x_0,\ldots,x_{d-1}} T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] .$$

By coordinate calculus, we notice that

$$T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] \left\langle T, e_{x_0,\ldots,x_{d-1}} \right\rangle [\varnothing] .$$

Given the image Γ^{EL} of one-hot encodings, the MAP query problem is equivalent to

$$\max_{x_0,\ldots,x_{d-1}} T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = \max_{\theta \in \Gamma^{\mathrm{EL}}} \left\langle T, \theta \right\rangle [\varnothing] .$$

We can thus understand MAP queries as a Tensor Network approximation problem, where the approximating tensor are the one-hot encodings of states.

Remark 2.3 (MAP queries on energy and probability tensors) Since the exponential function is monotonic, MAP queries on the energy tensor of an exponential family with uniform base measure are equivalent to MAP queries of their energies. ♢

2.1.3 Answering queries by energy contractions

Let us now interpret a probability tensor at hand as a member of an exponential family (see Sect. 1.7), which is always possible when taking the naive exponential family.

Lemma 2.4 *For any probability distribution $\mathbb{P}$ with $\mathbb{P} = \left\langle \exp\left[E\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$, disjoint subsets $A, B \subset [d]$ with $A \cup B = [d]$ and any x_B we have*

$$\mathbb{P}\left[X_A|X_B = x_B\right] = \left\langle \exp\left[E\left[X_A, X_B = x_B\right]\right]\right\rangle \left[X_A|\varnothing\right] .$$

Proof. Since no summation is commuted. ∎

Thus, it suffices to build the selection encoding of the statistics, and we can avoid the usage of the relational encoding.

We notice, that Lem. 2.4 does not generalize to situations, where $A \cup B \neq [d]$, since summation over the indices of the variables $[d]/A \cup B$ and contraction do not commute.

Lemma 2.5 *For any probability distribution $\mathbb{P}$ with $\mathbb{P} = \left\langle \exp\left[E\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$, disjoint subsets $A, B \subset [d]$ and any x_B we have*

$$\mathbb{P}\left[X_A|X_B = x_B\right] = \left\langle \sum_{x_{[d]/A\cup B} \in [m_{[d]/A\cup B}]} \exp\left[E\left[X_A, X_B = x_B, X_{[d]/A\cup B} = x_{[d]/A\cup B}\right]\right]\right\rangle \left[X_A|\varnothing\right] .$$

Proof. By splitting the contraction into terms to $A \cup B$. ∎

2.2 Sampling based on queries

Let us here investigate how to draw samples from distributions $\mathbb{P}$, based on queries on $\mathbb{P}$.

Since there are $\prod_{v\in\mathcal{V}} m_v$ coordinates stored in $\mathbb{P}$, naive methods are often infeasible. One can instead exploit a representation of $\mathbb{P}$ by a Markov network or the energy term in an exponential family for efficient algorithms and sample from local proxy distributions resulting from contractions and interpreted as marginal and conditional probabilities.

2.2.1 Exact Methods

Forward Sampling (see Algorithm 1) uses a chain decomposition (see The. 1.10) of a probability distribution to iteratively sample the variables.

Algorithm 1 Forward Sampling

for $k \in [d]$ **do**
Draw $x_k \in [m_k]$ from the conditional query

$$\mathbb{P}\left[X_k|X_{\tilde{k}} = x_{\tilde{k}} : \tilde{k} < k\right]$$

end for

Forward Sampling is especially efficient, when sampling from a Bayesian Network respecting the topological order of its nodes. The reason for this lies in trivilizations of all conditional distributions, which heads are not included in the evidence of previously sampled variables. More technically, we can show that

$$\mathbb{P}\left[X_k|X_{\tilde{k}} = x_{\tilde{k}} : \tilde{k} < k\right] = \mathbb{P}\left[X_k|X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\right] ,$$

which is only involving a single core of a Bayesian network. This can be shown using Corollary 11.13 to be derived in Chapter 11.

2.2.2 Approximate Methods

When there are many variables to be sample, the computation of the conditional probability to all variables can be infeasible. One way to overcome this is Gibbs Sampling: Iteratively resemble single variables given the rest as evidence.

Sample each variable independent from the marginal distribution. Then, alternate through the variables and sample each variable from the conditional distribution taking the others as evidence.

Algorithm 2 Gibbs Sampling

for $k \in [d]$ **do**
  Draw State for atom k from initialization distributions.
end for
while Stopping criterion is not met **do**
  for $k \in [d]$ **do**
    Draw $x_k \in [m_k]$ from the conditional query

$$\mathbb{P}\left[X_k | X_{\tilde{k}} = x_{\tilde{k}} : \tilde{k} \neq k\right]$$

  end for
end while

Gibbs can be implemented based on the energy tensor E of the probability tensor, as follows form the Lem. 2.4.

This is in contrast with forward sampling, where we need to sum over many coordinates of the exponentiated energy tensor, which amounts to the representation of the probability distribution as a tensor network using relational encodings.

2.2.3 Simulated Annealing

MAP queries are approximated by sampling from annealed distributions: Use T as the energy tensor, e.g. as parameter tensor to the naive exponential family.

Here by the naive exponential family! Simulated annealing manipulates the probability used to sample x_k in terms of an inverse temperature parameter β, by

$$\mathbb{P} \to \frac{\exp\left[\beta \cdot \ln\left[\mathbb{P}\right]\right]}{\langle \exp\left[\beta \cdot \ln\left[\mathbb{P}\right]\right]\rangle\left[\varnothing\right]} .$$

When the temperature is larger than 1, the probability of states with low probability increases while the probability of states with large probability decreases and for low temperatures the opposite. Simulated annealing, that is the decrease of the temperature to 0 during Gibbs sampling biases the algorithm towards states with large probability.

For any exponential family the transformation

$$E \to \beta \cdot E$$

can be performed by rescaling the canonical parameters as

$$\theta \to \beta \cdot \theta .$$

2.3 Maximum Likelihood Estimation

Let us now turn to inductive reasoning tasks, where a probabilistic model is trained on given data.

2.3.1 Likelihood and Loss

Given a datapoint $D(j)$ consisting of the images of the data selecting map D (see Def. 1.41), the likelihood given a Markov Logic Network is denoted as

$$\mathbb{P}\left[X_{[d]} = D(j)\right] .$$

When all $D(j)$ are drawn independently from $\mathbb{P}\left[X_{[d]}\right]$, we can factorize into

$$\mathbb{P}\left[\{D(j)\}_{j\in[m]}\right] = \prod_{j\in[m]} \mathbb{P}\left[X_{[d]} = D(j)\right] .$$

It is convenient to apply a logarithm on the objective, which does not influence the optimum when optimizing this quantity. This is especially useful, when investigating the convergence of the objective for $m \to \infty$ (see Chapter 8).

Definition 2.6 We define the loss of a distribution $\mathbb{P}$ as

$$\mathcal{L}_D(\mathbb{P}) = \frac{1}{m} \ln\left[\mathbb{P}\left[\{D(j)\}_{j\in[m]}\right]\right]$$

We now state the Maximum Likelihood Problem in the form

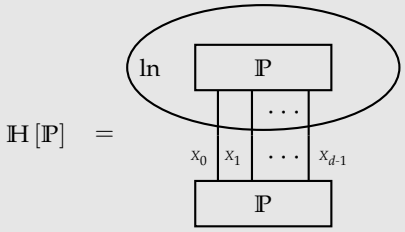
$$\operatorname{argmin}_{\mathbb{P}\in\Gamma} \mathcal{L}_D(\mathbb{P}) . \qquad (\mathrm{P}^{\mathcal{L}_D}_{\Gamma,\mathbb{P}^D})$$

2.3.2 Entropic Interpretation

Definition 2.7 (Shannon entropy) The information content or the Shannon entropy of a distribution is defined as

$$\mathbb{H}[\mathbb{P}] := \mathbb{E}_{X_{[d]}\sim\mathbb{P}}\left[-\ln\left[\mathbb{P}\left[X_{[d]}\right]\right]\right] = \langle \mathbb{P}, -\ln[\mathbb{P}]\rangle[\varnothing] .$$

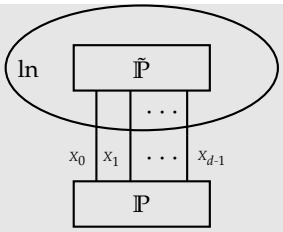
We depict this in a tensor network diagram with an ellipsis denoting a coordinatewise transform (see Chapter 10) with a natural logarithm ln as:



Definition 2.8 (Cross entropy) The cross entropy between two distributions is defined as

$$\mathbb{H}\left[\mathbb{P}, \tilde{\mathbb{P}}\right] := \mathbb{E}_{X_{[d]}\sim\mathbb{P}}\left[-\ln\left[\tilde{\mathbb{P}}[X_{[d]}]\right]\right] = \left\langle \mathbb{P}, -\ln\left[\tilde{\mathbb{P}}\right]\right\rangle[\varnothing] .$$

We depict this in a tensor network diagram with an ellipsis denoting a coordinatewise transform (here the ln) as :

$\mathbb{H}\left[\mathbb{P}, \tilde{\mathbb{P}}\right] =$ 

We here use $\ln[0] = -\infty$ and $0 \cdot \ln[0] = 0$. Then we have $\mathbb{H}\left[\mathbb{P}, \tilde{\mathbb{P}}\right] = \infty$ if and only if there is a $x_{[d]}$ such that $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0$ and $\tilde{\mathbb{P}}[X_{[d]} = x_{[d]}] = 0$.

The Gibbs inequality states that

$$\mathbb{H}\left[\mathbb{P}, \tilde{\mathbb{P}}\right] \geq \mathbb{H}\left[\mathbb{P}\right] .$$

The difference between both sides is called the Kullback Leibler Divergence and a useful metric in reasoning, since it vanishes for $\mathbb{P} = \tilde{\mathbb{P}}$.

Definition 2.9 (Kullback Leibler Divergence) The KL divergence between two distributions is defined as

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}||\tilde{\mathbb{P}}\right] = \mathbb{H}\left[\mathbb{P}, \tilde{\mathbb{P}}\right] - \mathbb{H}\left[\mathbb{P}\right] .$$

We are now ready to provide an entropic interpretation of the loss introduced in Def. 2.6.

Theorem 2.10 *Given a data selecting map D and a distribution $\mathbb{P}$ we have*

$$\mathcal{L}_D\left(\mathbb{P}\right) = \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}\right] .$$

Proof. We have

$$\begin{aligned}\mathcal{L}_D\left(\mathbb{P}\right) &= \frac{1}{m} \ln\left[\mathbb{P}\left[\left\{D(j)\right\}_{j\in[m]}\right]\right] = \frac{1}{m}\sum_{j\in[m]} \ln\left[\mathbb{P}\left[X_{[d]} = D(j)\right]\right] = \frac{1}{m}\sum_{j\in[m]} \left\langle \left\{\ln\left[\mathbb{P}\right], e_{D(j)}\right\}\right\rangle [\varnothing] \\ &= \left\langle \mathbb{P}^D, \ln\left[\mathbb{P}\right]\right\rangle [\varnothing] .\end{aligned}$$

Comparing with the negative log likelihood we notice that that loss coincides with the cross-entropy between the empirical distribution $\mathbb{P}^D$ and $\mathbb{P}$, i.e.

$$\mathcal{L}_D\left(\mathbb{P}\right) = \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}\right] .$$

■

We can therefore rewrite Problem $\mathrm{P}^{\mathcal{L}_D}_{\Gamma,\mathbb{P}^D}$ as minimization of cross-entropies and of Kullback Leibler divergences as

$$\mathrm{argmin}_{\mathbb{P}\in\Gamma}\mathcal{L}_D\left(\mathbb{P}\right) = \mathrm{argmin}_{\mathbb{P}\in\Gamma}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}\right] = \mathrm{argmin}_{\mathbb{P}\in\Gamma}\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^D||\mathbb{P}\right] .$$

Most general, the Maximum Likelihood Problem is the M-Projection of a distribution $\mathbb{P}^*$ onto a set Γ of probability tensors is

$$\mathrm{argmax}_{\mathbb{P}\in\Gamma}\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}\right]$$

where the Maximum Likelihood Estimation is the special case $\mathbb{P}^* = \mathbb{P}^D$.

Example 2.11 (Cross entropy with respect to exponential families) If $\tilde{\mathbb{P}}$ from an exponential family with boolean base measure, have with the representation from Lem. 1.28

$$\mathbb{H}\left[\mathbb{P}, \mathbb{P}^{(\phi,\theta,\nu)}\right] = \left\langle \mathbb{P}, \ln\left[\mathbb{P}^{(\phi,\theta,\nu)}\right]\right\rangle[\varnothing] = \left\langle \mathbb{P}, \gamma^{\phi}\right\rangle[\varnothing] - A^{(\phi,\nu)}(\theta) + \left\langle \mathbb{P}, \ln[\nu]\right\rangle[\varnothing] \ .$$

For the trivial base measure we can further exploit the existence of the energy tensor and have the representation

$$\mathbb{H}\left[\mathbb{P}, \mathbb{P}^{(\phi,\theta,\nu)}\right] = \left\langle \mathbb{P}, (E^{(\phi,\theta,\nu)}[X_{[d]}] - A^{(\phi,\nu)}(\theta)\cdot \mathbb{I})\right\rangle[\varnothing] = \left\langle \mathbb{P}, E^{(\phi,\theta,\nu)}[X_{[d]}]\right\rangle[\varnothing] - A^{(\phi,\nu)}(\theta) \, .$$

$\lozenge$

2.4 Forward Mapping in Exponential Families

Mean parameter coordinates are expectation queries to ϕ_l, by

$$\mu[L = l] = \mathbb{E}[\phi_l] \ .$$

Forward mappings have a closed form representation by

$$F^{(\phi,\nu)}(\theta) = \left\langle \gamma^{\phi}, \left\langle \nu, \exp\left[\left\langle \gamma^{\phi}, \theta\right\rangle[\varnothing]\right]\right\rangle \left[X_{[d]}|\varnothing\right]\right\rangle[L] \ .$$

This contraction can, however, be infeasible, since it requires the instantiation of the probability tensor, which can be done by basis encodings of the statistic. We in this section provide alternative characterization of the forward map and approximations of it, which can be computed based on the selection encoding instead. Following [WJ08], we can characterize the forward mapping to exponential families as a variational problem and provide an alternative characterization to this contraction.

2.4.1 Variational Formulation

Besides the direct computation of the mean parameter tensor we can give a variational characterization of the forward mapping. This is especially useful, when the contraction is intractable, for example because the tensor $\mathbb{P}^{(\phi,\theta,\nu)}$ is infeasible to create.

Theorem 2.12 *We have*

$$F^{(\phi,\nu)}(\theta) = \operatorname{argmax}_{\mu\in\mathcal{M}_{\phi,\nu}} \left\langle \mu, \theta\right\rangle[\varnothing] + \mathbb{H}[\mathbb{P}^{\mu}]$$

where by $\mathbb{P}^{\mu}$ we denote a probability distribution with respect to a base measure ν, which reproduces the mean parameter μ.

Proof. Theorem 3.4 in [WJ08]. ∎

Let us now characterize the image of the forward map, which turns out to be the interior of the mean polytope, if the statistic is minimal (see Def. 1.29).

Theorem 2.13 *For any statistics ϕ, which is minimal with respect to a base measure ν, the image $\operatorname{im}\left(F^{(\phi,\nu)}\right)$ of the forward map is the interior of the convex polytope $\mathcal{M}_{\phi,\nu}$.*

Proof. Theorem 3.3 in [WJ08]. ∎

For the practicle usage of this theorem, we need a characterization of the interior of $\mathcal{M}_{\phi,\nu}$.

Theorem 2.14 *For any minimal statistics ϕ and boolean base measure ν we have for some $\mu[L]$ that $\mu[L] \in \mathcal{M}_{\phi,\nu}$ if and only if there is a positive distribution with respect to ν such that*

$$\mu[L] = \langle \mathbb{P}, \gamma^{\phi} \rangle [L] .$$

Proof. "⇒": By The. 2.13 we find a canonical parameter $\theta[L]$ such that

$$\mu[L] = \left\langle \mathbb{P}^{(\phi,\theta,\nu)} \left[X_{[d]} \right], \gamma^{\phi} \left[X_{[d]}, L \right] \right\rangle [L] .$$

We notice, that $\mathbb{P}^{(\phi,\theta,\nu)}$ is positive with respect to ν, as is any member of an exponential family with base measure ν.

"⇐": Since by assumption the statistics is minimal, the convex set $\mathcal{M}_{\phi,\nu}$ is full dimensional (see e.g. Appendix B in [WJ08]). We thus use a well-known property for full-dimensional convex sets (see [Roc97; HL93]), that $\mu \in \mathcal{M}_{\phi,\nu}{}^{\circ}$ if for any non-vanishing vector $V[L]$ there is a there is a $\tilde{\mu}[L]$ with

$$\langle V[L], \mu[L] \rangle [\varnothing] < \langle V[L], \tilde{\mu}[L] \rangle [\varnothing] .$$

It thus suffices to show for an arbitrary non-vanishing vector $V[L]$ the existence of a distribution $\tilde{\mathbb{P}}$, such that

$$\langle V[L], \mu[L] \rangle [\varnothing] < \left\langle V[L], \gamma^{\phi} \left[X_{[d]}, L \right], \tilde{\mathbb{P}}[X_{[d]}] \right\rangle [\varnothing] .$$

We define for $\epsilon \in \mathbb{R}$

$$\mathbb{P}^{\epsilon} \left[X_{[d]} \right] = \left\langle \mathbb{P} \left[X_{[d]} \right], \exp \left[\epsilon \cdot \left\langle \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle \left[X_{[d]} \right] \right] \right\rangle \left[X_{[d]} | \varnothing \right]$$

The derivation of this map at $\epsilon = 0$ is

$$\frac{\partial}{\partial \epsilon} \mathbb{P}^{\epsilon} \left[X_{[d]} \right] |_{\epsilon=0} = \left\langle \mathbb{P} \left[X_{[d]} \right], \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle \left[X_{[d]} \right] - \left\langle \mathbb{P} \left[X_{[d]} \right], \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle [\varnothing] \cdot \mathbb{P} \left[X_{[d]} \right]$$

and thus

$$\begin{aligned} \frac{\partial}{\partial \epsilon} \left\langle \mathbb{P}^{\epsilon} \left[X_{[d]} \right], \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle [\varnothing] |_{\epsilon=0} = & \left\langle \mathbb{P} \left[X_{[d]} \right], (\left\langle \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle [])^2 \right\rangle \left[X_{[d]} \right] \\ & - \left(\left\langle \mathbb{P} \left[X_{[d]} \right], \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle \left[X_{[d]} \right] \right)^2 . \end{aligned}$$

We can interpret this quantity as the variance of the random variable $\left\langle \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle \left[X_{[d]} = x_{[d]} \right]$, where $x_{[d]}$ is drawn from $\mathbb{P} \left[X_{[d]} \right]$. The variance is greater than zero, if this random variable is not constant. But from the minimality of ϕ with respect to ν it follows, that this variable is not constant and we therefore have

$$0 < \frac{\partial}{\partial \epsilon} \left\langle \mathbb{P}^{\epsilon} \left[X_{[d]} \right], \gamma^{\phi} \left[X_{[d]}, L \right], V[L] \right\rangle [\varnothing] |_{\epsilon=0} .$$

Thus, there is a $\epsilon > 0$ with

$$\langle V[L], \mu[L] \rangle [\varnothing] < \left\langle V[L], \gamma^{\phi} \left[X_{[d]}, L \right], \mathbb{P}^{\epsilon} \left[X_{[d]} \right] \right\rangle [\varnothing] .$$

■

2.4.2 Boundary of convex polytopes

For mean parameters $\mu[L]$ outside the interior of $\mathcal{M}_{\phi,\nu}$ we know by The. 2.14, that any distribution with mean parameter $\mu[L]$ is not positive with respect to ν and is therefore not in the exponential family. We investigate this situation further and provide here a construction scheme to adapt the base measure such that there are exponential families containing these boundary distributions.

Theorem 2.15 *Let there be a minimal ϕ with respect to the base measure ν and $\mu[L] \notin \mathcal{M}_{\phi,\nu}{}^{\circ}$. Then there is a $\theta[L]$ with*

$$\mu[L] \in \operatorname{argmax}_{\mu \in \mathcal{M}_{\phi,\nu}} \langle \theta[L], \mu[L] \rangle [\varnothing]$$

and all distributions with mean parameter $\mu[L]$ are representable with respect to the base measure

$$\tilde{\nu}\left[X_{[d]}\right] = \left\langle \nu, \mathbb{I}_{\mathcal{U}}\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right],$$

where the indicator is on the set

$$\mathcal{U} = \operatorname{argmax}_{x_{[d]}} \left\langle \theta, \phi(x_{[d]}) \right\rangle [\varnothing].$$

Proof. When $\mu \notin \mathcal{M}_{\phi,\nu}{}^{\circ}$ we find a face such that $\mu \in Q^{\mathcal{I}}_{\phi,\nu}$. The existence of $\theta[L]$ follows from The. 1.38, in which also a construction procedure is provided given a half-space representation (see The. 1.36).

Now, we have

$$\mu[L] \in \operatorname{argmax}_{\mu \in \mathcal{M}_{\phi,\nu}} \langle \theta[L], \mu[L] \rangle [\varnothing]$$

and thus

$$\mu[L] \in \operatorname{conv}\left(\gamma^{\phi} X_{[d]} = x_{[d]}, L \,:\, x_{[d]} \in \operatorname{argmax}_{x_{[d]} : \nu\left[X_{[d]} = x_{[d]}\right] = 1} \left\langle \theta[L], \gamma^{\phi} X_{[d]} = x_{[d]}, L \right\rangle [\varnothing]\right)$$

Thus, any distribution reproducing meanparam is a convex combination of the one-hot encodings of the states in $\operatorname{argmax}_{x_{[d]}} \left\langle \theta[L], \gamma^{\phi} X_{[d]} = x_{[d]}, L \right\rangle [\varnothing]$, and therefore representable with respect to the base measure $\tilde{\nu}$. ∎

Each face of $\mathcal{M}_{\phi,\nu}$ thus defines a refinement of a base measure, which is sufficient to reproduce the mean parameters on that face.

Definition 2.16 The base measure to the face of $\mathcal{M}$ with normal θ is

$$\nu^{\phi,\theta} = \mathbb{I}_{\operatorname{argmax}_{x_{[d]}} \langle \theta, \phi(x_{[d]}) \rangle [\varnothing]}\left[X_{[d]}\right].$$

The. 2.15 therefore states, that when a mean parameter is on a face of $\mathcal{M}_{\phi,\nu}$, then each distribution reproducing the mean parameter has a representation with respect to the refined base measure

$$\tilde{\nu}\left[X_{[d]}\right] = \left\langle \nu, \nu^{\phi,\theta} \right\rangle \left[X_{[d]}\right].$$

We now utilize these findings and provide in Algorithm 3 a procedure to refine the base measure until the reduced mean parameter is in the open set of a reduced mean parameter polytope.

Algorithm 3 Base Measure Refinement

Input: Base measure ν, statistic ϕ and mean parameter $\mu \in \mathcal{M}_{\phi,\nu}$
Output: Refined base measure $\tilde{\nu}$, remaining statistic $\tilde{\phi}$ and remaining mean parameter $\tilde{\mu}$

while $\mu \notin \left(\mathcal{M}_{\phi,\nu}\right)^{\circ}$ **do**
  while ϕ not minimal with respect to ν (see Def. 1.29) **do**
    Find non-vanishing vector $V[L]$ and scalar $\lambda \in \mathbb{R}$ such that

$$\left\langle \gamma^{\phi}\left[X_{[d]}, L\right], V[L], \nu\left[X_{[d]}\right]\right\rangle \left[X_{[d]}\right] = \lambda \cdot \nu\left[X_{[d]}\right] .$$

    Choose a coordinate $l \in [p]$ with $V[L = l] \neq 0$ and drop it from ϕ and μ
  end while
  Find a non-trivial face (i.e. a non-empty face, which is a proper subset of $\mathcal{M}_{\phi,\nu}$) with normal θ, such that

$$\mu \in Q^{\theta}_{\phi,\nu}$$

  Refine base measure

$$\nu \leftarrow \left\langle \nu, \nu^{\phi,\theta}\right\rangle \left[X_{[d]}\right]$$

end while
return ν, ϕ, μ

Theorem 2.17 *For arbitrary inputs ν, ϕ and $\mu \in \mathcal{M}_{\phi,\nu}$, Algorithm 3 terminates in finite time and outputs a triple of base measure $\tilde{\nu}$, statistic $\tilde{\phi}$ and mean parameter $\tilde{\mu}$ such that the following holds. Any probability tensor $\mathbb{P}$ reproducing μ is representable with respect to $\tilde{\nu}$ and $\tilde{\mu} \in \left(\mathcal{M}_{\tilde{\phi},\tilde{\nu}}\right)^{\circ}$.*

Proof. Let us first show, that Algorithm 3 always terminates. The inner while loop of Algorithm 3 always terminates, since ϕ has a finite number of coordinates, and in each iteration one of the coordinates is dropped. To show that the outer while loop also terminates, it suffices to show, that the non-vanishing coordinates of the refined base measure are a proper subset of the base measure before refinement. But if this would not be the case, we would have

$$\nu\left[X_{[d]}\right] = \left\langle \nu, \nu^{\phi,\theta}\right\rangle \left[X_{[d]}\right]$$

and thus $Q^{\theta}_{\phi,\nu} = \mathcal{M}_{\phi,\nu}$, which is a contradiction with the assumption of a non-trivial face.

The second claim follows from an iterative application of The. 2.15 and the fact, that a probability distribution reproduces μ in a non-minimal representation, if and only if it reproduces the corresponding reduced μ with respect to the reduced statistics. ∎

Example 2.18 (Faces with normals parallel to one-hot encodings) To get some intuition how to represent face base measures, let us consider face normals $\theta \in \{\lambda \cdot e_l\,[L] \,:\, l \in [p],\, \lambda \in \mathbb{R}/\{0\}\}$. We use relational encodings of the coordinates ϕ_l of the statistic ϕ, with head variables X_{ϕ_l} with dimension m_{ϕ_l} enumerating the image $\mathrm{im}\,(\phi_l) \subset \mathbb{R}$ in an ascending order. If $\theta\,[L] = \lambda \cdot e_l\,[L]$ with $\lambda > 0$, then $\mathrm{argmax}_{x_{[d]}} \left\langle \theta, \phi(x_{[d]})\right\rangle [\varnothing]$ consists of states $x_{[d]}$ with minimal statistic $\phi_l\left[X_{[d]} = x_{[d]}\right]$, that is

$$\nu^{\phi,\lambda\cdot e_l}\left[X_{[d]}\right] = \left\langle \rho^{\phi_l}\left[X_{[d]}, X_{\phi_l}\right], e_{m_{\phi_l}-1}\left[X_{\phi_l}\right]\right\rangle \left[X_{[d]}\right] .$$

If $\theta\,[L] = \lambda \cdot e_l\,[L]$ with $\lambda < 0$, then at the states with minimal statistic $\phi_l\left[X_{[d]} = x_{[d]}\right]$, that is

$$\nu^{\phi,\lambda\cdot e_l}\left[X_{[d]}\right] = \left\langle \rho^{\phi_l}\left[X_{[d]}, X_{\phi_l}\right], e_0\left[X_{\phi_l}\right]\right\rangle \left[X_{[d]}\right] .$$

◊

Theorem 2.19 *For the maximal graph $\mathcal{G}^{\max} = ([p], \{[p]\})$, which has a single hyperedge containing all head variables we have*

$$\mathcal{M}_{\phi,\nu} = \left\{ \left\langle \mathbb{P}\left[X_{[d]}\right], \gamma^{\phi}\left[X_{[d]}, L\right] \right\rangle \left[X_{[d]}\right] \ , \mathbb{P} \in \ \Lambda^{\phi,\mathcal{G}^{\max}} \right\}$$

Proof. It is enough show, that for any output tuples $\tilde{v}$, $\tilde{\phi}$ of the Base Measure Refinement Algorithm 3 we have

$$\Gamma^{\tilde{v},\tilde{\phi}} \subset \Lambda^{\phi,\mathcal{G}^{\max}}.$$

We notice, that the normation of any face base measure is realizable by $\Lambda^{\phi,\mathcal{G}^{\max}}$, since the objective in the maximation problem in Def. 2.16 depends only on ϕ. Providing a more technical argument, we have

$$\mathbb{I}_{\operatorname{argmax}_{x_{[d]}} \langle \theta, \phi(x_{[d]}) \rangle [\varnothing]}\left[X_{[d]}\right] = \left\langle \rho^{\phi}, \sum_{\phi(x_{[d]}) \, : \, x_{[d]} \in \operatorname{argmax}_{x_{[d]}} \langle \theta, \phi(x_{[d]}) \rangle [\varnothing]} e_{I(\phi(x_{[d]}))}\left[Y_{[p]}\right] \right\rangle \left[X_{[d]}\right].$$

Since during the execution of Algorithm 3, $\tilde{\phi}$ is a subset of ϕ, we can find a corresponding θ_i extending the face normal by vanishing coordinates to ϕ. We then have, that

$$\tilde{v} = \left\langle \{\rho^{\phi}\left[Y_{[p]}, X_{[d]}\right]\} \cup \left\{ \sum_{\phi(x_{[d]}) \, : \, x_{[d]} \in \operatorname{argmax}_{x_{[d]}} \langle \theta_i, \phi(x_{[d]}) \rangle [\varnothing]} e_{I(\phi(x_{[d]}))}\left[Y_{[p]}\right] : i \in [n] \right\} \right\rangle \left[Y_{[p]}\right]$$

represents the output base measure, where $i \in [n]$ label the faces chosen during in the loop of Algorithm 3. Now, any member $\mathbb{P}^{\tilde{v},\theta,\tilde{\phi}} \in \Gamma^{\tilde{\phi},\tilde{v}}$ can be represented by a member of $\Lambda^{\phi,\mathcal{G}^{\max}}$, by contracting these base measure representing cores with the activation cores $\bigotimes_{l \in [p]} W^{\phi_l, \theta[L=l]}[Y_l]$. ■

2.4.3 Mode Search by annealing

Finding the mode of a distribution is related to the forward mapping of $\beta \cdot \theta$: μ to a delta distribution (or in the convex hull of multiple maxima) in the limit.

This is because

$$\operatorname{argmax}_{\mu \in \mathcal{M}_{\phi,\nu}} \langle \mu, \theta \rangle [\varnothing]$$

is taken at an extreme point in $\mathcal{M}_{\phi,\nu}$ (since linear objective over closed convex set), which is a delta distribution of a set and

$$\operatorname{argmax}_{\mu \in \mathcal{M}_{\phi,\nu}} \langle \mu, \beta \cdot \theta \rangle [\varnothing] + \mathbb{H}\left[\mathbb{P}^{\mu}\right] = \operatorname{argmax}_{\mu \in \mathcal{M}_{\phi,\nu}} \langle \mu, \theta \rangle [\varnothing] + \frac{1}{\beta} \cdot \mathbb{H}\left[\mathbb{P}^{\mu}\right]$$

thus the entropy term is neglectible for large β. A more precise argument is using a limit of the maxima and can be found in Theorem 8.1 in [WJ08]

2.4.4 Mean Field Method

We rewrite

$$\max_{\mu \in \mathcal{M}_{\phi,\nu}} \langle \mu, \theta \rangle [\varnothing] + \mathbb{H}\left[\mathbb{P}^{\mu}\right] = \max_{\mathbb{P}} \langle E, \mathbb{P} \rangle [\varnothing] + \mathbb{H}\left[\mathbb{P}\right]$$

where

$$E = \left\langle \gamma^{\phi}, \theta \right\rangle \left[X_{[d]}\right].$$

We now restrict the distributions in the maximum. Typically we use the family of independent distributions, also called naive mean field method. The naive mean field is the approximation by distributions of independent random variables V^k, that is

$$\operatorname{argmax}_{V^k : k\in[d]} \left\langle \{E\} \cup \{V^k \,:\, k \in [d]\} \right\rangle [\varnothing] + \sum_{k\in[d]} \mathbb{H}\left[V^k\right] .$$

Theorem 2.20 (Update equations for the mean field approximation) *Keeping all legs but one constant, the problem*

$$\operatorname{argmax}_{V^k} \left\langle \{E\} \cup \{V^k \,:\, k \in [d]\} \right\rangle [\varnothing] + \sum_{k\in[d]} \mathbb{H}\left[V^k\right]$$

is solved at

$$V^k[X_k] = \left\langle \exp\left[\left\langle \{E[X_{[d]}]\} \cup \{V^{\tilde{k}}[X_{\tilde{k}}] \,:\, \tilde{k} \neq k\} \right\rangle \left[X_{[d]}\right]\right] \right\rangle [X_k|\varnothing] .$$

Proof. We have

$$\frac{\partial \mathbb{H}\left[V^k\right]}{\partial V^k} = -\ln\left[V^k[X_k]\right] + \mathbb{I}\left[X_k\right]$$

and by multilinearity of tensor contractions

$$\frac{\partial \left\langle \{E\} \cup \{V^{\tilde{k}} \,:\, \tilde{k} \in [d]\} \right\rangle [\varnothing]}{\partial V^k} = \left\langle \{E\} \cup \{V^{\tilde{k}} \,:\, \tilde{k} \in [d],\, \tilde{k} \neq k\} \right\rangle [X_k] .$$

Combining both, the condition

$$0 = \frac{\partial \left(\left\langle \{E\} \cup \{V^{\tilde{k}} \,:\, \tilde{k} \in [d]\} \right\rangle [\varnothing] + \sum_{k\in[d]} \mathbb{H}\left[V^k\right] \right)}{\partial V^k}$$

is equal to

$$\ln\left[V^k[X_k]\right] = \mathbb{I}\left[X_k\right] + \left\langle \{E\} \cup \{V^{\tilde{k}} \,:\, \tilde{k} \in [d],\, \tilde{k} \neq k\} \right\rangle [X_k] .$$

Together with the condition $\left\langle V^k \right\rangle [=] 1$ this is satisfied at

$$V^k[X_k] = \left\langle \exp\left[\left\langle \{E\} \cup \{V^{\tilde{k}} \,:\, \tilde{k} \neq k\} \right\rangle [X_k]\right] \right\rangle [X_k|\varnothing] .$$

■

Algorithm 4 is the alternation of legwise updates until a stopping criterion is met.

2.4.5 Structured Variational Approximation

More generically, we restrict the maximum over the mean parameters of efficiently contractable distributions and get a lower bound. In this section we use any Markov Network as the approximating family.

Let $\mathcal{G}$ be any hypergraph, we define the problem

$$\operatorname{argmax}_{\mathbb{P}\in\Gamma^{\mathcal{G},\mathbb{I}}} \langle E, \mathbb{P}\rangle [\varnothing] + \mathbb{H}\left[\mathbb{P}\right]$$

Algorithm 4 Naive Mean Field Approximation

for $k \in [d]$ **do**

$$V^k[X_k] \leftarrow \langle \mathbb{I} \rangle [X_k|\varnothing]$$

end for
while Stopping criterion is not met **do**
  for $k \in [d]$ **do**

$$V^k[X_k] \leftarrow \left\langle \exp\left[\left\langle \{E[X_{[d]}]\} \cup \{V^{\tilde{k}}[X_{\tilde{k}}] : \tilde{k} \neq k\} \right\rangle [X_k]\right] \right\rangle [X_k|\varnothing]$$

  end for
end while

We approximate the solution of this problem again by an alternating algorithm, which iteratively updates the cores of the approximating Markov Network.

Theorem 2.21 (Update equations for the structured variational approximation) *The Markov Network $\mathcal{T}^{\mathcal{G}}$ with hypercores $\{T^e : e \in \mathcal{E}\}$ is a stationary point for Problem $\mathrm{P}_{\Gamma^{\mathcal{G}},\mathbb{I},\mathbb{P}}$, if for all $e \in \mathcal{E}$*

$$T^e[X_e] = \lambda \cdot \exp\left[\frac{\langle \{E\} \cup \{T^{\tilde{e}} : \tilde{e} \neq e\} \rangle [X_e]}{\langle \{T^{\tilde{e}} : \tilde{e} \neq e\} \rangle [X_e]} - \sum_{\hat{e} \neq e} \frac{\langle \{\ln[T^{\hat{e}}]\} \cup \{T^{\tilde{e}} : \tilde{e} \neq \hat{e}\} \rangle [X_e]}{\langle \{T^{\tilde{e}} : \tilde{e} \neq \hat{e}\} \rangle [X_e]}\right]$$

for any $\lambda > 0$ (e.g. by the norm). Here, the quotient denotes the coordinatewise quotient.

Proof. We proof the theorem by first order condition on the objective $O\left(\mathcal{T}^{\mathcal{G}}\right) = \left\langle E, \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}|\varnothing] \right\rangle [\varnothing] + \mathbb{H}\left[\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}|\varnothing]\right]$.

To proof the theorem, we use Lem. 10.18, which shows a characterization of the derivative of functions

We have

$$\left\langle E, \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{[d]}|\varnothing\right] \right\rangle [\varnothing] = \frac{\langle \{E\} \cup \mathcal{T}^{\mathcal{G}} \rangle [\varnothing]}{\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing]}.$$

Further we have

$$\mathbb{H}\left[\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{[d]}|\varnothing\right]\right] = \left(\sum_{\tilde{e} \in \mathcal{E}} \left\langle -\ln\left[T^{\tilde{e}}\right], \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{[d]}|\varnothing\right] \right\rangle [\varnothing]\right) + \ln\left[\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing]\right]$$

We define the tensor

$$\tilde{T}[X_{\mathcal{V}}] = E[X_{\mathcal{V}}] - \sum_{\tilde{e} \neq e} \ln\left[T^{\tilde{e}}[X_{\tilde{e}}]\right] \otimes \mathbb{I}[X_{\mathcal{V}/\tilde{e}}]$$

and notice, that $\tilde{T}$ does not depend on T^e.

The objective has then a representation as

$$O\left(\mathcal{T}^{\mathcal{G}}\right) = \left\langle \tilde{T}[X_{\mathcal{V}}], \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}|\varnothing] \right\rangle [\varnothing] - \left\langle \ln[T^e], \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}|\varnothing] \right\rangle [\varnothing] + \ln\left[\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing]\right]$$

Let us now differentiate all terms. With Lem. 10.18 we now get

$$\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \tilde{T}[X_{\mathcal{V}}],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]=\left\langle \tilde{T}\left[X_{\mathcal{V}}\right],\delta\left[Y_e,X_e\right],\frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_e]}{T^e\left[X_e\right]},\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}/e}|X_e]\right\rangle [Y_e,X_{\mathcal{V}}]$$
$$-\left\langle \tilde{T}\left[X_{\mathcal{V}}\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]\otimes\left\langle \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [Y_e]}{T^e\left[Y_e\right]}\right\rangle [Y_e]\,.$$

Further we have

$$\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \ln\left[T^e\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]=\left\langle \ln\left[T^e\left[X_e\right]\right],\delta\left[Y_e,X_e\right],\frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_e]}{T^e\left[X_e\right]},\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}/e}|X_e]\right\rangle [Y_e,X_{\mathcal{V}}]$$
$$-\left\langle \ln\left[T^e\left[X_e\right]\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]\otimes\left\langle \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [Y_e]}{T^e\left[Y_e\right]}\right\rangle [Y_e]$$
$$-\left\langle \frac{1}{T^e\left[X_e\right]},\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]$$

and (see Proof of 10.17)

$$\frac{\partial}{\partial T^e\left[Y_e\right]}\ln\left[\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [\varnothing]\right]=\frac{\frac{\partial}{\partial T^e[Y_e]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [\varnothing]}{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [\varnothing]}=\frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [Y_e]}{T^e\left[Y_e\right]}\,.$$

Together, the first order condition

$$0=\frac{\partial}{\partial T^e\left[Y_e\right]}O\left(\mathcal{T}^{\mathcal{G}}\right)$$

is equal to all y_e satisfying

$$0=\frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [Y_e=y_e]}{T^e\left[Y_e=y_e\right]}\Big(\left\langle \tilde{T}\left[X_{\mathcal{V}/e},X_e=y_e\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}/e}|X_e=y_e]\right\rangle [\varnothing]$$
$$-\left\langle \tilde{T}\left[X_{\mathcal{V}}\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]$$
$$-\left\langle \ln\left[T^e\left[X_e=y_e\right]\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}/e}|X_e=y_e]\right\rangle [\varnothing]$$
$$+\left\langle \ln\left[T^e\left[X_e\right]\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]\Big)\,.$$

We notice, that by normation

$$\left\langle \ln\left[T^e\left[X_e=y_e\right]\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}/e}|X_e=y_e]\right\rangle [\varnothing]=\ln\left[T^e\left[X_e=y_e\right]\right]$$

and that the scalar

$$\lambda_1=\left\langle \tilde{T}\left[X_{\mathcal{V}}\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]-\left\langle \ln\left[T^e\left[X_e\right]\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}|\varnothing]\right\rangle [\varnothing]$$

is the constant for all y_e.

The first order condition is therefore equal to the existence of a $\lambda_1\in\mathbb{R}$ such that for all y_e

$$\ln\left[T^e\left[X_e=y_e\right]\right]=\left\langle \tilde{T}\left[X_{\mathcal{V}/e},X_e=y_e\right],\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}/e}|X_e=y_e]\right\rangle [\varnothing]+\lambda_1\,.$$

The claim follows when applying the exponential on both sides and with the observation, that

$$\left\langle \tilde{T}\left[X_{\mathcal{V}/e}, X_e = y_e\right], \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\mathcal{V}/e} | X_e = y_e\right] \right\rangle [\varnothing] = \frac{\left\langle \{\tilde{T}\} \cup \{T^{\tilde{e}} : \tilde{e} \neq e\} \right\rangle [X_e = y_e]}{\left\langle \{T^{\tilde{e}} : \tilde{e} \neq e\} \right\rangle [X_e = y_e]}$$

and reparametrization of λ_1 to

$$\lambda = \exp\left[\lambda_1\right] .$$

■

The mean field method corresponds with minimization of the KL Divergence to the efficiently contractable family, i.e. the I-projection onto the family.

Theorem 2.22 *For any hypergraph $\mathcal{G}$ and energy tensor E we have*

$$\operatorname{argmax}_{\mathbb{P} \in \Gamma^{\mathcal{G},\mathbb{I}}} \langle E, \mathbb{P} \rangle [\varnothing] + \mathbb{H}[\mathbb{P}] = \operatorname{argmax}_{\mathbb{P} \in \Gamma^{\mathcal{G},\mathbb{I}}} \mathrm{D}_{\mathrm{KL}} \left[\mathbb{P}^{(\mathcal{G},\theta)} || \left\langle \exp[E] \right\rangle \left[X_{[d]} | \varnothing \right] \right]$$

Problem $\mathbb{P}_{\Gamma^{\mathcal{G},\mathbb{I}},\mathbb{P}}$ is thus the I-projection onto the exponential family $\Gamma^{\mathcal{G},\mathbb{I}}$.

Proof. By rearranging the objective to the KL divergence. ■

2.5 Backward Mapping in Exponential Families

The parameters optimizing the likelihood, will be shown to coincide with the backward mapping evaluated on the expectation of the sufficient statistics (see The. **??**). This is in most generality true for the parameters of the M-projection of any distribution onto the exponential family. We therefore investigate methods to compute the backward mapping, in most generality by alternating algorithms and in the special case of Markov Logic Networks by closed form representations.

We have that θ is a solution of the backward problem at μ^*, if and only if

$$\left\langle \mathbb{P}^{(\phi,\theta,\nu)}, \gamma^{\phi} \right\rangle [L] = \mu^*[L] .$$

This contraction equation is called moment matching, since the moment of the empirical distribution is matched by the moment of the fitting distribution.

We find one backward mapping as the dual problem to the forward mapping.

2.5.1 Variational Formulation

The backward mapping to $\mu_D[L] = \left\langle \mathbb{P}^D, \gamma^{\phi} \right\rangle [L]$ is Maximum Likelihood estimation and the solution of the maximum entropy problem.

Theorem 2.23 *Let there be a sufficient statistic ϕ. The map $B^{(\phi,\nu)} : \mathbb{R}^p \to \mathbb{R}^p$ defined as*

$$B^{(\phi,\nu)}(\mu) = \operatorname{argmax}_{\theta \in \mathbb{R}^p} \langle \mu, \theta \rangle [\varnothing] - A^{(\phi,\nu)}(\theta) .$$

is a backward mapping.

Proof. We show the claim can be shown by the first order condition on the objective. It holds that

$$\frac{\partial}{\partial \theta[L]} A^{(\phi,\nu)}(\theta) = \frac{\partial}{\partial \theta[L]} \ln \left[\left\langle \exp\left[\left\langle \gamma^{\phi}, \theta \right\rangle \left[X_{[d]} \right] \right] \right\rangle [\varnothing] \right]$$

$$= \frac{\partial}{\partial \theta\,[L]} \frac{\left\langle \gamma^{\phi}[L], \exp\left[\left\langle \gamma^{\phi}, \theta \right\rangle \left[X_{[d]}\right]\right]\right\rangle [\varnothing]}{\left\langle \exp\left[\left\langle \gamma^{\phi}, \theta \right\rangle \left[X_{[d]}\right]\right]\right\rangle [\varnothing]}$$
$$= F^{(\phi,\nu)}(\theta)[L]$$

and thus

$$\frac{\partial}{\partial \theta\,[L]} \left(\langle \mu, \theta \rangle [\varnothing] - A^{(\phi,\nu)}(\theta) \right) = \mu\,[L] - F^{(\phi,\nu)}(\theta)[L]\,.$$

The first order condition is therefore

$$\mu\,[L] = F^{(\phi,\nu)}(\theta)[L]$$

and any θ satisfies this condition exactly when $\theta = B^{(\phi,\nu)}(\mu)$ for a backward map. ∎

2.5.2 Interpretation by Maximum Likelihood Estimation

Backward mapping coincides with the Maximum Likelihood Estimation Problem ($\mathrm{P}^{\mathcal{L}_D}_{\Gamma,\mathbb{P}^D}$), when we take Γ to the distributions in an exponential family $\Gamma^{\phi,\nu}$ for a sufficient statistic ϕ.

The loss is the cross entropy between a distribution with μ and the distribution $\mathbb{P}^{(\phi,\theta,\nu)}$.

Theorem 2.24 *Let there be any exponential family, a mean parameter vector $\mu^* \in \operatorname{im}\left(F^{(\phi,\nu)}\right)$ and a backward map $B^{(\phi,\nu)}$. Then $\hat{\theta} = B^{(\phi,\nu)}(\mu^*)$ is the parameter of the M-projection (Problem $\mathrm{P}_{\Gamma,\mathbb{P}^*}$) of any $\mathbb{P}^*$ with $\left\langle \gamma^{\phi}, \mathbb{P}^* \right\rangle [L] = \mu^*[L]$ on to $\Gamma^{\phi,\nu}$, that is*

$$\mathbb{P}^{(\phi,\hat{\theta},\nu)} \in \operatorname{argmax}_{\mathbb{P}\in\Gamma^{\phi,\nu}} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}\right]\,.$$

In particular, if $\mu = \mu_D$ for a data map D, the backward map is a maximum likelihood estimator.

Proof. We exploit the variational characterization of the backward map by The. 2.23, and first show that the objective coincides with the cross entropy between the distribution $\mathbb{P}^*$ and the respective member of the exponential family. For any $\mathbb{P}^*$ and θ we have with Example 2.11

$$\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right] = \left\langle \mathbb{P}^*, \gamma^{\phi}, \theta \right\rangle [\varnothing] - A^{(\phi,\nu)}(\theta)\,.$$

We use that by assumption $\left\langle \mathbb{P}^*, \gamma^{\phi} \right\rangle [L] = \mu^*[L]$ and thus

$$\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right] = \langle \mu^*, \theta \rangle [\varnothing] - A^{(\phi,\nu)}(\theta)\,.$$

This shows, that the backward map coincides with the M-projection onto $\Gamma = \Gamma^{\phi,\nu}$.

Further, if $\mu = \mu_D$ for a data map D, we have that the corresponding empirical distribution $\mathbb{P}^D$ satisfies $\left\langle \gamma^{\phi}, \mathbb{P}^D \right\rangle [L] = \mu\,[L]$. The backward map of μ is therefore the M-projection of $\mathbb{P}^D$, which is with The. 2.10 the maximum likelihood estimator. ∎

2.5.3 Connection with Maximum Entropy

The Maximum entropy problem with respect to matching expected statistics $\mu^* \in \mathcal{M}_{\phi,\nu}$

$$\operatorname{argmax}_{\mathbb{P}\in\Gamma^{\nu}} \mathbb{H}\left[\mathbb{P}\right] \quad \text{subject to} \quad \left\langle \mathbb{P}, \gamma^{\phi} \right\rangle [L] = \mu^*[L]$$

where the optimization is over all the distributions Γ^ν, which are representable with respect to the base measure ν.

Theorem 2.25 *Let ϕ be a statistic and ν a base measure. For any $\mu^* \in \left(\mathcal{M}_{\phi,\nu}\right)^\circ$ the solution of Problem $\mathrm{P}^{\mathbb{H}}_{\phi,\nu,\mu^*}$ is the distribution $\mathbb{P}^{(\tilde{\phi},\hat{\theta},\tilde{\nu})}$, where $\hat{\theta} = B^{\tilde{\phi},\tilde{\nu}}(\tilde{\mu})$.*

Proof. Since $\mu^* \in \left(\mathcal{M}_{\phi,\nu}\right)^\circ$, The. 2.14 implies the existence of $\hat{\theta}$ such that

$$\mu^*[L] = \left\langle \mathbb{P}^{(\phi,\hat{\theta},\nu)}, \gamma^\phi \right\rangle [L] \ .$$

We now follow the argumentation of the proof of Theorem 20.2 in [KF09]. Let $\tilde{\mathbb{P}}$ further be an arbitrary distribution, possibly different from $\mathbb{P}^{(\phi,\hat{\theta},\nu)}$, such that

$$\mu^*[L] = \left\langle \tilde{\mathbb{P}}, \gamma^\phi \right\rangle [L] \ .$$

We then have

$$\mathbb{H}\left[\mathbb{P}^{(\phi,\hat{\theta},\nu)}\right] = \mathbb{H}\left[\tilde{\mathbb{P}}, \mathbb{P}^{(\phi,\hat{\theta},\nu)}\right]$$

With the Gibbs inequality we have if $\tilde{\mathbb{P}} \neq \mathbb{P}^{(\phi,\hat{\theta},\nu)}$

$$\mathbb{H}\left[\mathbb{P}^{(\phi,\hat{\theta},\nu)}\right] - \mathbb{H}\left[\tilde{\mathbb{P}}\right] = \mathbb{H}\left[\tilde{\mathbb{P}}, \mathbb{P}^{(\phi,\hat{\theta},\nu)}\right] - \mathbb{H}\left[\tilde{\mathbb{P}}\right] > 0 \, .$$

Therefore, if $\tilde{\mathbb{P}}$ does not coincide with$\mathbb{P}^{(\phi,\hat{\theta},\nu)}$, it is not a solution of Problem $\mathrm{P}^{\mathbb{H}}_{\phi,\nu,\mu^*}$. ∎

Let us highlight the fact, that in Problem $\mathrm{P}^{\mathbb{H}}_{\phi,\nu,\mu^*}$ we did not restrict to distributions in an exponential family and only demanded representability with respect to the base measure. When choosing the trivial base measure, this does not pose a restriction on the distributions. The. 2.25 states, that when the maximum entropy problem has a solution (i.e. $\mu^* \in \mathcal{M}_{\phi,\nu}$), then the solution is in the exponential family to the statistic ϕ.

When $\mu^* \notin \left(\mathcal{M}_{\phi,\nu}\right)^\circ$, the mean paramater is by The. 2.14 not reproducable by a member of the exponential family $\Gamma^{\phi,\nu}$. Instead, in combination with the base measure refinement Algorithm 3, we show that the solution is in a refined exponential family.

Theorem 2.26 *Let ϕ be a statistic and ν a base measure. For any $\mu^* \in \mathcal{M}_{\phi,\nu}$, let $\tilde{\phi}, \tilde{\nu}$ and $\tilde{\mu}$ be the outputs of Algorithm 3 when passing ϕ, ν and μ^* as input. Then, the distribution $\mathbb{P}^{(\tilde{\phi},\hat{\theta},\tilde{\nu})}$, where $\hat{\theta} = B^{\tilde{\phi},\tilde{\nu}}(\tilde{\mu})$, solves Problem $\mathrm{P}^{\mathbb{H}}_{\phi,\nu,\mu^*}$.*

Proof. The. 2.17 and the above Lemma. ∎

The. 2.26 further implies, that the base measure $\tilde{\nu}$ identified by Algorithm 3 is minimal for the maximum entropy problem, in the sense that the solving distribution is positive with respect to it and all feasible distributions have to be representable by it. This highlights the fact, that the maximum entropy distribution does not vanish beyond those states, which are necessary by The. 2.17.

2.5.4 Alternating Algorithms to Approximate the Backward Map

While the forward map always has a representation in closed form by contraction of the probability tensor, the backward map in general fails to have a closed form representation. Computation of the Backward map can instead be performed by alternating algorithms, as we show here.

Alternate through the coordinates of the statistics and adjust $\theta\,[L = l]$ to a minimum of the likelihood, i.e. where for any $l \in [p]$

$$0 = \frac{\partial}{\partial \theta\,[L = l]} \mathcal{L}_D \left(\mathbb{P}^{(\phi,\theta,\nu)} \right) .$$

This condition is equal to the collection of moment matching equations

$$\left\langle \mathbb{P}^{(\phi,\theta,\nu)}, \gamma^{\phi} \right\rangle [L = l] = \left\langle \mathbb{P}^{D}, \gamma^{\phi} \right\rangle [\varnothing]\, L = l\,.$$

Lemma 2.27 *For any sufficient statistic ϕ a parameter vector θ and a $l \in [p]$ we define*

$$T\left[X_{\phi_l}\right] = \left\langle \{\rho^{\phi}\} \cup \{W^{\tilde{l}} : \tilde{l} \in [p], \tilde{l} \neq l\} \right\rangle \left[X_{\phi_l}\right] .$$

Then the moment matching condition for ϕ_l relative to θ and μ is satisfied for any $\theta\,[L = l]$ with

$$\left\langle W^l, \mathrm{Id}|_{\mathrm{im}(\phi_l)}, T\left[L_{\phi}\right] \right\rangle [\varnothing] = \left\langle W^l, T\left[L_{\phi}\right] \right\rangle [\varnothing] \cdot \mu\,[L = l] .$$

Proof. We have

$$\mathbb{P}^{(\phi,\theta,\nu)} = \frac{\left\langle W^l, T \right\rangle \left[X_{[d]}\right]}{\left\langle W^l, T \right\rangle [\varnothing]}$$

and

$$\left\langle \mathbb{P}^{(\phi,\theta,\nu)}, \phi_l \right\rangle [\varnothing] = \frac{\left\langle W^l, \mathrm{Id}|_{\mathrm{im}(\phi_l)}, T \right\rangle \left[X_{[d]}\right]}{\left\langle W^l, T \right\rangle [\varnothing]} .$$

Here we used

$$\phi_l = \left\langle W^l, \mathrm{Id}|_{\mathrm{im}(\phi_l)} \right\rangle \left[X_{[d]}\right]$$

and redundancies of copies of relational encodings. It follows that

$$\left\langle \mathbb{P}^{(\phi,\theta,\nu)}, \phi_l \right\rangle [\varnothing] = \left\langle \mathbb{P}^{D}, \phi_l \right\rangle [\varnothing]$$

is equal to

$$\left\langle W^l, \mathrm{Id}|_{\mathrm{im}(\phi_l)}, T\left[X_{\phi_l}\right] \right\rangle [\varnothing] = \left\langle W^l, T\left[X_{\phi_l}\right] \right\rangle [\varnothing] \cdot \mu\,[L = l] .$$

■

The steps have to be alternated until sufficient convergence, since matching the moment to l by modifying $\theta\,[L = l]$ will in general change other moments, which will have to be refit.

An alternating optimization is the coordinate descent of the negative likelihood, seen as a function of the coordinates of θ, see Algorithm 5. Since the log likelihood is concave, the algorithm converges to a global minimum.

In general, if $\mathrm{im}\,(\phi_l)$ contains more than two elements, there exists no closed form solutions. We will investigate the case of binary images, where there are closed form expressions, later in Sect. 7.3.

The computation of T^l in Algorithm 5 can be intractable and be replaced by an approximative procedure based on message passing schemes.

Algorithm 5 Alternating Moment Matching

Set $\theta[L] = 0$
Compute $\mu_D[L] = \langle \mathbb{P}^D, \gamma^\phi \rangle [L]$
while Stopping criterion is not met **do**
  for $l \in [p]$ **do**
    Compute

$$T^l[X_{\phi_l}] \leftarrow \left\langle \{\rho^\phi\} \cup \{W^{\tilde{l}} : \tilde{l} \in [p], \tilde{l} \neq l\} \right\rangle [X_{\phi_l}]$$

    Set $\theta[L = l]$ to a solution of

$$\left\langle W^l, \mathrm{Id}|_{\mathrm{im}(\phi_l)}, T^l \right\rangle [\varnothing] \leftarrow \left\langle W^l, T^l \right\rangle [\varnothing] \cdot \mu_D[L = l] \ .$$

  end for
end while

2.6 Discussion

Further in [WJ08]: Convex Duality. Forward mapping coincides with gradient, i.e. $\mu = \nabla A^{(\phi,\nu)}(\theta)$.

In [WJ08]: The objective is the conjugate dual $\left(A^{(\phi,\nu)}\right)^*$ of $A^{(\phi,\nu)}$, and backward mapping has an expression by the gradient, i.e. $\theta = \nabla \left(A^{(\phi,\nu)}\right)^*(\mu)$.

3

Probability Distributions

Propositional logics describes systems with d boolean variables, which are called atoms and denoted by X_k for $k \in [d]$. Indices $x_k \in [2]$ to the atoms $k \in [d]$ enumerate the 2^d states of these systems, which are called worlds. In each world indexed by $x_{[d]} = x_0, \ldots, x_{d-1}$ the indices X_k encode whether the corresponding variable is True.

The epistemological commitments of propositional logics are whether the state is True or False reflected by the coordinate of the one-hot encoding being 1 or 0. Intuitively this describes, whether a specific world can be the state of a factored system. Propositional logic amounts to reason about boolean variables, which are categorical variables with 2 possible values.

Boolean tensors have been already used as boolean base measures in the previous chapters.

Before discussing the semantics and syntax of propositional formulas, we first investigate how Boolean can be represented by vectors in order to mechanize their processing based on contractions.

3.1 Encoding of Booleans

Booleans are variables valued by $\{\text{False}, \text{True}\}$ and consist a basic data structure.

3.1.1 Representation by coordinates

To represent Booleans by categorical variables X with two states we use the index interpretation function

$$I : \{\text{False}, \text{True}\} \to \{0, 1\}$$

defined as

$$I(\text{True}) = 1 \quad \text{and} \quad I(\text{False}) = 0\,.$$

In Def. 11.1 in Part III will define encodings of arbitrary sets based on index interpretation maps.

One motivation for this particular choice of the interpretation function I is the effective execution of the conjunction as we show in the next Lemma.

Lemma 3.1 *I is a homomorphism between the groups*

$$\big(\{\text{False}, \text{True}\}, \wedge\big) \quad \textit{and} \quad \big(\{0, 1\}, \cdot\big)\,.$$

Proof. It suffices to notice, that for arbitrary $z_0, z_1 \in \{\text{False}, \text{True}\}$ we have

$$I(z_0 \wedge z_1) = I(z_0) \cdot I(z_1)\,.$$

■

Based on this homomorphism, contractions of boolean tensors, in which all variables are kept open, can be regarded as parallel calculations of the conjunction $\wedge$ encoded by I. This homomorphism is further applied in type conversion in dynamically-typed languages (e.g. in python [Fou25]).

Operations like the negation fail to be linear and are only affine linear, since for $z \in \{\text{False}, \text{True}\}$ we have

$$I(\neg z) = 1 - I(z)\,. \tag{3.1}$$

Since any logical connective can be represented as a composition of conjunctions and negations, any logical connective corresponds with an affine linear function on the interpreted truth values. Direct applications of this insight to execute logical calculus will be discussed later in Sect. 11.5. For our purposes here, we would like to execute logical connective based on single contractions and avoid summations over them. This is why we call the negation representation as in (3.1) the affine representation problem, which we in the following want to resolve.

While in this work, we will always encode boolean states by I, other index interpretation functions could be chosen. For example, the interpretation

$$I_\vee : \{\text{False}, \text{True}\} \to \{0, 1\}$$

defined as

$$I_\vee(\text{True}) = 0 \quad \text{and} \quad I_\vee(\text{False}) = 1\,,$$

results is a homomorphism between the groups

$$\left(\{\text{False}, \text{True}\}, \vee\right) \quad \text{and} \quad \left(\{0, 1\}, \cdot\right).$$

While placing the disjunction $\vee$ as the logical connective effectively executed by contractions, the negation will for arbitrary interpretations mapping onto $\{0, 1\}$ remain the function

$$I_\vee(\neg z) = 1 - I_\vee(z)\,.$$

Thus, the problem of affine linear operations cannot be resolved by a clever choice of an interpretation function with image in $\{0, 1\}$.

3.1.2 Representation by basis vectors

While contractions can just perform conjunctions, we need a representation trick to extend the contraction expressivity to arbitrary connectives and resolve the affine representation problem. To this end we now compose I with the one-hot encoding e and get an encoding

$$e \circ I : \{\text{False}, \text{True}\} \to \{e_0\,[X]\,, e_1\,[X]\}\,,$$

where X is a categorical variable with $m = 2$. For any $z \in \{\text{False}, \text{True}\}$ we have

$$e \circ I(z) = \begin{bmatrix} I(\neg z) \\ I(z) \end{bmatrix}.$$

Performing the negation now amounts to switching the coordinates of the encoded vector, which can be performed by contraction with a transposition matrix

$$\rho^\neg\,[Y_\neg, X] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

where in this notation we always understand the first variable X as the row index selector and the second variable $Y_\neg$ as the column index selector. We then have

$$e \circ I(\neg z)[Y_\neg] = \langle \rho^\neg\,[Y_\neg, X]\,, e \circ I(z)[X] \rangle\,[Y_\neg]\,.$$

We therefore arrived at our aim to resolve the affine representation problem and have found a procedure to represent logical negations by a contraction, which is a linear operation. Besides

negations, we will show in this chapter, that arbitrary logical formulas can be represented by contractions.

3.1.3 Coordinate and Basis Calculus

Our findings on the encoding of booleans hint towards more general schemes to encode information into boolean tensors, which will be explored in more detail in Chapter 10 and Chapter 11. When each coordinate in a boolean tensor represents one in $\{0,1\}$ interpreted boolean we call the scheme coordinate calculus. In basis calculus on the other hand, booleans are represented by elements of $\{e_0[X], e_1[X]\}$. In that scheme, there are pairs of two coordinates (building slice vectors of the tensors), which are restricted to be different from each other. This amounts to posing a global directionality constraint on the boolean tensor, as will be shown in The. 11.8.

3.2 Semantics of Propositional Formulas

We now choose a semantic centric approach to propositional logic, by defining formulas as boolean tensors. Then we investigate the corresponding syntax of formulas as specification of a tensor network decomposition of the relational encoding of formulas.

3.2.1 Formulas

Logics is especially useful in interpreting boolean tensors representing Propositional Knowledge Bases, based on connections with abstract human thinking. To make this more precise, we associate each such tensor is associated with a formula f being a composition of the atomic variables with logical connectives as we proof next.

Definition 3.2 A propositional formula $f\left[X_{[d]}\right]$ depending on d atoms X_k is a tensor

$$f\left[X_{[d]}\right] : \bigtimes_{k\in[d]} [2] \to [2] \subset \mathbb{R}\,.$$

We call $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ a model of a propositional formula f, if

$$f[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] = 1$$

. If there is a model to a propositional formula, say the formula is satisfiable.

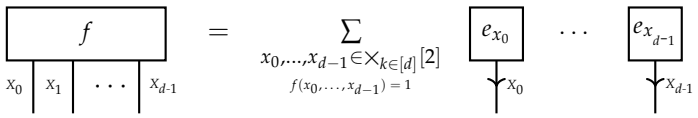
The propositional formulas coincide therefore with the boolean tensors (see Def. 0.16).

Since propositional formulas are binary valued tensors, the generic decomposition of Lem. 10.1 simplifies to

$$\begin{aligned} f\left[X_{[d]}\right] &= \sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} f[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] \cdot e_{x_{[d]}}\left[X_{[d]}\right] \\ &= \sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2] \,:\, f[X_0=x_0,\ldots,X_{d-1}=x_{d-1}]=1} e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]\,. \end{aligned}$$

Thus, any propositional formula is the sum over the one-hot encodings of its models. This is equal to the encoding of the set of models, which will be introduced in Chapter 11 (see Def. 11.1).

We depict this decomposition in the diagrammatic notation by



We here chose a semantic approach to propositional logic in contrary to the standard syntactical approach. Instead of defining formulas by connectives acting on atomic formulas, we define them here as binary valued functions of the states of a factored system. They are interpreted by marking possible states as models, given the knowledge of f. The syntactical side will then be introduced later by studying decompositions of formulas.

3.2.2 Relational encoding of formulas

There are two ways to represent formulas by tensors. One way is to understand $[2]$ as subset of $\mathbb{R}$ and interpreting the formula directly as a tensor (as in Def. 3.2). Another way is to understand $[2]$ as the possible values of a categorical variable. Following this second perspective, formulas are maps between factored systems, where the image system is the factored systems of atoms and the target system the atomic system defined by a variable Y_f representing the formula satisfaction. We can then build the relational encoding (Def. 0.19) of that map to represent the formula (see Figure 3.1).

Given a factored system with d atoms $X_{[d]}$ and a propositional formula f, the relational encoding of f (see Def. 0.19) is the tensor

$$\rho^f\left[Y_f, X_{[d]}\right] \in \left(\bigotimes_{k\in[d]} \mathbb{R}^2\right) \otimes \mathbb{R}^2$$

decomposable as

$$\rho^f\left[Y_f, X_{[d]}\right] = \sum_{x_{[d]}\in\times_{k\in[d]}[2]} e_{x_{[d]}}\left[X_{[d]}\right] \otimes e_{f\left[X_{[d]}=x_{[d]}\right]}\left[Y_f\right] .$$

We can build relational encodings more generally of any tensors, where we identify the image of the tensor with the states of a categorical variable. Exactly for propositional formulas, this construction will lead to Boolean image variables.

Lemma 3.3 *For any formula f we have*

$$\rho^f\left[Y_f, X_{[d]}\right] = f\left[X_{[d]}\right] \otimes e_1\left[Y_f\right] + \neg f\left[X_{[d]}\right] \otimes e_0\left[Y_f\right] .$$

In particular

$$f\left[X_{[d]}\right] = \left\langle \rho^f\left[Y_f, X_{[d]}\right], e_1\left[Y_f\right]\right\rangle \left[X_{[d]}\right] .$$

Proof. We can decompose relational encodings of formulas into the sum (see Figure 3.1)

$$\begin{aligned}\rho^f\left[Y_f, X_{[d]}\right] =& e_0\left[Y_f\right] \otimes \left(\sum_{x_{[d]}\,:\,f\left[x_{[d]}\right]=0} e_{x_{[d]}}\left[X_{[d]}\right]\right) \\ &+ e_1\left[Y_f\right] \otimes \left(\sum_{x_{[d]}\,:\,f\left[x_{[d]}\right]=1} e_{x_{[d]}}\left[X_{[d]}\right]\right)\end{aligned}$$

where the second term sums up the models of f and the first one the models of $\neg f$. ∎

Compared with the direct interpretation of a formula as a tensor and the decomposition into models in Equation 3.2.1, we notice that the relational encoding also represents encoding of worlds where the formula is not satisfied. This representation is required to represent arbitrary propositional formulas by contracted tensor networks of its components, as will be investigated in the following sections.

The relational coordinate ρ^f has slices

$$\left\langle \rho^f, e_{x_{[d]}} \right\rangle \left[Y_f \right] \rho^f \left[X_{[d]} = x_{[d]}, Y_f \right] = \begin{cases} e_1[Y_f] & \text{if the world } x_{[d]} \text{ is a model of } f \\ e_0[Y_f] & \text{else}\,. \end{cases}$$

The contractions of the relational encoding therefore calculate whether an assignment of atoms is a model of the formula, using basis calculus (see The. 11.9).

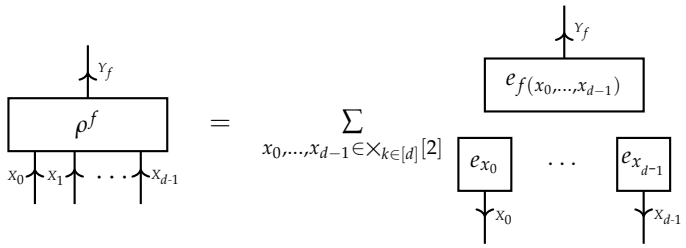


Figure 3.1: Relational encoding of a propositional formula. The encoding is a sum of the one hot encodings of all states of the factored system in a tensor product with basis vectors, which encode whether the state is a model of the formula. The tensor is directed, since any contraction with an encoded state results in the basis vector evaluating the formula, which we called basis calculus.

3.3 Syntax of Propositional Formulas

Relational encodings of propositional formulas are especially useful when representing function compositions by the representation of their components (see The. 11.10). In propositional logics, the syntax of defining propositional formulas is oriented on compositions of formulas by connectives. We in this section investigate the decomposition schemes of relational encodings into tensor networks of component encodings for binary tensors following propositional logic syntax.

3.3.1 Atomic Formulas

We call atomic formulas the most granular formulas, which are not splitted into compositions of other formulas. Our syntactic decomposition of propositional formulas will then investigate, how any propositional formula can be represented by these.

Definition 3.4 The tensors $f_k \left[X_{[d]} \right]$ defined for $x_0, \ldots, x_{d-1}$ as

$$f_k \left[X_{[d]} = x_{[d]} \right] = x_k$$

are called atomic formulas.

Atomic formulas and their relational encodings have an especially compelling representation.

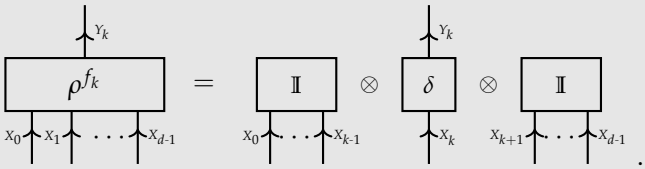
Theorem 3.5 *Any atomic formula* $f_k\left[X_{[d]}\right]$ *is represented as*

$$f_k\left[X_{[d]} = x_{[d]}\right] = \langle e_1\left[X_k\right]\rangle\left[X_{[d]}\right] = e_1\left[X_k\right] \otimes \mathbb{I}\left[X_{[d]/\{k\}}\right].$$

The relational encoding of any atomic formula $X_k\left[X_{[d]}\right]$ *has a tensor decomposition by*

$$\rho^{X_k}\left[Y_d, X_{[d]}\right] = \langle \delta\left[X_k, Y_k\right]\rangle\left[X_{[d]}\right] = \delta\left[X_k, Y_k\right] \otimes \mathbb{I}\left[X_{[d]/\{k\}}\right].$$

The decomposition is depicted in a network diagram as



Proof. We have by definition

$$\begin{aligned}\rho^{X_k}\left[Y_k, X_{[d]}\right] &= \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right] \otimes e_{f_k[X_0=x_0,\ldots,X_{d-1}=x_{d-1}]}\left[Y_k\right] \\ &= \left(e_{0,0}\left[X_k, Y_k\right] + e_{1,1}\left[X_k, Y_k\right]\right) \otimes \mathbb{I}\left[X_l : l \neq k\right] \\ &= \langle \delta\left[X_k, Y_k\right]\rangle\left[X_{[d]}, Y_k\right].\end{aligned}$$

■

3.3.2 Syntactical combination of formulas

Propositional formulas are elements of tensor spaces with d axis. The number of coordinates thus grows exponentially with the number of atoms, which is

$$\dim\left(\bigotimes_{k\in[d]} \mathbb{R}^2\right) = 2^d.$$

When the number of atoms is large, the naive representation of formula tensors will be thus intractable. In contrast, typcial logical formulas appearing in practical knowledge bases are sparse in the sense that they have short representations in a logical syntax. Motivated by this consideration we now discuss propositional syntax and investigate the sparse decomposition of formula tensors along their formula structure to avoid the curse of dimensionality.

In logical syntax formulas are described by atomic formulas recursively connected via connectives. We show, that representations of logical connectives can be represented by feasible tensor cores ρ° contracted along a tensor network. Let us first provide in Example 3.6 unary ($d = 1$) and binary ($d = 2$) connectives.

Example 3.6 We use the following connectives:

- negation $\neg : [2] \to [2]$ by the vector

$$\neg[Y_f] = \begin{bmatrix}0\\1\end{bmatrix}$$

- conjunctions $\wedge : [2] \times [2] \to [2]$

$$\wedge[Y_f, Y_h] = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

- disjunctions $\vee : [2] \times [2] \to [2]$

$$\vee[Y_f, Y_h] = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

- exact disjunction $\oplus : [2] \times [2] \to [2]$

$$\oplus[Y_f, Y_h] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- implications $\Rightarrow$: $[2] \times [2] \to [2]$

$$\Rightarrow [Y_f, Y_h] = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

- biimplication $\Leftrightarrow$: $[2] \times [2] \to [2]$

$$\Leftrightarrow [Y_f, Y_h] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$\Diamond$

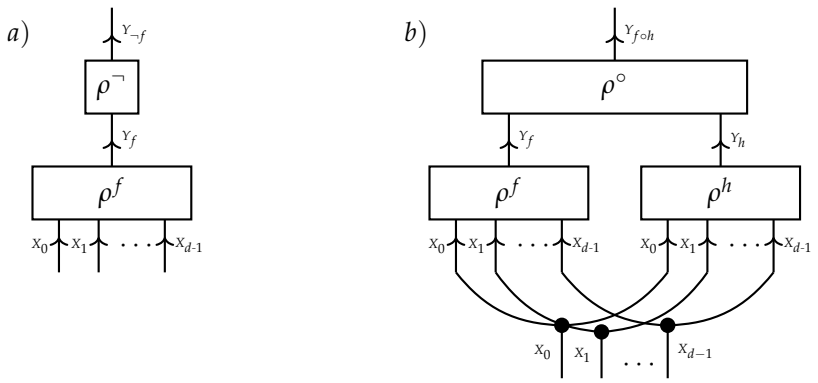


Figure 3.2: a) Relational encoding of a negated formula f as a tensor network of the encoded formula and the encoded connective $\neg$. b) Relational encoding of a composition of formulas f, h by a connective $\circ \in \{\wedge, \vee, \oplus, \Rightarrow, \Leftrightarrow\}$. The encoding is a contraction of encodings to f, h and $\circ$.

Let there be formulas f and h depending on categorical variables $X_{[d]}$ and a binary connective

$$\circ : [2] \times [2] \to [2]\,.$$

Then we can show as a special case of the next theorem, that (see Figure 3.2)

$$\rho^{f\circ h}\left[X_{[d]}, X_{f\circ h}\right] = \left\langle \rho^{\circ}\left[Y_f, Y_h, X_{f\circ h}\right], \rho^{f}\left[X_{[d]}, Y_f\right], \rho^{h}\left[X_{[d]}, Y_h\right]\right\rangle \left[X_{[d]}, X_{f\circ h}\right]\,.$$

For any unary connective $\circ : [2] \to [2]$ we have

$$\rho^{\circ f}\left[X_{[d]}, X_{\circ f}\right] = \left\langle \rho^{\circ}\left[Y_f, X_{\circ f}\right], \rho^{f}\left[X_{[d]}, Y_f\right]\right\rangle \left[X_{[d]}, X_{\circ f}\right] .$$

Let us now generalize this observation to arbitrary arity of connectives and provide a proof of its correctness.

Theorem 3.7 (Composition of Formulas) *Let there be a set of binary variables $X_{\mathcal{V}}$ including atoms $X_{[d]}$ and image variables to some formulas. For any formula $f\,[X_0, \ldots, X_{d-1}]$, which has a syntactical composition into connectives $\{\circ_l\,[X_{\mathcal{V}^l}] : l \in [p]\}$ taking their inputs by variables $X_{\mathcal{V}^l} \subset X_{\mathcal{V}}$ and output by a variable $Y_{\circ_l}$ we have that*

$$\rho^{f}\left[Y_f, X_{[d]}\right] = \left\langle \left\{\rho^{\circ_l}\left[Y_{\circ_l}, X_{\mathcal{V}^l}\right] : l \in [p]\right\}\right\rangle \left[Y_f, X_{[d]}\right] .$$

Proof. When a variable in $X_{\mathcal{V}}$ appears multiple times as input to connectives, we replace it by a set of copies (which wont change the contraction, since all tensors are binary and The. 13.3 can be applied). This follows from an iterative application of The. 11.10 to be shown in Chapter 11. ∎

Remark 3.8 (d-ary connectives such as $\wedge$ and $\vee$) Since the decomposition of relational encoding can be applied to generic function compositions (see The. 11.10), we can also allow for d-ary connectives

$$\circ : \bigtimes_{k\in[d]} [2] \to [2] .$$

The connectives $\wedge$ and $\vee$ satisfy associativity and have thus straightforward generalizations to the d-ary case. This is because associativity can be exploited to represent the relational encoding by any tree-structured composition of binary $\wedge$ and $\vee$ connectives. ♢

Propositional syntax consists in the application of connectives on atomic formulas, and recursively on the results of such constructions. When passed towards connective cores, atomic formula tensors act trivial on the legs and just identify the corresponding atomic formula index x_{X_k} with x_k. This is due to the fact, that contractions with the trivial tensor $\mathbb{I}$ leaves any tensor invariant, and the contraction with the elementary matrix δ identifies indices with each other. We can thus savely ignore the atomic formula tensors appearing in the decomposition of formula tensors to non-atomic formulas. An example of such a decomposition is depicted in Figure 3.3.

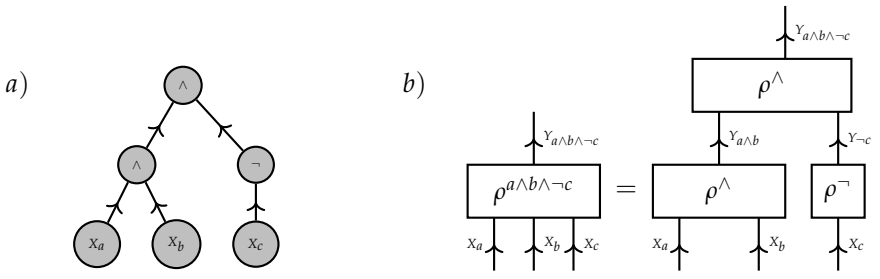


Figure 3.3: Decomposition of the formula tensor to $f = a \wedge b \wedge \neg c$ into unary (matrix) and binary (third order tensor) cores. a) Visualization of f as a graph. b) Tensor Network decomposition of f. We can make use of the invariance of a Hadamard product with a constant tensor $\mathbb{I}$ and thus not draw axis to atoms not affected by a formula.

Remark 3.9 (Tensor Network Decomposition of Formulas) The decomposition of the propositional into a tensor network is a hierarchical decomposition of the formula tensor, which we will describe in more detail in Sect. 12.5. Of special interest are tree hypergraphs, where the format is

called Hierarchical Tucker. At each decomposition of a formula into sub-formulas, two subspaces spanned by the respective atomic spaces are selected. ◊

3.3.3 Syntactical decomposition of formulas

We have seen how the decomposition of complex formulas into connectives acting on the component formulas can be exploited to find effective representations of the semantics by tensor networks. Here the question arises here, how to perform such decompositions in case of a missing syntactical representation of a formula. By Def. 3.2 any binary tensor is a formula. We show in the following, how we can find a syntactic specification of a formula given its tensor.

Definition 3.10 (Terms and Clauses) Given two disjoint subsets $\mathcal{V}^0$ and $\mathcal{V}^1$ of $[d]$, the corresponding term is the formula defined on the indices $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ by

$$Z^{\wedge}_{\mathcal{V}^0,\mathcal{V}^1}\left[X_{[d]}\right] = \left(\bigwedge_{k\in\mathcal{V}^0} \neg f_k\right) \wedge \left(\bigwedge_{k\in\mathcal{V}^1} f_k\right)$$

and the corresponding clause is the formula defined on the indices $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ by

$$Z^{\vee}_{\mathcal{V}^0,\mathcal{V}^1}\left[X_{[d]}\right] = \left(\bigvee_{k\in\mathcal{V}^0} f_k\right) \vee \left(\bigvee_{k\in\mathcal{V}^1} \neg f_k\right),$$

where by $\wedge_{k\in\mathcal{V}}$ and $\vee_{k\in\mathcal{V}}$ we refer to the n-ary connectives $\wedge$ and $\vee$. We call the term a minterm and the clause a maxterm, if $\mathcal{V}^0 \cup \mathcal{V}^1 = [d]$.

Terms and Clauses have for any index tuple $x_{[d]}$ a polynomial representation by

$$Z^{\wedge}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]} = x_{[d]}] = \left(\prod_{k\in\mathcal{V}^0}(1-x_k)\right)\left(\prod_{k\in\mathcal{V}^1} x_k\right)$$

and

$$Z^{\vee}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]} = x_{[d]}] = 1 - \left(\prod_{k\in\mathcal{V}^0}(1-x_k)\right)\left(\prod_{k\in\mathcal{V}^1} x_k\right).$$

Lemma 3.11 *Terms are contractions of one-hot encodings, that is for any disjoint subsets $\mathcal{V}^0, \mathcal{V}^1 \subset [d]$ we have*

$$Z^{\wedge}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]}] = \left\langle e_{\{x_k=0:k\in\mathcal{V}^0\}\cup\{x_k=1:k\in\mathcal{V}^1\}}\right\rangle \left[X_{[d]}\right].$$

Clauses are substractions of one-hot encodings from the trivial tensor, that is for any disjoint subsets $\mathcal{V}^0, \mathcal{V}^1 \subset [d]$ we have

$$Z^{\vee}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - \left\langle e_{\{x_k=0:k\in\mathcal{V}^0\}\cup\{x_k=1:k\in\mathcal{V}^1\}}\right\rangle \left[X_{[d]}\right].$$

The reference of the formulas in the case $\mathcal{V}^0 \dot\cup \mathcal{V}^1 = [d]$ as minterms and maxterms is due to the fact, that minterms are formulas with unique models and maxterms are formulas with a unique world not satisfying the formula. We use this insight and enumerate maxterms and minterms by the index $x \in \bigtimes_{k\in[d]}[2]$ of the unique world where the minterm is satisfied, respectively the maxterm is not satisfied. For any $\mathcal{V}^0 \dot\cup \mathcal{V}^1 = [d]$ we take the index tuple $x_0, \ldots, x_{d-1}$ where $x_k = 0$

if $k \in \mathcal{V}^0$ and $x_k = 1$ if $k \in \mathcal{V}^1$ and define

$$Z^{\vee}_{x_0,\ldots,x_{d-1}} = Z^{\vee}_{\mathcal{V}^0,\mathcal{V}^1} \quad \text{and} \quad Z^{\wedge}_{x_0,\ldots,x_{d-1}} = Z^{\wedge}_{\mathcal{V}^0,\mathcal{V}^1} .$$

Corollary 3.12 *Minterms are basis elements of the tensor space, that is for any* $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ *we have*

$$Z^{\wedge}_{x_{[d]}} = e_{x_{[d]}}\left[X_{[d]}\right]$$

Maxterms are substraction of basis elements from the trivial tensor, that is for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ *we have*

$$Z^{\vee}_{x_{[d]}} = \mathbb{I}\left[X_{[d]}\right] - e_{x_{[d]}}\left[X_{[d]}\right] .$$

Proof. Follows from Lem. 3.11, since when $\mathcal{V}^0 \cup \mathcal{V}^1 = [d]$ the contraction of the one-hot encodings coincides with the one-hot encoding of a fully specified state. ∎

Based on this insight, we can decompose any propositional formula into a conjunction of maxterms or a disjunction of minterms as we show next.

Theorem 3.13 *For any binary tensor* $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^2$ *with two-dimensional axes we have*

$$T\left[X_{[d]}\right] = \left(\bigvee_{x_{[d]} \,:\, T\left[X_{[d]}=x_{[d]}\right]=1} Z^{\wedge}_{\{k:x_k=0\},\{k:x_k=0\}} \right)[X_{[d]}]$$

and

$$T\left[X_{[d]}\right] = \left(\bigwedge_{x_{[d]} \,:\, T\left[X_{[d]}=x_{[d]}\right]=0} Z^{\vee}_{\{k:x_k=0\},\{k:x_k=0\}} \right)[X_{[d]}] .$$

Proof. To show the representation by minterms we use the decomposition

$$T\left[X_{[d]}\right] = \sum_{x_{[d]} \,:\, T\left[X_{[d]}=x_{[d]}\right]=1} e_{x_{[d]}}\left[X_{[d]}\right]$$

and notice that each term in the disjunction modifies the formula by adding respective world $x_{[d]}$ to the models of the formula. To show the representation by maxterms we use the decomposition

$$T\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] \quad - \sum_{x_{[d]} \,:\, T\left[X_{[d]}=x_{[d]}\right]=0} e_{x_{[d]}}\left[X_{[d]}\right]$$

and notice that each term in the conjunction modifies the formula by removing the respective world $x_{[d]}$ from the models of the formula. Thus, both decompositions are propositional formulas with the same set of models as the formula T and are thus identical to T. ∎

The decompositions found in The. 3.13 are also called canonical normal forms to propositional formulas $T\left[X_{[d]}\right]$.

Remark 3.14 (Efficient Representation in Propositional Syntax) The decomposition in The. 3.13 is a basis CP decomposition of the binary tensor and will further be investigated in Chapter 12. The formulas constructed in the proof of The. 3.13 are however just one possibility to represent a formula tensor in propositional syntax. Typically there are much sparser representations for many formula tensors, in the sense that less connectives and atomic symbols are required. Having such a sparser syntactical description of a propositional formula can be exploited to find a shorter conjunctive normal form of the formula and construct a sparse polynomial based on similar ideas as in The. 3.13. We will provide such constructions in Chapter 12, where we show that dropping the demand of directionality and investigating binary CP Decompositions will improve the sparsity of the polynomial formula representation. ◊

3.3.4 Comparing with probabilistic approaches

Both probability and logic provide a human-understandable interface to machine learning. As we will describe in Part II, they can be combined in one formalism providing efficient reasoning.

Probability represents the uncertainty of states. The categorical variables are called random variables and their joint distribution is represented by a probability tensor. Humans interpret probabilities by Bayesian and frequentist approaches. Reasoning based on Bayes Theorem has an intuitive interpretation in terms of evidence based update of prior distributions to posterior distributions. However it is based on interpreting (large amounts) of numbers, which makes it hard for humans to assess the probabilistic reasoning process.

Logics explains relations between sets of worlds in a human understandable way. Categorical variables have dimension 2, where the first is interpreted as indicating a False state and the second as a True state. We mainly restrict to propositional logics, where there are finite sets of such variables called atomic formulas. Using model-theoretic semantics it defines entailment of sets by other sets, which is understandable as a consequence relation.

Tensors unify both approaches since they are natural numerical structures to represent properties of states in factored systems. The potential is then based in employing scalable multilinear algorithms to solve reasoning problems. Further, algorithms formulated in tensor networks have a high parallelization potential, which is why they are of central interest in the development of AI-dedicated software and hardware.

The different areas have developed separated languages to describe similar objects. Here we want to provide a rough comparison of those in a dictionary.

	Probability Theory	**Propositional Logic**	**Tensors**
Atomic System	Random Variable	Atomic Formula	Vector
Factored System	Joint Distribution	Knowledge Base	Tensor
Categorical Variable	Random Variable	Atomic Formula	Axis of the Tensor

While the probability theory lacks to provide an intuition about sets of events, propositional syntax has limited functionality to represent uncertainties. Tensors on the other side can build a bridge by representing both functionalities and relying on probability theory and logics for respective interpretations.

3.4 Discussion and Outlook

Further study of representing Knowledge Bases based on Tensor Networks of its formulas in Sect. 6.2 (see The. 6.11).

4

Logical Inference

We approach logical inference by defining probability distributions based on propositional formulas and then apply the methodology introduced in the more generic situation of probabilistic inference. Logical approaches pay here special attention to situations of certainty, where a state of a variable has probability 1. In this situation, we say that the corresponding formula is entailed.

We start the discussion by showing how formulas can be interpreted by distributions and define logical entailment based on corresponding probabilistic queries. This enables us to define logical entailment based on the resulting conditional distributions.

Remark 4.1 (Interpretation of Contractions in Logical Reasoning) The coordinates of contracted binary tensor networks describe whether the by the coordinate indexed world is a model of the Knowledge Base at hand. Contractions, which only leave a part variables open, store the counts of the world respecting conditions given by the choice of slices. When contracting without open variables, we thus get the total worldcount.

This is consistent with the probabilistic interpretation of contractions, when applying the frequentist interpretation of probability and defining normed worldcounts as probabilities. ♢

4.1 Entailment in Propositional Logics

Definition 4.2 (Entailment of propositional formulas) Given two propositional formulas $\mathcal{KB}$ and f we say that $\mathcal{KB}$ entails f, denoted by $\mathcal{KB} \models f$, if any model of $\mathcal{KB}$ is also a model of f, that is

$$\forall_{x_{[d]} \in \times_{k \in [d]}[2]} \left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

If $\mathcal{KB} \models \neg f$ holds, we say that $\mathcal{KB}$ contradicts f.

Entailment can be understood by subset relations of the models of formulas. This perspective can be applied with subset encodings in Chapter 11.

4.1.1 Deciding Entailment by contractions

Theorem 4.3 (Contraction Criterion of Entailment) *We habe $\mathcal{KB} \models f$ if and only if*

$$\langle \mathcal{KB}, \neg f \rangle [\varnothing] = 0.$$

Proof. If for a $x_0, \ldots, x_{d-1}$ we have $\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1$ but not $\left(f(x_0, \ldots, x_{d-1}) = 1\right)$, the contraction would be at least 1. Conversely, if the contraction is at least one, we would find $x_0, \ldots, x_{d-1}$ with $\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1$ and $\neg f\left[X_{[d]} = x_{[d]}\right] = 1$, therefore $f\left[X_{[d]} = x_{[d]}\right] = 0$. It follows that $\mathcal{KB} \models f$ does not hold. ■

To decide whether a formula is entailed, or its negation is entailed (in which case one says that the formula is contradicted) by a single contraction, one can perform the contraction

$$T = \left\langle \mathcal{KB}\left[X_{[d]}\right], f\left[X_{[d]}, Y_f\right] \right\rangle \left[Y_f\right]$$

and use that

$$\langle \mathcal{KB}, \neg f \rangle [\varnothing] = T \left[Y_f = 0 \right]$$

and

$$\langle \mathcal{KB}, f \rangle [\varnothing] = T \left[Y_f = 1 \right] .$$

4.1.2 Contraction Knowledge Base

We now show how to implement a propositional Knowledge Base with the TELL and ASK operations based on The. 4.9.

Algorithm 6 Contraction Knowledge Base

ASK(formula f)
  $T \left[Y_f \right] \leftarrow \left\langle \mathcal{KB}, \rho^f \right\rangle \left[Y_f \right]$
  if $T \left[Y_f = 0 \right] = 0$ **then**
    return Entailed
  end if
  if $T \left[Y_f = 1 \right] = 0$ **then**
    return Contradicted
  end if
  return Contingent
TELL(formula f)
  if ASK(f) returns Contingent: **then**
    $\mathcal{KB} \leftarrow \mathcal{KB} \wedge f$
  end if

Comment: TELL checks whether the formula to be added is entailed, in which case it is redundant to add, and whether the formula to be added is contradicted, in which case the knowledge base would become unsatisfiable.

4.1.3 Sparse Representation of a Knowledge Base

Let us now investigate how to sparsely represent a Knowledge Base. Towards getting insights on this we first show that entailed formulas can be dropped from the Knowledge Base.

Theorem 4.4 *If and only if $\mathcal{KB} \models f$ we have*

$$\mathcal{KB} \left[X_{[d]} \right] = \langle \mathcal{KB}, f \rangle \left[X_{[d]} \right] .$$

Proof. For any world indexed by a coordinate $x_0, \ldots, x_{d-1}$, $\mathcal{KB} \left[X_{[d]} = x_{[d]} \right]$ indicates whether the world is a model of $\mathcal{KB}$. We have entailment, when the models of $\mathcal{KB} \cup f$ coincide with those of $\mathcal{KB}$. ■

Remark 4.5 (Sparsest Description of a Knowledge Base) Given a set of worlds indexed by T, find the sparsest set of formulas $\mathcal{KB}$ such that

$$T = \mathcal{KB}$$

would be benefitial for small computational complexity. Since the formula tensors are invariant under entailment, we can drop entailed formulas. ♢

4.2 Formulas as Random Variables

Aim here: Relate with the probabilistic reasoning concepts of marginal and conditional distributions.

Given a probability distribution $\mathbb{P}$ of atoms we add a variable by building the Markov Network of $\mathbb{P}$ and ρ^f to get a joint distribution of the atoms and a query formula f

There are two ways of interpreting formula tensors as conditional probabilities. The standard one, which we also used above, understands the atomic legs as conditions and calculates the truth of the formula. Another understands a formula as a condition.

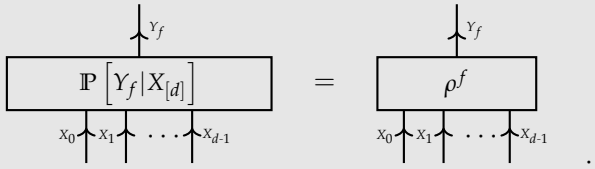
4.2.1 Conditioning on the atoms

Our main interpretation understands each tuple of indices $x_0, \dots, x_{d-1}$ as conditions of a probability tensor. Given a truth assignment to the atomic variables X_k, that is a choice of indices x_k, determines the truth of the formula. We thus interpret the formula tensors as defining a conditional probability of f given the atoms X_k indexed by x_k.

Theorem 4.6 *The relational encoding of any propositional formula f coincides with the conditional probability of that formula conditioned on the identity on the atoms, that is*

$$\rho^f = \mathbb{P}\left[Y_f | X_{[d]}\right] .$$

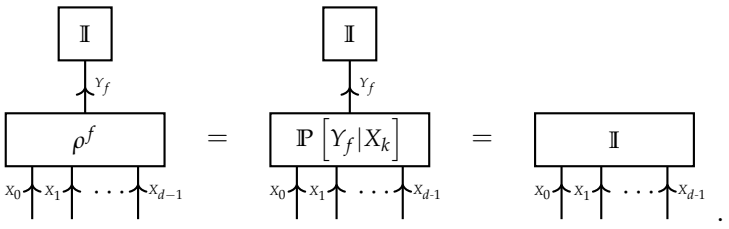
We depict this by



Proof. The distribution $\mathbb{P}$ does not influence the conditional query, since the normation acts on any state. ∎

The conditional query $\mathbb{P}\left[Y_f | X_{[d]}\right]$ provides an interpretation of ρ^f as a conditional probability. This is also reflected in the fact that both $\mathbb{P}\left[Y_f | X_{[d]}\right]$ and ρ^f are directed, since the first is a normation by Defintion 2.1 and the second a encoding of a function.

This directly implies using The. **??** the trivialization of the formula tensor when contracting its head axis indexed by x_f with the trivial vector $\mathbb{I}$, depicted as

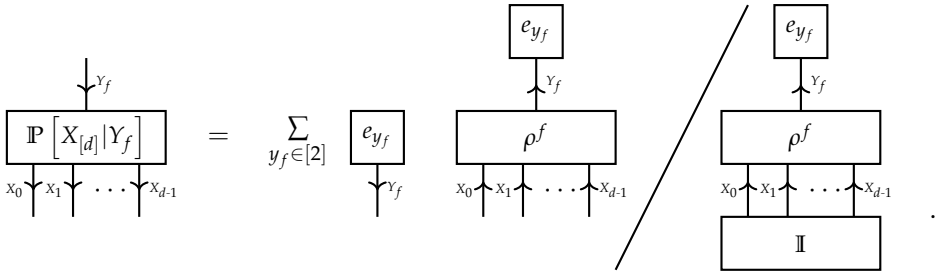


4.2.2 Conditioning on the formula

Let us now converse the order of conditioning from $\mathbb{P}\left[f | X_{[d]}\right]$ to $\mathbb{P}\left[X_{[d]} | f\right]$. In this way, we have propositonal formulas defining probability distributions on the factored system of atoms.

Given a Markov Network $\mathbb{P}$ with a single core ρ^f for a propositional formula f. By definition we have

$$\mathbb{P}\left[X_{[d]}|Y_f\right] = \left\langle \rho^f \right\rangle \left[X_{[d]}|Y_f\right] .$$



Let us further investigate the slices of $\mathbb{P}\left[X_{[d]}|f\right]$ with respect to f, which define distributions of the states of the factored system. To this end, let us condition on the event of $f = 1$, for which we have the distribution

$$\mathbb{P}\left[X_{[d]}|Y_f = 1\right] = \frac{1}{\langle f \rangle [\varnothing]} \sum_{x_{[d]} \in \times_{k\in[d]}[2] : f\left[X_{[d]} = x_{[d]}\right]=1} e_{x_{[d]}} \left[X_{[d]}\right] . \tag{4.1}$$

With $\langle f \rangle [\varnothing]$ being the number of models of f, this is the uniform distribution among the models of f. Conversely, when conditioning on the event $Y_f = 0$ we get a uniform distribution of the models of $\neg f$.

The probability distribution in Equation (4.1) is well defined except for the case that $\langle f \rangle [\varnothing] = 0$. In this case we have $f = 0$ and call f unsatisfiable, since it has no models.

From an epistemological point of view, probability theory is a generalization of logics, since we allow for probability values in the interval $[0, 1]$. The set of distributions being constructed by conditioning on propositional formulas as in Equation (4.1) correspond within the set of probability distributions with those having constant coordinates on their support. While the probability tensors with nonvanishing coordinates build a $2^d - 1$-dimensional manifold, where the formulas parametrize 2^{2^d} probability tensors, most of which having vanishing coordinates.

4.2.3 Probability of a function given a Knowledge Base

We can now combine the ideas of the previous two subsections and define probabilities of formulas f given the satisfaction of another formula $\mathcal{KB}$, which we call a Knowledge Base. We have by The. 4.6

$$\begin{aligned}\mathbb{P}\left[Y_f|Y_{\mathcal{KB}}\right] &= \left\langle \mathbb{P}\left[Y_f|X_{[d]}\right], \mathbb{P}\left[X_{[d]}|Y_{\mathcal{KB}}\right] \right\rangle \left[Y_f, Y_{\mathcal{KB}}\right] \\ &= \left\langle \rho^f, \rho^{\mathcal{KB}} \right\rangle \left[Y_f|Y_{\mathcal{KB}}\right]\end{aligned}$$

Of special interest is the marginal probability of Y_f given that $Y_{\mathcal{KB}}$ is satisfied, that is

$$\begin{aligned}\mathbb{P}\left[Y_f|Y_{\mathcal{KB}} = 1\right] &= \left\langle \{\rho^f, \mathcal{KB}\} \right\rangle \left[Y_f|\varnothing\right] \\ &= \frac{\left\langle \{\rho^f, \mathcal{KB}\} \right\rangle \left[Y_f\right]}{\langle \{\mathcal{KB}\} \rangle [\varnothing]} .\end{aligned}$$

Remark 4.7 (Case of Unsatisfiable Knowledge Bases) When the Knowledge Base is not satisfiable, one cannot normate it and the probability distribution is not dedfined. ♢

Theorem 4.8 *Given a satisfiable formula $\mathcal{KB}$, we have $\mathcal{KB} \models f$, if and only if*

$$\mathbb{P}\left[Y_f = 0 | Y_{\mathcal{KB}} = 1\right] = 0\,.$$

Proof. Since $\mathcal{KB}$ is satisfiable, we have $\langle \mathcal{KB} \rangle [\varnothing] > 0$ and

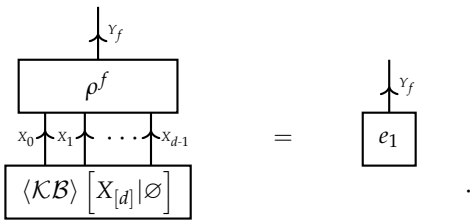
$$\mathbb{P}\left[Y_f = 0 | Y_{\mathcal{KB}} = 1\right] = \frac{\langle \neg f, \mathcal{KB} \rangle [\varnothing]}{\langle \mathcal{KB} \rangle [\varnothing]}\,.$$

This term vanishes if and only if $\langle \neg f, \mathcal{KB} \rangle [\varnothing]$ vanish. Thus, the condition is equivalent to the condition in The. 4.3. ∎

Given that $\mathcal{KB}$ is satisfiable, we therefore have $\mathcal{KB} \models f$ if and only if

$$\mathbb{P}\left[Y_f | Y_{\mathcal{KB}} = 1\right] = e_1\,.$$

We depict this condition by the contraction diagram



We can omit the normation by $\langle \mathcal{KB} \rangle [\varnothing]$ when deciding entailment, as we state next.

Corollary 4.9 *Given a satisfiable formula $\mathcal{KB}$, we have $\mathcal{KB} \models f$ (respectively $\mathcal{KB} \models \neg f$), if and only if*

$$\left\langle \mathcal{KB}, \rho^f \right\rangle \left[Y_f = 0\right] = 0 \quad (\textit{ respectively } \left\langle \mathcal{KB}, \rho^f \right\rangle \left[Y_f = 1\right] = 0\,.$$

Relating entailment to probability distributions motivates an extension of Definition4.2 of entailment to arbitrary probability distributions.

Definition 4.10 For any propositional formula f and a probability distribution $\mathbb{P}$ we say that $\mathbb{P}$ probabilistically entails f, denoted as $\mathbb{P} \models f$, if

$$\left\langle \mathbb{P}, \rho^f \right\rangle \left[Y_f = 0\right] = 0.$$

If $\mathbb{P} \models \neg f$ we say that $\mathbb{P}$ probabilistically contradicts f.

By The. 4.8 the definition of entailment reduces to propositional formulas by choosing $\mathbb{P} = \langle \mathcal{KB} \rangle \left[X_{[d]} | \varnothing\right]$

4.2.4 Knowledge Bases as Base Measures for Probability Distributions

Let us now relate the probabilistic entailment definition 4.10 with the logical entailment. Given a generic probability distribution $\mathbb{P}$ we can build a Knowledge Base by the indicator function of the support as

$$\mathcal{KB}^{\mathbb{P}} = \mathbb{I}_{\neq 0} \circ \mathbb{P}$$

where $\mathbb{I}_{\neq 0} : \mathbb{R} \to \mathbb{R}$ is defined as $\mathbb{I}_{\neq 0}(x) = 1$ if $x \neq 0$ and $\mathbb{I}_{\neq 0}(x) = 0$ else.

Theorem 4.11 *Any probability distribution $\mathbb{P}$ probabilistically entails a formula f, if and only if the Knowledge Base $\mathcal{KB}^{\mathbb{P}}$ logically entails f.*

Proof. Whenever $\mathbb{P}$ does not entail f probabilistically we find a state $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ such that

$$\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0 \quad \text{and} \quad f\left[X_{[d]} = x_{[d]}\right] = 0\,.$$

We further have $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0$ if and only if $\mathcal{KB}^{\mathbb{P}}[X_{[d]} = x_{[d]}] = 1$ and

$$\left((\mathcal{KB}^{\mathbb{P}}[X_{[d]} = x_{[d]}] = 1) \to (f\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

is not satisfied. Together, $\mathbb{P} \models f$ does not holds if and only if

$$\forall X_{[d]}(\mathcal{KB}^{\mathbb{P}}[X_{[d]} = x_{[d]}] = 1) \to (f\left[X_{[d]} = x_{[d]}\right] = 1)$$

is not satisfied. Therefore, probabilistic entailment of f by $\mathbb{P}$ is equivalent to logical entailment of f by $\mathcal{KB}^{\mathbb{P}}$. ∎

Let us use this to connect the entailment formalism with the representability (see Def. 1.2) and positivity (see Def. 1.3) of distributions with respect to boolean base measures.

Theorem 4.12 *A distribution $\mathbb{P}$ of boolean variables is representable with respect to ν, if and only if $\mathbb{I}_{\neq 0} \circ \mathbb{P} \models \nu$. A distribution $\mathbb{P}$ of boolean variables is positive with respect to ν, if and only if $\nu = \mathbb{I}_{\neq 0} \circ \mathbb{P}$.*

Proof. To show the first claim, let $\mathbb{P}$ be a distribution and ν be a base measure. With Def. 1.2, $\mathbb{P}$ is representable with respect to ν, if and only if

$$\forall_{x_{[d]}\in\times_{k\in[d]}[2]}\left(\nu\left[X_{[d]} = x_{[d]}\right] = 0\right) \to \left(\mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0\right).$$

This is equal to

$$\forall_{x_{[d]}\in\times_{k\in[d]}[2]}\left(\mathbb{I}_{\neq 0} \circ \mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(\nu\left[X_{[d]} = x_{[d]}\right] = 1\right)$$

and by definition Def. 4.2 equal to $\nu \models \mathbb{I}_{\neq 0} \circ \mathbb{P}$.

To show the second claim, we show that when $\mathbb{P}$ is in addition positive with respect to ν, then also $\nu \models \mathbb{I}_{\neq 0} \circ \mathbb{P}$ and thus $\nu = \mathbb{I}_{\neq 0} \circ \mathbb{P}$. Let $\mathbb{P}$ be a distribution, which is representable with respect to ν. Then $\mathbb{P}$ is positive with respect to ν, if and only if

$$\forall_{x_{[d]}\in\times_{k\in[d]}[2]}\left(\nu\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0\right)$$

This is equal to

$$\forall_{x_{[d]}\in\times_{k\in[d]}[2]}\left(\nu\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(\mathbb{I}_{\neq 0} \circ \mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 1\right)$$

and thus $\nu \models \mathbb{I}_{\neq 0} \circ \mathbb{P}$. ∎

4.2.5 Deciding entailment on Markov Networks

Theorem 4.13 *Let $\mathcal{T}^{\mathcal{G}} = \{T^e : e \in \mathcal{E}\}$ be a non-negative Tensor Network on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\tilde{\mathcal{V}} \subset \mathcal{V}$ be a subset and*

$$\mathbb{P}[X_{\tilde{\mathcal{V}}}] = \langle \{T[e] : e \in \mathcal{E}\} \rangle [X_{\tilde{\mathcal{V}}} | \varnothing]$$

and

$$\tilde{\mathbb{P}}[X_{\tilde{\mathcal{V}}}] = \left\langle \{\mathbb{I}_{\neq 0} \circ T[e] : e \in \mathcal{E}\} \right\rangle [X_{\tilde{\mathcal{V}}} | \varnothing]$$

Then we have for any f that $\mathbb{P} \models f$ if and only if $\tilde{\mathbb{P}} \models f$.

Proof. We first show

$$\mathbb{I}_{\neq 0} \circ \mathbb{P} = \mathbb{I}_{\neq 0} \circ \tilde{\mathbb{P}} \,. \tag{4.2}$$

The claim follows then from The. 4.11. To show (4.2) let there be $X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}$ such that $\mathbb{P}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$. Then for any $X_{\mathcal{V}} = x_{\mathcal{V}}$ extending $X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}$ we have $\langle \{T[e] : e \in \mathcal{E}\} \rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0$ and thus also $\left\langle \{\mathbb{I}_{\neq 0} \circ T[e] : e \in \mathcal{E}\} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0$ and $\tilde{\mathbb{P}}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$. One can similarly show, that when $\tilde{\mathbb{P}}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$ then also $\mathbb{P}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$. The support of the distributions $\mathbb{P}$ and $\tilde{\mathbb{P}}$ is thus identical and (4.2) holds. ∎

For any positive tensor T we have

$$\mathbb{I}_{\neq 0} \circ T[X_e] = \mathbb{I}[X_e] \,,$$

which does not influence the distribution and can be omitted from the Markov Network. By The. 4.13, when deciding eintailment, we can reduce all tensors of a Markov Network to their support and omit those with full support. Since the support indicating tensors $\mathbb{I}_{\neq 0} \circ T[X_e]$ are Boolean, each is a propositional formula and the Markov Network is turned into a Knowledge Base of their conjunctions. Deciding probabilstic entailment is thus traced back to logical entailment.

4.3 Deciding Entailment by partial ordering

Classically entailment in propositional logics is defined by a model-theoretic approach. According to that approach, the entailment statement $\mathcal{KB} \models f$ holds, whenever any model of $\mathcal{KB}$ is also a model of f. We will in the following show, that this is equal to our definition based on probabilistic queries.

Definition 4.14 (Partial ordering of tensors) We say that two tensors f and h in a tensor space $\bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ are partially ordered, denoted by

$$f \prec h \,,$$

if for all $x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [m_k]$

$$f(x_0, \ldots, x_{d-1}) \leq h(x_0, \ldots, x_{d-1}) \,.$$

We notice, that whenever $f \prec h$ holds, for any model $x_0, \ldots, x_{d-1}$ of f we have

$$1 = f(x_0, \ldots, x_{d-1}) \leq h(x_0, \ldots, x_{d-1})$$

and thus $h(x_0, \ldots, x_{d-1}) = 1$. Therefore any model of f is also a model of h. We show in the next theorem, that this is equivalent to the entailment statement $f \models h$.

Theorem 4.15 (Partial Ordering Criterion) *We have $\mathcal{KB} \models f$ if and only if $\mathcal{KB} \prec f$.*

Proof. Directly by definition, since both $\mathcal{KB}$ and f are Boolean and therefore for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have that

$$\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] \leq f\left[X_{[d]} = x_{[d]}\right]$$

is equivalent to

$$\left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

Therefore, $\mathcal{KB} \prec f$ is equivalent to

$$\forall_{x_{[d]}\in\bigtimes_{k\in[d]}[2]}\left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right),$$

which is equal to $\mathcal{KB} \models f$. ∎

4.3.1 Monotonicity of Entailment

Vanishing local contractions provide sufficient but not necessary criterion to decide entailment, as we show in the next theorem.

Theorem 4.16 (Monotonicity of Entailment) *For any Markov Network on the decorated hypergraph $\mathcal{G}$ and any subgraph $\tilde{\mathcal{G}}$, we have for any formula that $\mathbb{P}^{\mathcal{G}} \models f$ if $\mathbb{P}^{\tilde{\mathcal{G}}} \models f$.*

Proof. Based on the reduction to Knowledge Bases by The. 4.11 and the monotonocity of binary contractions as shown in The. 13.2. ∎

Remark 4.17 To make use of The. 4.16 we can exploit any entailment criterion. However, there is no claim about entailment being false, when the entailment The. 4.16 therefore just provides a sufficient but not necessary criterion of entailment with respect to $\mathbb{P}^{\mathcal{G}}$. ♢

4.4 Deciding Entailment by local contractions

Global entailment can become inefficient, when

- we are interested in batches of entailment checks. Here we can make use of dynamic programming (store partial contraction results in the Knowledge Cores).
- the network is large. Although efficient tensor network contraction often work, they might get infeasible when the tensor network has a large connectivity. For many

An alternative to deciding entailment by global operations is the use of local operations. Here we interpret a part of the network (for example a single core) as an own knowledge base (with atomic formulas being the roots of the directed subgraph, that is potentially differing with the atoms in the global perspective) and perform entailment with respect to that.

Remark 4.18 Tradeoff between generality and efficiency While generic entailment decision algorithms (those by the full network) can decide any entailment, local algorithms as presented here can only perform some, but therefore more effectively as operating batchwise (dynamically deciding entailment for many leg variables). This is a typical phenomenon in logical reasoning and related to decidability. ♢

4.4.1 Knowledge Propagation

Let us now draw on these insights and store partial entailment results in Knowledge Cores, which is a use of the dynamic programming paradigm. We then iterate over local entailment checks, where we recursively add further entailment checks to be redone due to additional knowledge. We then call the local checks until convergence Entailment Propagation, since different stadia of knowledge are propagated through the network. We describe local Knowledge Propagation in a generic way in Algorithm 7.

Algorithm 7 Knowledge Propagation (KP)

Tensor Network $\mathcal{T}^{\mathcal{G}}$, $K^e = \mathbb{I}\left[X_e\right]$
while Stopping Criterion is not met **do**
Choose e, subset M of $\mathcal{T}^{\mathcal{G}}$ and of $\{K^e : e \in \mathcal{E}\}$ containing K^e
Update

$$K^e \leftarrow \mathbb{I}_{\neq 0} \circ \langle M \rangle \left[X_e\right]$$

end while

Each chosen subset M is understood as a local knowledge base, which is then applied for local entailment.

There are different ways of implementing Algorithm 7, by choosing an order of local knowledge bases M and a stopping criterion.

Theorem 4.19 *In Entailment Propagation Algorithm 7, K^e is monotonically decreasing with respect to the partial ordering and greater than $\tilde{K}^e$ defined as*

$$\tilde{K}^e = \mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [e]\right) .$$

Proof. We deduce the theorem from generic properties of the support of contractions, see Sect. 13.2. Monotonic decreasing follows from montonocity of tensor contractions, see The. 13.2. By The. 13.3 we have during any state of the algorithm

$$\mathbb{I}_{\neq 0} \circ \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\mathcal{V}}\right] = \mathbb{I}_{\neq 0} \circ \left\langle \mathcal{T}^{\mathcal{G}} \cup \{K^e : e \in \mathcal{E}\} \right\rangle \left[X_{\mathcal{V}}\right] .$$

If follows that

$$\tilde{K}^e = \mathbb{I}_{\neq 0} \circ \left\langle \mathcal{T}^{\mathcal{G}} \cup \{K^e : e \in \mathcal{E}\} \right\rangle \left[X_e\right]$$

and by The. 13.2

$$\tilde{K}^e \prec K^e .$$

■

Corollary 4.20 *Whenever for a formula $f\left[X_{\tilde{\mathcal{V}}}\right]$ and a K^e we have*

$$\left\langle K^e, \rho^f \right\rangle \left[Y_f = 0\right] = 0$$

then the Markov Network $\mathcal{T}^{\mathcal{G}}$ probabilistically entails f.

Another way to use Algorithm 7 to decide entailement of formulas f is adding each ρ^f to $\mathcal{T}^{\mathcal{G}}$ and defining Knowledge Cores $K^f[Y_f]$. Since then the Knowledge Core has only two dimensions, there are only four possible cores with the interpretation

- e_1: the formula is known to be true

- e_0: the formula is known to be false
- $\mathbb{I}$: the formula is not known
- 0: the knowledge base is inconsistent

Part II

Neuro-Symbolic Approaches

5

Formula Selecting Networks

In this chapter we will investigate efficient schemes to represent collections of formulas with similar structure in one tensor network.

Definition 5.1 Given a set of p formulas $\{f_l : l \in [p]\}$, we define the formula selecting map as

$$\mathcal{H}\left[X_{[d]}, L\right] : \bigtimes_{k\in[d]} [2] \times [p] \to [2]$$

defined for $l \in [p]$ by

$$\mathcal{H}\left[X_{[d]} = x_0, \ldots, x_{d-1}, L = l\right] = f_l\left[X_{[d]} = x_0, \ldots, x_{d-1}\right] .$$

We introduce a selection variable L and depict the formula selection in Figure 5.1.

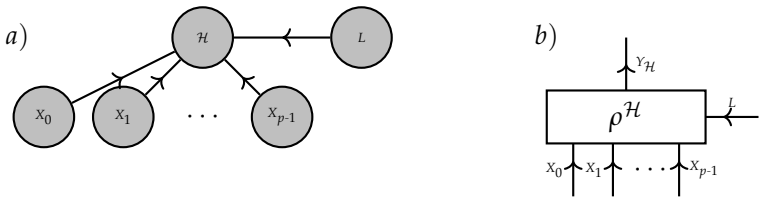


Figure 5.1: Representation of the Formula Selecting map as a a) Graphical Model with a selection variable $\mathcal{H}$. b) Dual Tensor Core with selection variable corresponding with an additional axis.

A naive representation of the formula selecting map is as a sum

$$\mathcal{H} = \sum_{l\in[p]} f_l\left[X_{[d]}\right] \otimes e_l\left[L\right] .$$

Such a representation scheme requires linear resources in the number of formulas. We will show in the following, that we can exploit common structure in formulas to drastically reduce this resource consumption.

5.1 Construction schemes

Let us now investigate efficient schemes to define sets of formulas to be used in the definition of $\mathcal{H}$. We will motivate the folding of the selection variable into multiple selection variables by compositions of selection maps.

5.1.1 Connective Selecting Tensors

We represent choices over connectives with a fixed number of arguments by adding a selection variable to the cores and defining each slice by a candidate connective.

Definition 5.2 Let $\{\circ_0, \ldots, \circ_{p_C-1}\}$ be a set of connectives with d arguments. The associated connective selection map is

$$\mathcal{H}_C\left[X_{[d]}, L_C\right] : \bigtimes_{k\in[d]} [2] \times [p_C] \to [2]$$

defined for each $l_C \in [p_C]$ and $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ by

$$\mathcal{H}_C\left[X_{[d]} = x_{[d]}, L_C = l_C\right] = \circ_{l_C}\left[X_{[d]} = x_{[d]}\right] .$$

We depict the relational encoding of connective selection maps in Figure 5.2.

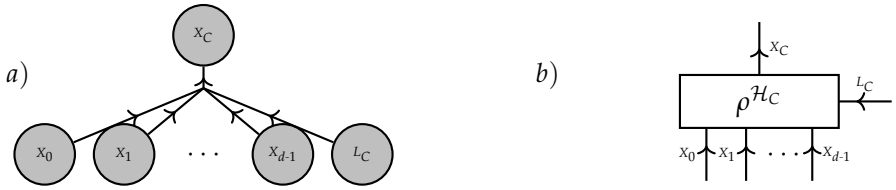


Figure 5.2: Connective Selector.

Remark 5.3 (HT Interpretation of Superposed Formula Tensor Networks) Continuing Remark **??**: Superposed Formula Tensors have a decomposition into a HT as sketched here, where we distinguish between formula selection subspaces (indices l_s) and atomic subspaces (indices x_k). At each formula selection we thus have a decomposition into three subspaces, two of atomic formulas and one for the formula selection. ♢

5.1.2 Variable Selecting Tensor Network

Definition 5.4 The selection of one out of p variables in a list $X_{[p]}$ is done by variable selecting maps

$$\mathcal{H}_V\left[X_{[p]}, L_V\right] : \left(\bigtimes_{l\in[p]} [2]\right) \times [p] \to [2]$$

are defined coordinatewise by

$$\mathcal{H}_V\left[X_0 = x_0, \ldots, X_{p-1} = x_{p-1}, L_V = l_V\right] = x_l .$$

Variable selecting maps appear in the literature as multiplex gates (see Definition 5.3 in [KF09]). The relational encoding of the variable selection map has a decomposition

$$\rho^{\mathcal{H}_V}\left[Y_V, X_{[p_V]}\right] = \sum_{l_V\in[p_V]} \rho^{X_{l_V}}\left[Y_V, X_{l_V}\right] \otimes e_{l_V}\left[L_V\right] .$$

This structure is exploited in the next theorem to derive a tensor network decomposition of $\rho^{\mathcal{H}_V}$.

Theorem 5.5 (Decomposition of Variable Selecting Maps) *Given a list $X_{[p_V]}$ of variables, we*

define for each $l_V \in [p_V]$ the tensors

$$T^{V_{l_V}}\left[X_{l_V}, L_V\right] = \delta\left[Y_V, X_{l_V}\right] \otimes e_{l_V}\left[L_V\right] + \mathbb{I}\left[Y_V, X_{l_V}\right] \otimes \left(\mathbb{I}\left[L_V\right] - e_{l_V}\left[L_V\right]\right) .$$

Then we have (see Figure 5.3)

$$\rho^{\mathcal{H}_V}\left[Y_V, X_{[p]}, L_V\right] = \left\langle \{T^{V_{l_V}}\left[Y_V, X_{l_V}, L_V\right] : l_V \in [p_V]\} \right\rangle \left[Y_V, X_{[p]}, L_V\right] .$$

Proof. We show the equivalence of the tensors on an arbitrary coordinates. For $\tilde{l}_V \in [p_V]$, $Y_V \in [2]$ and $x_{[p_V]} \in \bigtimes_{k\in[p_V]}[2]$ we have

$$\begin{aligned}
&\left\langle \{T^{V_{l_V}}\left[Y_V, X_{l_V}, L_V\right] : l_V \in [p_V]\} \right\rangle \left[Y_V = y_V, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V\right] \\
&= \prod_{l_V \in [p_V]} T^{V_{l_V}}\left[Y_V = y_V, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V\right] \\
&= T^{V_{\tilde{l}_V}}\left[Y_V = y_V, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V\right] \\
&= \begin{cases} 1 & \text{if} \quad y_V = x_{l_V} \\ 0 & \text{else} \end{cases} \\
&= \rho^{\mathcal{H}_V}\left[Y_V = y_V, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V\right]
\end{aligned}$$

In the second equality, we used that the tensor $T^{V_{l_V}}$ have coordinates 1 whenever $\tilde{l}_V \neq l_V$. ∎

The decomposition provided by The. 5.5 is in a CP format, as will be further discussed in Chapter 12. The introduced tensors $T^{V_{l_V}}$ are Boolean, but not directed and therefore encodings of relations but not functions (see Chapter 11).

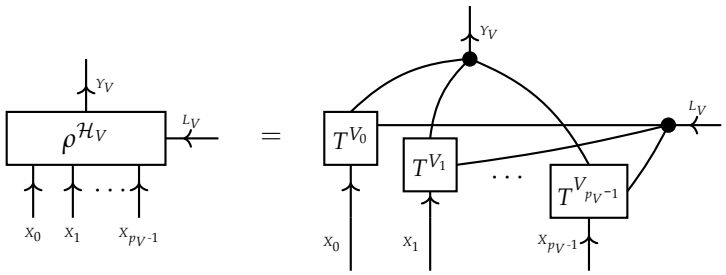


Figure 5.3: Decomposition of the relational encoding of a variable selecting tensor into a network of tensors defined in The. 5.5. The decomposition is in a CP-Format (see Chapter 12.

5.2 State Selecting Tensors

As an alternative, one can select a state of a categorical variable X.

Definition 5.6 Given a categorical variable X_S with dimension m_S and a selection variable L_S with dimension $p_S = m_S$ the state selecting tensor

$$\mathcal{H}_S\left[X_S, L_S\right] : [m_S] \times [p_S] \to [2]$$

is defined on $x_S \in [m_S]$ and $l_S \in [p_S]$ by

$$\mathcal{H}_S\left[X = x, L_S = l_S\right] = \begin{cases} 1 & \text{if} \quad x = l_S \\ 0 & \text{else} \end{cases} .$$

State selecting tensors can also be realized by variable selecting tensors. In Sect. 6.2.4 we will describe methods to build atomic variables indicating the states of a categorical variable. This would, however, increase the number of variables in a tensor network and can thus lead to an exponential overhead of dimensions. State selecting tensors can therefore be seen as a mean to avoid such dimension increases.

Comment: State Selectors can be integrated in Variable Selection framework. In this perspective, Variable selection networks are the specific case to $X = 1$.

5.3 Composition of formula selecting maps

We will now parametrize the sets $\mathcal{F}$ with additional indices and define formula selector maps subsuming all formulas. To handle large sets of formulas, we further fold the selection variable into tuples of selection variables.

Definition 5.7 Let there be a formula $f_{l_0,\ldots,l_{n-1}}$ for each index tuple in $l_0, \ldots, l_{n-1} \in \bigtimes_{s\in[n]}[p_s]$, where $n, p_0, \ldots, p_{n-1} \in \mathbb{N}$. The folded formula selector map (see Figure 5.4) is the map

$$\mathcal{H}\left[X_{[d]}, L_{[n]}\right] : \left(\bigtimes_{k\in[d]}[2]\right) \times \left(\bigtimes_{s\in[n]}[p_s]\right) \to [2]$$

with the coordinates at the indices $x_{[d]} \in \bigtimes_{k\in[d]}[2]$, $l_{[n]} \in \bigtimes_{s\in[n]}[p_s]$

$$\mathcal{H}\left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]}\right] = f_{l_{[n]}}\left[X_{[d]} = x_{[d]}\right] .$$

We will find formula selector maps by composition variables selector maps (Def. 5.4) and connective selector maps (Def. 5.2). This is especially useful to provide efficient decompositions of relational encodings.

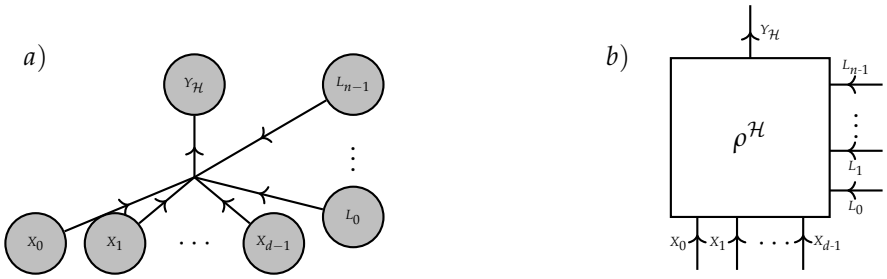


Figure 5.4: Relational encoding of the folded map $\mathcal{H}$.

5.3.1 Formula Selecting Neuron

The folding of the selection variable is motivated by the composition of selection maps. We call the composition of a connective selection with variable selection maps for each argument a formula selecting neuron.

Definition 5.8 Given an order $n \in \mathbb{N}$ let there be a connective selector $L_\circ$ selecting connectives of order n and let $\mathcal{H}_{V,0}, \ldots, \mathcal{H}_{V,n-1}$ be a collection of variable selectors. The corresponding logical neuron is the map

$$\sigma\left[X_{[d]}, L_{[n]}\right] : \left(\bigtimes_{k\in[d]} [2]\right) \times [p_C] \times \left(\bigtimes_{s\in[n]} [p_s]\right) \to [2]$$

defined for $x_{[d]} \in \bigtimes_{k\in[d]}[2]$, $l_C \in [p_C]$ and $l_0, \ldots, l_{n-1} \in \bigtimes_{s\in[n]}[p_s]$ by

$$\sigma(x_0, \ldots, x_{d-1}, l_C, l_0, \ldots, l_{n-1}) = \mathcal{H}_C(\mathcal{H}_{V,0}(x_0, \ldots, x_{d-1}, l_0), \ldots, \mathcal{H}_{V,n-1}(x_0, \ldots, x_{d-1}, l_{n-1}), l_C)\,.$$

Each neuron has a tensor network decomposition by a connective selector tensor and a variable selector tensor network for each argument, as we state in the next theorem.

Theorem 5.9 *Decomposition of formula selecting neurons Let σ a logical neuron, defined for a connective selector $L_\circ$ and variable selectors $\mathcal{H}_{V,0}, \ldots, \mathcal{H}_{V,n-1}$. Then we have (see Figure 5.5 for the example of $n = 2$):*

$$\begin{aligned}
&\rho^\sigma\left[Y_\sigma, X_{[d]}, L_C, L_{V,0}, \ldots, L_{V,n-1}\right] \\
&\quad = \langle\{\rho^{\mathcal{H}_C}\left[Y_\sigma, Y_{V,0}, \ldots, Y_{V,n-1}\right], \\
&\qquad \rho^{\mathcal{H}_{V,0}}\left[Y_{V,0}, X_{[d]}, L_{V,0}\right], \ldots, \rho^{\mathcal{H}_{V,n-1}}\left[Y_{V,n-1}, X_{[d]}, L_{V,n-1}\right]\}\rangle\left[Y_\sigma, X_{[d]}, L_C, L_{V,0}, \ldots, L_{V,n-1}\right]\,.
\end{aligned}$$

Proof. By composition The. 11.10. ∎

Example of a formula selecting neuron: Given a skeleton expression and a set of candidates at each placeholder, we parameterize a set of formulas by the assignment of candidate atoms to each placeholder position. Let us denote the set of formulas, which are generated through choosing atoms from $\mathcal{M}^s$ for the skeleton formula S by

$$\mathcal{F}_S := \left\{S\left(Z^1, \ldots, Z^d\right) \,:\, Z^k \in \mathcal{M}^k\right\}$$

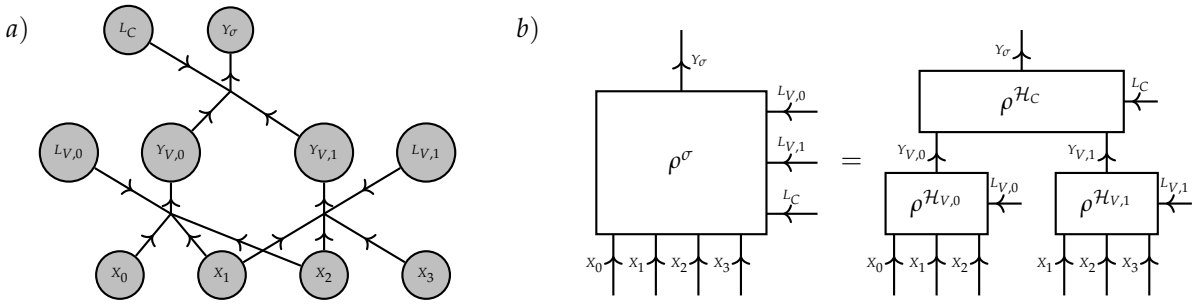


Figure 5.5: Example of a logical neuron σ of order $n = 2$. a) Selection and categorical variables and their interdependencies visualized in a hypergraph. b) Relational encoding of the logical neuron and tensor network decomposition into variable selecting and connective selecting tensors.

5.3.2 Formula Selecting Neural Network

Single neurons have a limited expressivity, since for each choice of the selection variables they can just express single connectives acting on atomic variables. The expressivity is extended to all

propositional formulas, when allowing for networks of neurons, which can select each others as input arguments.

Definition 5.10 An architecture graph $\mathcal{G}^{\mathcal{A}} = (\mathcal{V}^{\mathcal{A}}, \mathcal{E}^{\mathcal{A}})$ is an acyclic directed hypergraph with nodes appearing at most once as outgoing nodes. Nodes appearing only as outgoing nodes are input neurons and are labeled by $\mathcal{A}^{\text{in}}$ and nodes not appearing as outgoing nodes are the output neurons in the set $\mathcal{A}^{\text{out}}$ (see Figure 5.6 for an example).

Given an architecture graph $\mathcal{G}^{\mathcal{A}} = (\mathcal{V}^{\mathcal{A}}, \mathcal{E}^{\mathcal{A}})$, a *formula selecting neural network* $\mathcal{H}_{\mathcal{A}}$ is a tensor network of logical neurons at each $\sigma \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}}$, such that each neuron depends on variables $X_{\text{Pa}(\sigma)}$ and on selection variables L_σ. The collection of all selection variable is notated by $L_{\mathcal{A}}$.

The activation tensor of each neuron $\sigma \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}}$ is

$$\sigma^{\mathcal{A}} \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} \right] = \left\langle \{ \rho^{\tilde{\sigma}} \, : \, \tilde{\sigma} \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}} \} \cup \{ e_1 \left[Y_\sigma \right] \} \right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} \right] .$$

The activation tensor of the formula selecting neural network is the contraction

$$\mathcal{H}_{\mathcal{A}} \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} \right] = \left\langle \{ \rho^{\sigma^{\mathcal{A}}} \left[Y_\sigma, X_{\text{Pa}(\sigma)}, L_{\mathcal{A}} \right] \, : \, \sigma \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}} \} \cup \{ e_1 \left[Y_\sigma \right] \, : \, \sigma \in \mathcal{A}^{\text{out}} \} \right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} \right] .$$

The expressivity of a formula selecting neural network $\mathcal{H}_{\mathcal{A}}$ is the formula set

$$\mathcal{F}_{\mathcal{A}} = \left\{ \mathcal{A} \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} = l_{\mathcal{A}} \right] : l_{\mathcal{A}} \in \bigtimes_{s \in [n]} [p_s] \right\} .$$

The activation tensor of each neuron depends in general on the activation tensor of its ancestor neurons with respect to the directed graph $\mathcal{G}^{\mathcal{A}}$, and thus inherits the selection variables.

We notice that the architecture graph is a scheme to construct the variable dependency graph of the tensor network $\mathcal{F}_{\mathcal{A}}$. To this end, we replace each neuron $\sigma \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}}$ by an output variable Y_σ and further add selection variables L_σ to the directed edges, that is to each directed hyperedge $(\{\sigma\}, \text{Pa}(\sigma)) \in \mathcal{E}^{\mathcal{A}}$ we construct a directed hyperedge $(\{Y_\sigma\}, X_{\text{Pa}(\sigma)} \cup L_\sigma)$.

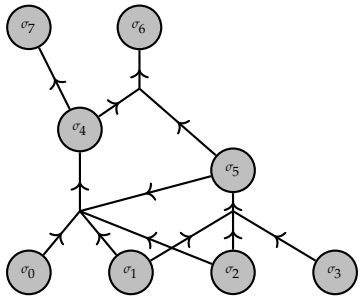


Figure 5.6: Example of an architecture graph $\mathcal{G}^{\mathcal{A}}$ with input neurons $\mathcal{A}^{\text{in}} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$ and output neurons $\mathcal{A}^{\text{out}} = \{\sigma_6, \sigma_7\}$

Theorem 5.11 *Given fixed selection variables $L_{\mathcal{A}}$, the formula selecting neural network is the conjunction of output neurons, that is*

$$\mathcal{H}_{\mathcal{A}} \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} \right] = \bigwedge_{\sigma \in \mathcal{A}^{\text{out}}} \sigma \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} \right] .$$

Proof. By effective calculus (see The. 11.21), we have

$$\left\langle \rho^{\wedge}\left[X_{\wedge}, X_{[d]}\right], e_1\left[X_{\wedge}\right]\right\rangle \left[X_{[d]}\right] = \bigotimes_{k\in[d]} e_1\left[X_k\right]$$

and thus

$$\mathcal{H}_{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \{\rho^{\sigma} : \sigma \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}\} \cup \{\rho^{\wedge}\left[X_{\wedge}, Y_{\sigma} : \sigma \in \mathcal{A}^{\text{out}}\right], e_1\left[X_{\wedge}\right]\}\right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right].$$

■

By the commutation of contractions, we can further use The. 5.9 to decompose each tensor ρ^{σ} into connective and variable selecting components to get a sparse representation of a formula selecting neural network $\mathcal{H}_{\mathcal{A}}$.

5.4 Application of Formula Selecting Networks

There are two main applications of formula selecting networks. First, when contracting the selection variables with a weight tensor we get a weighted sum of the parametrized formulas. Second, when contracting the categorical variables with a distribution or a knowledge base, we get a tensor storing the satisfaction rates respectively the world counts of the parametrized formulas.

5.4.1 Representation of selection encodings

In technical perspective: FSN provide efficient representation of $\gamma^{\mathcal{F}}$ -> Use for exponential families, structure learning.

Lemma 5.12 *Given a set* $\{f_{l_0,\ldots,l_{n-1}} : l_0,\ldots,l_{n-1} \in \bigtimes_{s\in[n]}[p_s]\}$ *of propositional formulas we define the statistic*

$$\mathcal{F} : x_0,\ldots,x_{d-1} \to (f_{l_0,\ldots,l_{n-1}}(x_0,\ldots,x_{d-1}))_{l_0,\ldots,l_{n-1}}.$$

and the formula selecting map

$$\mathcal{H} : x_0,\ldots,x_{d-1},l_0,\ldots,l_{n-1} \to f_{l_0,\ldots,l_{n-1}}(x_0,\ldots,x_{d-1}).$$

Then

$$\gamma^{\mathcal{F}}\left[X_{[d]}, L_{[n]}\right] = \mathcal{H}\left[X_{[d]}, L_{[n]}\right].$$

Proof. For any indices $l_{[n]} \in \bigtimes_{s\in[n]}[p_s]$ and $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have

$$\gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]}\right] = f_{l_0,\ldots,l_{n-1}}(x_0,\ldots,x_{d-1}) = \mathcal{H}\left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]}\right].$$

■

Technically, relational encodings have been exploited to derive decompositions based on basis calculus. Selection encodings on the other hand enable the application of formula selecting networks as superpositions of formulas.

5.4.2 Efficient Representation of Formulas

Weight contracted at the selection variables is elementary, then single formula retrieved.

Formula Selecting Neural Networks are means to represent exponentially many formulas with linear (in sum of candidates list lengths) storage. Their contraction with probability tensor

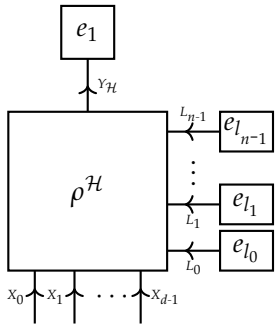
networks, is thus a batchwise evaluation of exponentially many formulas. This is possible due to redundancies in logical calculus due to modular combinations of subformulas.

We can retrieve specific formulas by slicing the selection variables, i.e. for $l_0, \dots, l_{n-1}$ we have

$$f_{l_0,\dots,l_{n-1}}[X_{[d]}] = \mathcal{H}\left[X_{[d]}, L = l_0, \dots, l_{n-1}\right] .$$

In a tensor network diagram we depict this by

$f_{l_0,\dots,l_{n-1}}$ =



Another perspective on the efficient formula evaluation by selection tensor networks is dynamic computing. Evaluating a formula requires evaluations of its subformulas, which are done by subcontractions and saved for different subformulas due to the additional selection legs.

However, we need to avoid contracting the tensor with leaving all selection legs open, since this would require exponential storage demand.

We can avoid this storage bottleneck by extending the contractions by additional cores leaving less variable legs open. This is the case when contracting gradients of the parameter tensor networks in alternating least squares approaches. Other methods avoiding the bottleneck can be constructed by MCMC sampling, for example Gibbs Sampling. Here we only need to vary local components of the formula reflected in keeping only single variable legs open.

5.4.3 Batch contraction of parametrized formulas

Given a set $\mathcal{F}$ of formulas, we build a formula selecting network parametrizing the formulas. The contraction

$$\left\langle \mathcal{T}^{\mathcal{G}}, \mathcal{H} \right\rangle \left[L_{[n]}\right]$$

is a tensor containing the contractions of the formulas $f_{l_{[n]}}$ with an arbitrary tensor network $\mathcal{T}^{\mathcal{G}}$ as

$$\left\langle \mathcal{T}^{\mathcal{G}}, f_{l_{[n]}} \right\rangle [\varnothing] = \left\langle \mathcal{T}^{\mathcal{G}}, \mathcal{H} \right\rangle \left[L_{[n]} = l_{[n]}\right] .$$

5.4.4 Average contraction of parametrized formulas

We show in the next two examples, how a full contraction of the formula selecting map with a probability distribution or a knowledge base can be interpreted.

Example 5.13 (Average satisfaction of formulas) The average of the formula satisfactions in $\mathcal{F}$ giben a probability tensor $\mathbb{P}$ is

$$\frac{1}{\prod_{s\in[n]} p_s} \cdot \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle [\varnothing] .$$

$\Diamond$

Example 5.14 (Deciding whether any formula is not contradicted) For example: We want to decide, whether there is a formula in $\mathcal{F}$ not contradicted by a Knowledge base $\mathcal{KB}$. This is the case if and only if

$$\left\langle \mathcal{KB}, \gamma^{\mathcal{F}} \right\rangle [\varnothing] = 0 .$$

We use Lem. 5.12 to get that $\gamma^{\mathcal{F}} = \mathcal{H}$. When the formulas are representable in a folded scheme, we find tensor network decompositions of $\mathcal{H}$ and exploit them along efficient representations of $\mathcal{KB}$ in an efficient calculation of $\left\langle \mathcal{KB}, \gamma^{\mathcal{F}} \right\rangle [\varnothing]$. This is further equal to

$$\mathcal{KB} \models \neg \left(\bigvee_{f \in \mathcal{F}} f \right) .$$

$\Diamond$

5.5 Examples of formula selecting neural networks

5.5.1 Correlation

For example (see Figure 5.7) consider the logical neuron with single activation candidate $\{\wedge\}$ and two variable selectors selecting d atomic variables $X_{[d]}$. The expressivity of this network is the set of all conjunctions of the atoms

$$\{X_k \wedge X_l \,:\, k, l \in [d]\}$$

Contracting with a probability distribution, we use the tensor

$$T\,[L_{V,0}, L_{V,1}] = \langle \mathcal{H}_{\mathcal{A}} \rangle\,[L_{V,0}, L_{V,1}]$$

to read of covariances as

$$\mathrm{Cov}(X_k, X_l) = T\,[L_{V,0} = k, L_{V,1} = l] - T\,[L_{V,0} = k, L_{V,1} = k] \cdot T\,[L_{V,0} = l, L_{V,1} = l] .$$

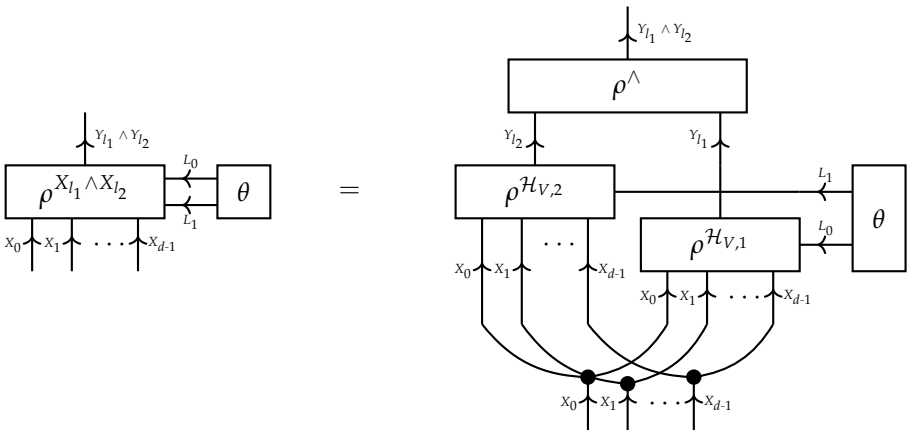


Figure 5.7: Superposition of the encoded formulas $\rho^{X_{l_1} \wedge X_{l_2}}$ with weight $\theta_{l_1 l_2}$

5.5.2 Conjunctive and Disjunctive Normal Forms

We can represent any propositional knowledge base by the following scheme: Literal selecting neurons by connective identity/negation (selecting positive/negative literal) and selecting one

of the atoms. Single output neuron representing the disjunction combining the literal selecting neurons. Number of neurons defined by the maximal clause size plus one. Smaller clauses can be covered when adding False as a possible choice (The respective neuron has to choose the identity, otherwise the full clause will be trivial).

The parameter core is in the basis CP format and each slice selects a clause of the knowledge base. When taking the slice values to infinity (e.g. by an annealing procedure), the represented member of the exponential family converges to the uniform distribution of the models of the knowledge base.

Useful to derive basis+ CP format based on CNF!

Remark 5.15 (Minterms and Maxterms) All minterms and maxterms can be represented by a two layer selection tensor networks without variable selection in two layers. The bottom layer has an $\neg$/Id connective selection neuron to each atom and the upper layer consists of a single $d$ary conjunction. ♢

5.6 Extension to variables of larger dimension

Connective selecting tensors: Can encode arbitrary functions h_l of discrete variables, but need $X_{\mathcal{H}_C}$ to be an enumeration of the states, in particular to be of dimension

$$m_{\mathcal{H}_C} = \left| \cup_{l \in [p]} \mathrm{im}\,(h_l) \right| .$$

Variable selecting tensors can be understood as specific cases of connective selecting tensors and can thus also be generalized in a straight forward manner by

$$m_{\mathcal{H}_C} = \left| \cup_{l \in [p]} \mathrm{im}\,(h_l) \right| .$$

State selecting tensors are directly

Example 5.16 (Discretization of a continuous neuron) Let there be a neuron

$$\sigma(w, y) : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R} .$$

When $w \in \mathcal{U}^{weight} \subset \mathbb{R}^d$ and $x \in \mathcal{U}^x \subset \mathbb{R}^d$ have

$$|\sigma(\langle w, x \rangle\,[])| \leq \left| \mathcal{U}^{weight} \right| \cdot |\mathcal{U}^x| .$$

To represent the discretization of the neuron, we use the subset encoding scheme of Def. 11.1. The variables O_{weight} indexing $\mathcal{U}^{weight}$ will be understood as selection incoming variables and the variables O_{weight} indexing $\mathcal{U}^{weight}$ as categorical incoming variables. We further define a variable O_σ indexing $\mathrm{im}\left(\sigma|_{\mathcal{U}^{weight} \times \mathcal{U}^x}\right)$ and have a tensor

$$\rho^\sigma \left[O_\sigma, O_{\mathcal{U}^x}, O_{weight} \right] .$$

If the neuron is of the form

$$\sigma(w, x) = \psi(\sum_i w_i \cdot x_i)$$

a decomposition into multiplication at each coordinate and summation of the results, with relational encodings for each, can be done. ♢

6

Logical Network Representation

Logic networks are graphical models with an interpretation by propositional logics. We first distinguish between Markov Logic Networks, which are an approach to soft logics in the framework of exponential families, and Hard Logic Networks, which correspond with propositional knowledge bases. Then we exploit non-trivial boolean base measures to unify both approaches by Hybrid Logic Networks, which are itself in exponential families.

6.1 Markov Logic Networks

Markov Logic Networks exploit the efficiency and interpretability of logical calculus as well as the expressivity of graphical models.

6.1.1 Markov Logic Networks as Exponential Families

We introduce Markov Logic Networks in the formalism of exponential families (see Sect. 1.7).

Definition 6.1 (Markov Logic Networks) Markov Logic Networks are exponential families $\Gamma^{\mathcal{F},\mathbb{I}}$ with sufficient statistics by functions

$$\mathcal{F} : \bigtimes_{k\in[d]} [2] \to \bigtimes_{f\in\mathcal{F}} [2] \subset \mathbb{R}^{|\mathcal{F}|}$$

defined coordinatewise by propositional formulas $f \in \mathcal{F}$.

Since the image of each feature is contained in $[2]$, they are propositional formulas (see Def. 3.2).

Conversely, any binary feature ϕ_l of an exponential family defines a propositional formula (see Def. 3.2). Thus, any exponential family of distributions of $\bigtimes_{k\in[d]}[2]$, such that $\operatorname{im}(\phi_l) \subset \{0,1\}$ for all $l \in [p]$ is a set of Markov Logic Networks with fixed formulas.

The sufficient statistics consistent in a map $\mathcal{F}$ of formulas brings the following advantages:

- Numerical Advantage: The sufficient statistics is decomposable into logical connectives. If the formulas are sparse (in the sense of limited number of connectives necessary in their representation), this gives rise to efficient tensor network decompositions of the relational encoding.

- Statistical Advantage: Since each formula is Boolean valued, the coordinates of the sufficient statistic are Bernoulli variables. Due to their boundedness, they and their averages (by Hoeffdings inequality) are sub-Gaussian variables with favorable concentration properties (absence of heavy tails).

Remark 6.2 (Alternative Definitions) We here defined MLNs on propositional logic, while originally they are defined in FOL. The relation of both frameworks will be discussed further in Chapter 9. ♢

6.1.2 Tensor Network Representation

Based on the previous discussion on the representation of exponential families by tensor networks in Sect. 1.7 we now derive a representation for Markov Logic Networks.

Theorem 6.3 (Relational Encodings for Markov Logic Networks) *A Markov Logic Network to a set of formulas $\mathcal{F} = \{f_l : l \in [p]\}$ is represented as*

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \{\rho^{f_l}\left[Y_l, X_{[d]}\right] : l \in [p]\} \cup \{W^{f_l,\theta[L=l]}[Y_{f_l}] : l \in [p]\} \right\rangle \left[X_{[d]}|\varnothing\right]$$

where we denote for each $l \in [p]$ an activation core

$$W^{f_l,\theta[L=l]}[Y_{f_l}] = \begin{bmatrix} 1 \\ \exp\left[\theta\left[L = l\right]\right] \end{bmatrix}[Y_l]\,.$$

Proof. The claim follows from Theorem 1.30 and the following contraction equations. We have with the grouped variable $Y_{\mathcal{F}} = \{Y_l : l \in [p]\}$

$$\rho^{\mathcal{F}}\left[Y_{\mathcal{F}}, X_{[d]}\right] = \left\langle \{\rho^{f_l}\left[Y_l, X_{[d]}\right] : l \in [p]\} \right\rangle \left[Y_{\mathcal{F}}, X_{[d]}\right]\,.$$

Since we have a Markov Logic Network we have $\mathrm{im}\,(f_l) \subset [2]$ and thus

$$W^{f_l,\theta[L=l]}\left[Y_l = y_l\right] = \begin{cases} 1 & \text{for} \quad y_l = 0 \\ \exp\left[\theta\left[L = l\right]\right] & \text{for} \quad y_l = 1 \end{cases}$$

Using these equations, the claim follows from Theorem 1.30. ∎

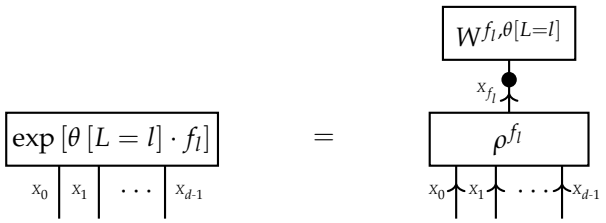


Figure 6.1: Factor of a Markov Logic Network to a formula f_l, represented as the contraction of a computation core ρ^{f_l} and an activation core $W^{f_l,\theta[L=l]}$. While the computation core ρ^{f_l} prepares based on basis calculus a categorical variable representing the value of the statistic formula f_l dependent on assignments to the distributed variables, the activation core multiplies an exponential weight to coordinates satisfying the formula.

Since any member of an exponential family is a Markov Network with tensors to each coordinate of the statistic, also Markov Logic Networks are Markov Networks.

Corollary 6.4 *Given a set $\mathcal{F}$ of formulas on atomic variables $X_{\mathcal{V}}$, we construct a $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ are decorated by the atoms and*

$$\mathcal{E} = \{\mathcal{V}^f : f \in \mathcal{F}\}\,,$$

where by $\mathcal{V}^f$ we denote the minimal set such that there exists a tensor $T\left[X_{\mathcal{V}^f}\right]$ with

$$f\left[X_{\mathcal{V}}\right] = T\left[X_{\mathcal{V}^f}\right] \otimes \mathbb{I}\left[X_{\mathcal{V}/\mathcal{V}^f}\right]\,.$$

Any Markov Logic Network $\mathcal{F}, \theta$ is then a Markov Network given the graph $\mathcal{G}^{\mathcal{F}}$ $\{\exp[\theta[L = l] \cdot f_l] : l \in [p]\}$.

Markov Logic Networks are Markov Networks with the factors given in a restricted form from the weighted truth of a formula. Each formula is seen as a factor of the graphical model.

There are two sparsity mechanisms drastically reducing the number of parameters (and loosing generality):

- Factors/Formulas contain only subsets of atoms (already in Corollary 6.4 exploited): The underlying assumptions of conditional independence loss generality.
- Structure in the factors: In MLN each factor corresponds with a formula evaluated on possible worlds. Again, any possible factor can be represented by a formula, but we concentrate on small formulas (see Theorem **??**).

We can extend the set of variables, by including the hidden formulas, and get a Markov Network of the relational encodings of connectives and headcores. Here hidden variables are additional variables facilitating the decomposition, but not appearing in open variables of contractions when doing reasoning. One can then exploit redundancies and make sure that every subresult is computed just once, by dropping relational encodings with identical head functions.

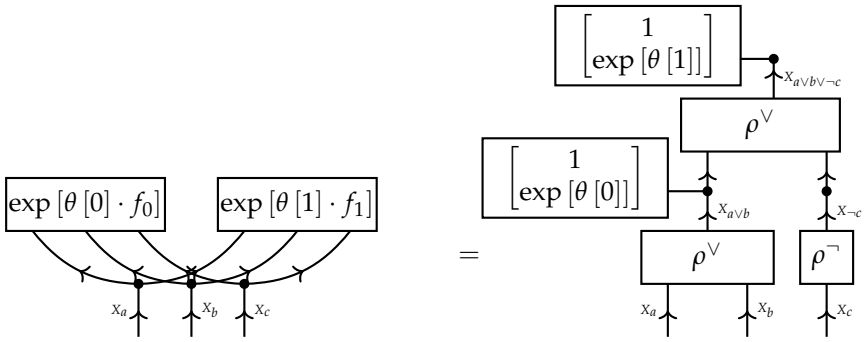


Figure 6.2: Example of a decomposed Markov Network representation of a Markov Logic Network with formulas $\{f_0 = a \vee b, f_1 = a \vee b \vee \neg c\}$. Since both formulas share the subformula $a \vee b$, their contracted factors have a representation by a connected tensor network.

6.1.3 Energy tensors

With the energy tensor

$$E^{\mathcal{F},\theta}\left[X_{[d]}\right] = \sum_{l \in [p]} \theta[L = l] \cdot f_l\left[X_{[d]}\right] = \left\langle \gamma^{\mathcal{F}}\left[X_{[d]}, L\right], \theta[L]\right\rangle \left[X_{[d]}\right]$$

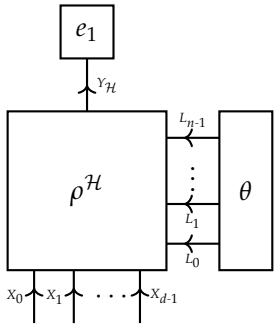
the MLN is the distribution

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \exp\left[E^{\mathcal{F},\theta}\right]\right\rangle \left[X_{[d]}|\varnothing\right] .$$

In case of a common structure of the formulas in a Markov Logic Network, Formula selecting networks can be applied to represent their energies.

We represent the superposition of formulas as a contraction with s parameter tensor. Given a factored parametrization of formulas $f_{l_0,\ldots,l_{n-1}}$ with indices l_s we have the superposition by the network representation:

$$\sum_{l_{[n]} \in \times_{s \in [n]} [p_s]} \theta \left[L_{[n]} = l_{[n]} \right] f_{l_{[n]}} \quad =$$



If the number of atoms and parameters gets large, it is important to represent the tensor $f_{l_0,\ldots,l_{n-1}}$ efficiently in tensor network format and avoid contractions. To avoid inefficiency issues, we also have to represent the parameter tensor θ in a tensor network format to improve the variance of estimations (see Chapter 8) and provide efficient numerical algorithms.

However, when required to instantiate the probability distribution of a Markov Logic Network as a tensor network, we need to exponentiate and normate the energy tensor, a task for which relational encodings are required. For such tasks, contractions of formula selecting networks are not sufficient and each formula with a nonvanishing weight needs to be instantiated as a factor tensor of a Markov Network.

6.1.4 Expressivity

Based on Markov Logic Networks containing only maxterms and minterms (see Def. 3.10), we here provide an expressivity study. There are 2^d maxterms and 2^d minterms which are enough to represent any probability distribution as we show next.

Theorem 6.5 *Let there be a positive probability distribution*

$$\mathbb{P}\left[X_{[d]}\right] \in \bigotimes_{k \in [d]} \mathbb{R}^2 .$$

Then the Markov Logic Network of minterms (see Def. 3.10)

$$\mathcal{F}_\wedge = \{ Z^\wedge_{x_0,\ldots,x_{d-1}} \,:\, x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [2] \}$$

with parameters

$$\theta\left[L_0 = x_0, \ldots, L_{d-1} = x_{d-1}\right] = \ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]$$

coincides with $\mathbb{P}\left[X_{[d]}\right]$.

Further, the Markov Logic Network of maxterms

$$\mathcal{F}_\vee = \{ Z^\vee_{x_0,\ldots,x_{d-1}} \,:\, x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [2] \}$$

with wparameters

$$\theta\left[L_0 = x_0, \ldots, L_{d-1} = x_{d-1}\right] = -\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]$$

coincides with $\mathbb{P}\left[X_{[d]}\right]$.

Proof. It suffices to show, that in both cases of choosing $\mathcal{F}$ by minterms or maxterms with the

respective parameters

$$E^{\mathcal{F},\theta} = \ln \mathbb{P}\left[X_{[d]}\right]$$

and therefore

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \exp\left[E^{\mathcal{F},\theta}\right]\right\rangle \left[X_{[d]}|\varnothing\right] = \left\langle \exp\left[E^{\mathcal{F},\theta}\right]\right\rangle \left[X_{[d]}\right] = \mathbb{P}\left[X_{[d]}\right] .$$

In the case of minterms, we notice that for any $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$

$$Z^{\wedge}_{x_0,\ldots,x_{d-1}}[X_{[d]}] = e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]$$

and thus with the weights in the claim

$$\sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} (\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]) \cdot Z^{\wedge}_{x_0,\ldots,x_{d-1}}[X_{[d]}] = \ln \mathbb{P}\left[X_{[d]}\right] .$$

For the maxterms we have analogously

$$Z^{\vee}_{x_0,\ldots,x_{d-1}}[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]$$

and thus that the maximal clauses coincide with the one-hot encodings of respective states. We thus have

$$\begin{aligned}
&\sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} (-\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]) \cdot Z^{\vee}_{x_0,\ldots,x_{d-1}}[X_{[d]}] \\
&= \left(\sum_{\mathcal{V}_0\subset[d]} (-\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]) \cdot \mathbb{I}\left[X_{[d]}\right]\right) \\
&\quad + \left(\sum_{\mathcal{V}_0\subset[d]} (\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]) \cdot e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]\right) \\
&= \ln \mathbb{P}\left[X_{[d]}\right] + \lambda \cdot \mathbb{I}\left[X_{[d]}\right] ,
\end{aligned}$$

where λ is a constant. ■

In general, this representation is redundant, since any offset of the weight by $\lambda \cdot \mathbb{I}$ results in the same distribution. However, the only $\bar{\theta}$ are multiples of $\mathbb{I}\left[X_{[d]}\right]$.

Theorem 6.5 is the analogue in Markov Logic to Theorem 3.13, which shows that any binary tensor has a representation by a logical formula, to probability tensors. Here we require positive distributions for well-defined energy tensors.

Remark 6.6 (Representation of Markov Networks) If a probability distribution is representable as a Markov Network, we only need to activate clauses and terms, which variables are contained in factors. Make a theorem out of that? ♢

6.1.5 Distribution of independent variables

We show next, the independent positive distributions are representable by tuning the d weights of the atomic formulas and keeping all other weights zero.

Theorem 6.7 *Let $\mathbb{P}\left[X_{[d]}\right]$ be a positive probability distribution, such that disjoint subsets of atoms are independent from each other. Then $\mathbb{P}\left[X_{[d]}\right]$ is the Markov Logic Network of atomic formulas*

$$\mathcal{F}_{[d]} = \{X_k : k \in [d]\}$$

and parameters

$$\theta[L = k] = \ln\left[\frac{\langle\mathbb{P}\rangle[X_k = 1]}{\langle\mathbb{P}\rangle[X_k = 0]}\right]$$

Proof. By Theorem 1.12 we get a decomposition

$$\mathbb{P}\left[X_{[d]}\right] = \bigotimes_{k\in[d]} \mathbb{P}^k[X_k]$$

where

$$\mathbb{P}^k[X_k] = \langle\mathbb{P}\rangle[X_k] .$$

By assumption of positivity, the vector $\mathbb{P}^k[X_k]$ is positive for each $k \in [d]$ and the parameter

$$\theta[L = k] = \ln\left[\frac{\mathbb{P}^k[X_k = 1]}{\mathbb{P}^k[X_k = 0]}\right]$$

well-defined.

We then notice, that

$$\mathbb{P}^{(\{X_k\},\theta[L=k])}[X_k] = \mathbb{P}^k[X_k]$$

and therefore with the parameter vector of dimension $p = d$ defined as

$$\theta[L] = \sum_{k\in[d]} \theta[L = k] \cdot e_k[L]$$

we have

$$\begin{aligned}\mathbb{P}^{(\{X_k : k\in[d]\},\theta)}[X_{[d]}] &= \bigotimes_{k\in[d]} \mathbb{P}^{(\{X_k\},\theta[L=k])}[X_k] \\ &= \bigotimes_{k\in[d]} \mathbb{P}^k[X_k] \\ &= \mathbb{P}\left[X_{[d]}\right] .\end{aligned}$$

■

In Theorem 6.7 we made the assumption of positive distributions. If the distribution fails to be positive, we still get a decomposition into distributions of each variable, but there is at least one factor failing to be positive. Such factors need to be treated by hybrid logic networks, that is they are base measure for an exponential family coinciding with a logical literal (see Sect. 6.2).

All atomic formulas can be selected by a single variable selecting tensor, that is

$$E^{(\{X_k : k\in[d]\},\theta)}[X_{[d]}] = \left\langle \mathcal{H}_V\left[X_{[d]}, L\right], \theta[L]\right\rangle\left[X_{[d]}\right] .$$

In case of negative coordiantes $\theta[L = k]$ it is convenient to replace X_k by $\neg X_k$, in order to facilitate the interpretation. The probability distribution is left invariant, when also replacing $\theta[L = k]$ by $-\theta[L = k]$.

6.1.6 Boltzmann machines

A Boltzmann machine is a member of an exponential family with the energy tensor (see e.g. Chapter 43 in [Mac03])

$$E^{W,b}[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] = \sum_{k,l \in [d]} W[L_{V,0} = k, L_{V,1} = l] \cdot x_k \cdot x_l + \sum_{k \in [d]} b[L_{V,0} = k] \cdot x_k \,.$$

We notice, that this coincides with the energy tensor of a Markov Logic Network with formula set

$$\mathcal{F} = \{X_k \Leftrightarrow X_l \,:\, k, l \in [d]\} \cup \{X_k \,:\, k \in [d]\}$$

with cardinality $d^2 + d$.

Each formula is in the expressivity of an architecture consisting of a single binary logical neuron selecting any variable of $X_{[d]}$ in each argument and selecting connectives $\{\Leftrightarrow, \triangleleft\}$, where by $\triangleleft$ we refer to a connective passing the first argument, defined for $x_0 \in [m_0], x_1 \in [m_1]$ as

$$\triangleleft[X_0 = x_0, X_1 = x_1] = \mathcal{H}_V\left[X_0 = x_0, X_1, L_V = 0\right] .$$

The weight is

$$\theta = e_0\left[L_C\right] \otimes W + e_1\left[L_C\right] \otimes b[L_{V,0}] \otimes e_0\left[L_{V,0}\right]$$

And we have

$$E^{W,b}[X_{[d]}] = \left\langle \mathcal{H}_{\mathcal{A}}\left[X_{[d]}, L_C, L_{V,0}, L_{V,1}\right], \theta\left[L_C, L_{V,0}, L_{V,1}\right]\right\rangle \left[X_{[d]}\right] .$$

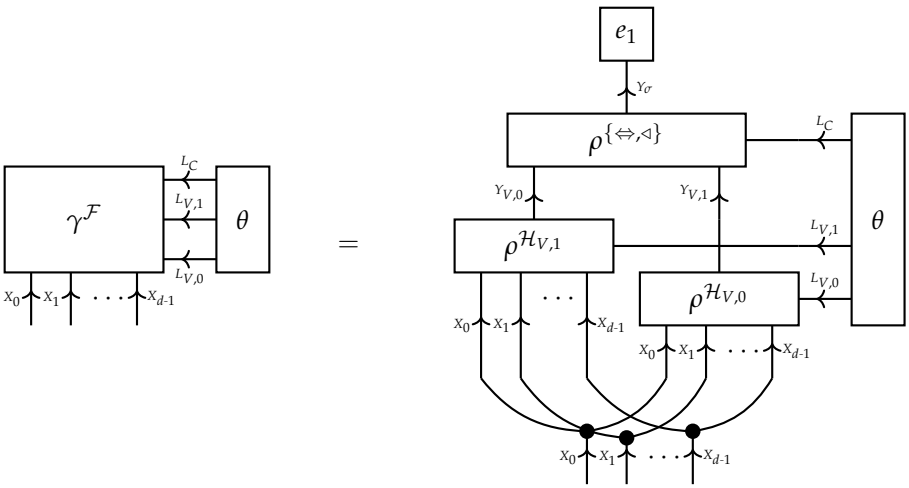


Figure 6.3: Tensor network representation of the energy of a Boltzmann machine

Often Boltzmann machines are formulated with hidden variables. To average those out, one needs to instantiate the probability distribution instead of the energy tensor and leave only visible variables open in a contraction.

Markov Logic Networks go beyond the Boltzmann machines already for binary formulas, by the flexibility to capture further dependencies beyond the correlation. We can use any binary logical connective and have an associated formula where we can put a weight on.

6.2 Hard Logic Networks

While exponential families are positive distributions, in logics probability distributions can assign states zero probability. As a consequence, Markov Logic Networks have a soft logic interpretation in the sense that violation of activated formulas have nonzero probability. We here discuss their hard logic counterparts, where worlds not satisfying activated formulas have zero probability.

6.2.1 The limit of hard logic

The probability function of Markov Logic Networks with positive weights mimiks the tensor network representation of the knowledge base, which is the conjunction of the formulas. The maxima of the probability function coincide with the models of the corresponding knowledge base, if the latter is satisfiable. However, since the Markov Logic Network is defined as a normed exponentiation of the weighted formula sum, it is a positive distribution whereas uniform distributions among the models of a knowledge base assign zero probability to world failing to be a model. Since both distributions are tensors in the same space to a factored system, we can take the limits of large weights and observe, that Markov Logic Networks indeed converge to normed knowledge bases.

Lemma 6.8 *For any satisfiable formula $f\left[X_{[d]}\right]$ and a variable weight $\theta \in \mathbb{R}$, we have for $\theta \to \infty$*

$$\left\langle \exp\left[\theta \cdot f\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right] \to \langle f\rangle \left[X_{[d]}|\varnothing\right]$$

and for $\theta \to -\infty$

$$\left\langle \exp\left[\theta \cdot f\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right] \to \langle \neg f\rangle \left[X_{[d]}|\varnothing\right] .$$

Here we denote the understand the convergence of tensors as a convergence of each coordinate.

Proof. We have

$$\mathcal{Z}(\mathcal{F},\theta) = \left(\prod_{k\in[d]} m_k\right) - \langle f\rangle[\varnothing] + \langle f\rangle[\varnothing]\cdot\exp[\theta]$$

and therefore for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ with $f\left[X_{[d]} = x_{[d]}\right] = 1$

$$\begin{aligned}\langle \exp[\theta\cdot f]\rangle \left[X_{[d]} = x_{[d]}|\varnothing\right] &= \frac{\exp[\theta]}{\left(\prod_{k\in[d]} m_k\right) - \langle f\rangle[\varnothing] + \langle f\rangle[\varnothing]\cdot\exp[\theta]} \\ &\to \frac{1}{\langle f\rangle[\varnothing]} = \langle f\rangle[X_0 = x_0,\ldots,X_{d-1} = x_{d-1}|\varnothing] .\end{aligned}$$

For any $x_0,\ldots,x_{d-1} \in \bigtimes_{k\in[d]}[2]$ with $f\left[X_{[d]} = x_{[d]}\right] = 0$ we have on the other side

$$\begin{aligned}\langle \exp[\theta\cdot f]\rangle [X_0 = x_0,\ldots,X_{d-1} = x_{d-1}|\varnothing] &= \frac{1}{\left(\prod_{k\in[d]} m_k\right) - \langle f\rangle[\varnothing] + \langle f\rangle[\varnothing]\cdot\exp[\theta]} \\ &\to 0 = \langle f\rangle[X_0 = x_0,\ldots,X_{d-1} = x_{d-1}|\varnothing] .\end{aligned}$$

■

We can by the above Lemma represent both the situation of non-asymptotic weights and the

limit for diverging weights by the same computation core $\rho^f\left[Y_f, X_{[d]}\right]$, with different activation cores, since

$$\left\langle \exp\left[\theta \cdot f\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right] = \left\langle \rho^f\left[Y_f, X_{[d]}\right], W^{f,\theta}\right\rangle \left[X_{[d]}\right]$$

and

$$\langle f\rangle \left[X_{[d]}|\varnothing\right] = \left\langle \rho^f\left[Y_f, X_{[d]}\right], e_1\left[Y_f\right]\right\rangle \left[X_{[d]}\right] \quad \text{respectively} \quad \langle \neg f\rangle \left[X_{[d]}|\varnothing\right] = \left\langle \rho^f\left[Y_f, X_{[d]}\right], e_0\left[Y_f\right]\right\rangle \left[X_{[d]}\right].$$

Theorem 6.9 *Let $\mathcal{F}$ be a formulaset and θ a positive parameter vector. If the formula*

$$\mathcal{KB} = \bigwedge_{f\in\mathcal{F}} f$$

is satisfiable we have in the limit $\beta \to \infty$ the coordinatewise convergence

$$\mathbb{P}^{(\mathcal{F},\beta\cdot\theta)}[X_{[d]}] \to \langle \mathcal{KB}\rangle \left[X_{[d]}|\right].$$

Proof. Since $\mathcal{KB}$ is satisfiable we find $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ with

$$\left\langle \exp\left[\sum_{f\in\mathcal{F}} \beta\cdot\theta_f\cdot f\right]\right\rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] = \exp\left[\beta\cdot\sum_{f\in\mathcal{F}}\theta_f\right]$$

and the partition function obeys

$$\left\langle \exp\left[\sum_{f\in\mathcal{F}} \beta\cdot\theta_f\cdot f\right]\right\rangle [\varnothing] \geq \exp\left[\beta\cdot\sum_{f\in\mathcal{F}}\theta_f\right].$$

For any state $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ with $\mathcal{KB}(x_0, \ldots, x_{d-1}) = 0$ we find $h \in \mathcal{F}$ with $h(x_0, \ldots, x_{d-1}) = 0$ and have

$$\frac{\left\langle \exp\left[\sum_{f\in\mathcal{F}}\beta\cdot\theta_f\cdot f\right]\right\rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1}]}{\left\langle \exp\left[\sum_{f\in\mathcal{F}}\beta\cdot\theta_f\cdot f\right]\right\rangle [\varnothing]} \leq \frac{\exp\left[\beta\cdot\sum_{f\in\mathcal{F}:f\neq h}\theta_f\right]}{\exp\left[\beta\cdot\sum_{f\in\mathcal{F}}\theta_f\right]} = \exp\left[\beta\cdot\theta_h\right] \to 0.$$

The limit of the distribution has thus support only on the models of $\mathcal{KB}$. Since any model of $\mathcal{KB}$ has same energy at any β the limit is a uniform distribution and coincides therefor with

$$\langle \mathcal{KB}\rangle \left[X_{[d]}|\right].$$

■

Remark 6.10 (More generic situation of simulated annealing) The process of taking $\beta \to \infty$ is known as simulated annealing, see Chapter 2. From the discussion there we have the more general statement, that the limiting distribution is the uniform distribution among the maxima of $\mathbb{P}^{(\mathcal{F},\theta)}[X_{[d]}]$. If the formula $\mathcal{KB}$ is not satisfiable the normation $\langle \mathcal{KB}\rangle \left[X_{[d]}|\varnothing\right]$ does not exist and the limit distribution has another syntactical representation, to be gained e.g. by minterm or maxterm representation (see Theorem 3.13). ♢

6.2.2 Tensor Network Representation

Hard Logic Network coincide with Knowledge Bases and are thus representable by contractions of formulas (which can be interpreted as an effective calculus scheme, see Sect. 11.5). We use $\wedge$ symmetry to represent them as a contraction of the formulas building the Knowledge Base as conjunction.

Theorem 6.11 (Conjunction Decomposition of Knowledge Bases) *For a Knowledge Base*

$$\mathcal{KB} = \bigwedge_{f \in \mathcal{F}} f$$

we have

$$\mathcal{KB}\left[X_{[d]}\right] = \left\langle f\left[X_{[d]}\right]\right\rangle \left[X_{[d]}\right]$$

and

$$\mathcal{KB}\left[X_{[d]}\right] = \left\langle \{\rho^f\left[Y_f, X_{[d]}\right] : f \in \mathcal{F}\} \cup \{e_1\left[Y_f\right] : f \in \mathcal{F}\}\right\rangle \left[X_{[d]}\right].$$

Proof. By the $\wedge$-symmetry, see effective calculus and

$$f\left[X_{[d]}\right] = \left\langle \{\rho^f\left[Y_f, X_{[d]}\right], e_1\left[Y_f\right]\}\right\rangle \left[X_{[d]}\right]$$

■

Remark 6.12 $\wedge$ symmetry does not generalize to Markov Logic Networks In Markov Logic, similar decompositions are not possible. For example, consider a MLN with a single formula $X_0 \wedge X_1$ and nonvanishing weight θ. This does not coincide with the distribution of a MLN of two formulas X_0 and X_1. To see this, we notice that with respect to the distribution of the first MLN, both variables are not independent, while for any MLN constructed by the two atomic formulas they are. ◊

6.2.3 Polynomial Representation

We now apply the representation symmetries to represent a propositional Knowledge Base in conjunctive normal form. A Knowledge Base in Conjunctive Normal Form is a conjunction of clauses, where clauses are disjunctions of literals being atoms (positive literals) or negated atoms (negative literals).

Formulas can be represented as sparse polynomials, which will be discussed in more detail in Chapter 12 (see Def. 12.6).

Lemma 6.13 *Any term is representable by a single monomial and any clause is representable by at most two monomials.*

Proof. Let $\mathcal{V}_0$ and $\mathcal{V}_1$ be disjoint subsets of $\mathcal{V}$, then we have

$$Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1} = e_{\{x_k=0:k\in\mathcal{V}_0\}\cup\{x_k=1:k\in\mathcal{V}_1\}}\left[X_{\mathcal{V}_0\cup\mathcal{V}_1}\right] \otimes \mathbb{I}\left[X_{\mathcal{V}/(\mathcal{V}_0\cup\mathcal{V}_1)}\right]$$

and

$$Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1} = \mathbb{I}\left[X_{\mathcal{V}}\right] - e_{\{x_k=0:k\in\mathcal{V}_0\}\cup\{x_k=1:k\in\mathcal{V}_1\}}\left[X_{\mathcal{V}_0\cup\mathcal{V}_1}\right] \otimes \mathbb{I}\left[X_{\mathcal{V}/(\mathcal{V}_0\cup\mathcal{V}_1)}\right].$$

We notice, that any tensors $\mathbb{I}$ and $e_x \otimes \mathbb{I}$ habe basis+-rank of 1 and therefore $Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1}$ of 1 and $Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1}$ of at most 2. ■

We apply Lem. 6.13 to show the following sparsity bound on the energy tensor of Markov Logic Networks.

Theorem 6.14 *Any formula f with a conjunctive normal form of n clauses satisfies*

$$\operatorname{rank}^d(f) \leq 2^n .$$

For any set $\mathcal{F}$ of formulas each with a conjunctive normal form of n_f clauses satisfies for any θ

$$\operatorname{rank}^d\left(\sum_{f\in\mathcal{F}} \theta_f \cdot f\right) \leq \sum_{f\in\mathcal{F}} 2^{n_f} .$$

Proof. Let f have a CNF with clauses indexed by $l \in [n]$ and each clause represented by subsets $\mathcal{V}_0^l, \mathcal{V}_1^l$, that is

$$f = \bigwedge_{l\in[n]} Z^{\vee}_{\mathcal{V}_0^l,\mathcal{V}_1^l} .$$

We now use the rank bound of Theorem 12.12 and Lem. 6.13 to get

$$\operatorname{rank}^d(f) \leq \prod_{l\in[n]} \operatorname{rank}^d\left(Z^{\vee}_{\mathcal{V}_0^l,\mathcal{V}_1^l}\right) \leq 2^n .$$

Given a collection of formulas $\mathcal{F}$, each with a CNF of n_f clauses we apply Theorem 12.11 and get

$$\operatorname{rank}^d\left(\sum_{f\in\mathcal{F}} \theta_f \cdot f\right) \leq \sum_{f\in\mathcal{F}} \operatorname{rank}^d(f) \leq \sum_{f\in\mathcal{F}} 2^{n_f} .$$

■

6.2.4 Categorical Constraints

We made the assumption that all categorical variables in factored systems to be represented by propositional logics take binary values (i.e. $m = 2$). In cases where a categorical variable X takes multiple values we define for each x an atomic formula X_x representing whether X is assigned by x in a specific state. Following this construction we have the constraint that exactly one of the atoms X_x is 1 at each state.

To capture the constraints resulting from this construction we introduce auxiliary parts. Such constraints can also be expressed by a formula but would result in an unnecessary large tensor network.

Definition 6.15 (Categorical Constraint and Atomization Variables) Given a list $X_0, \ldots, X_{m-1}$ of boolean variables and a categorical variable X with dimension m a categorical constraint is a tensor $Z[X, X_{[m]}]$ defined as

$$Z(x, x_A) = \begin{cases} 1 & \text{if} \quad x_{[m]} = e_x \\ 0 & \text{else} . \end{cases}$$

We then call the variables $X_0, \ldots, X_{m-1}$ the atomization variables to the categorical variable X.

With Theorem 12.20 the tensor representation of ρ^Z decomposes in a basis CP format (see

Figure 6.4b) of if its coordinate maps Z_x, where $x \in [m]$. For the cores

$$\rho^{Z_x} = e_x\left[X\right] \otimes e_1\left[X_x\right] + \left(\mathbb{I}\left[X\right] - e_x\left[X\right]\right) \otimes e_0\left[X_x\right]$$

we have by Theorem 12.20

$$\rho^{Z}\left[X, X_0, \ldots, X_{m-1}\right] = \left\langle \{\rho^{Z(x)} \,:\, x \in [m]\} \right\rangle \left[X, X_0, \ldots, X_{m-1}\right] .$$

In the next theorem we show how a categorical constraint can be enforced in a tensor network by adding the tensor Z to a contraction.

Theorem 6.16 *For any tensor $T\left[X_{[d]}\right]$ and a categorical constraint defined by an ordered subset $X_A \subset X_{[d]}$, a variable $X \in X_{[d]}$ we have*

$$\left\langle \{T\left[X_{[d]}\right], Z\} \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = \begin{cases} T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] & \text{if} \quad x_A = e_x \\ 0 & \text{else} . \end{cases}$$

Proof. For any $x_{[d]}$ we have

$$\left\langle \{T\left[X_{[d]}\right], Z\} \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = T\left[x_{[d]}\right] \cdot Z[X = x, X_A = x_A] .$$

If $x_A = e_x$ we have $Z[X = x, X_A = x_A] = 1$ and thus

$$\left\langle T\left[X_{[d]}\right], Z \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = T\left[x_{[d]}\right] .$$

If $x_A \neq e_x$ then $Z[X = x, X_A = x_A] = 0$ and

$$\left\langle T\left[X_{[d]}\right], Z \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = 0 .$$

■

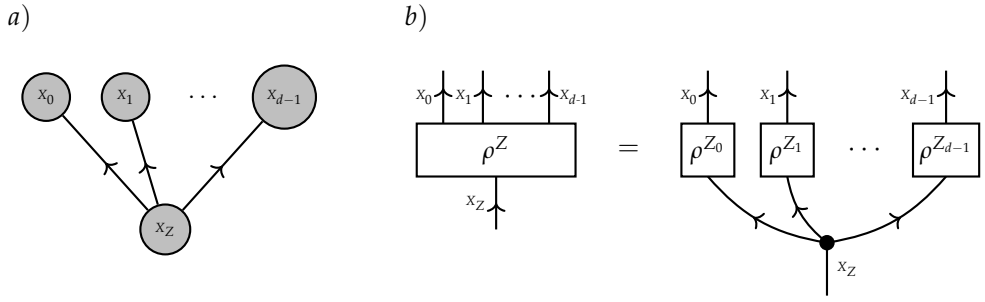


Figure 6.4: Representation of a categorical constraint in a CP Format tensor network. a) Representation of the dependency of the graphical model. b) Tensor Representation with further network decomposition. We average by contraction with the dashed tensor $\mathbb{I}$, if we do not specify the active atom.

Remark 6.17 (Combination of Constraints) We can combine constraint cores by Hadamard products in the dual tensor network representation, as long as they can be satisfied together. An example, where this is not the case, are the categorical constraints to the three sets

$$\{X_0, X_1, X_2, X_3\} , \{X_0, X_1\} , \{X_2, X_3\} .$$

Besides the categorical cores also the datacores have a similar bayesian network affecting the atoms by another hidden variable. Combining both is welldefined, only when all datapoints satisfy the categorical constraints (that is only one of the atoms in each constraint is active). ♢

Example 6.18 (Sudoku) An interesting example, where categorical constraints are combined is Sudoku, the game of assigning numbers to a grid (see for example Section 5.2.6 in [RN21]). The basic variables therein are $X_{i,j}$, with $m_{i,j} = n^2$ and $i, j \in [n^2]$. By understanding i as a line index and j as a column index, they are ordered in a grid as sketched in Figure **??** in the case $n = 3$.

For a $n \in \mathbb{N}$ we further define the atomization variables $X_{i,j,k}$ where $i, j, k \in [n^2]$ and $m_{i,j,k} = 2$. These n^6 variables are the booleans indicating whether a specific position has a specific number assigned. The consistency of the atomization variables to the basic variables is then for each $i, j \in [n^2]$ ensured by the constraints

$$\{X_{i,j,k} : k \in [n^2]\}.$$

We further have $3 \cdot n^2$ constraints by the

- Row constraints: Each number k appears exactly once in each row $i \in [n^2]$, captured by the constraints
$$\{X_{i,j,k} : j \in [n^2]\}.$$
- Column constraints: Each number k appears exactly once in each column $j \in [n^2]$, captured by the constraints
$$\{X_{i,j,k} : i \in [n^2]\}.$$
- Square constraints: Each number appears exactly once in each square $s, r \in [n]$, captured by the constraints
$$\{X_{i+n\cdot s,j+n\cdot r,k} : i, j \in [n]\}.$$

In total we have $3 \cdot n^2 + n^4$ constraints for n^6 variables.

Reasoning by Entailment propagation! Also, probabilistic choices possible when exact (!) contraction at a position not a basis vector, then can choose one possibility.

♢

6.3 Hybrid Logic Network

Markov Logic Networks are by definition positive distributions. In contrary, Hard Logic Networks model uniform distributions over model sets of the respective Knowledge Base and therefore have vanishing coordinates. We now show how to combine both approaches by defining Hybrid Logic Networks, when understanding Hard Logic Networks as base measures. This trick is known to the field of variational inference, see for Example 3.6 in [WJ08].

Definition 6.19 Given a set of formulas $\mathcal{F}$ with weights θ and set $\mathcal{KB}$ of formulas, which conjunction is satisfiable, the hybrid logic network is the probability distribution

$$\mathbb{P}^{(\mathcal{F},\theta,\nu^{\mathcal{KB}})}[X_{[d]}] = \left\langle \{f : f \in \mathcal{KB}\} \cup \{\exp\left[\theta_f \cdot f\right] : f \in \mathcal{F}\} \right\rangle \left[X_{[d]}|\varnothing\right],$$

which is the member of the exponential family with statistic by $\mathcal{F}$ and the base measure

$$\nu^{\mathcal{KB}}[X_{[d]}] = \langle \{f : f \in \mathcal{KB}\} \rangle \left[X_{[d]}\right].$$

Given a set of formulas $\mathcal{F}$, we define the set of hybrid logic networks realizable with $\mathcal{F}$

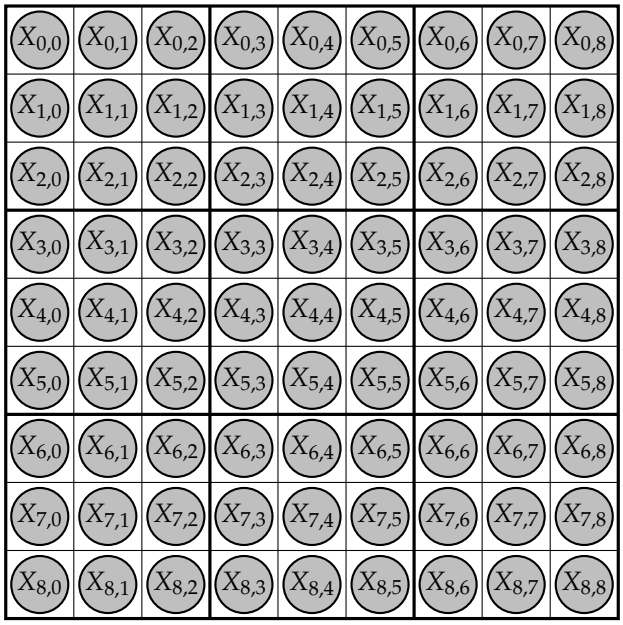


Figure 6.5: Sudoku grid of basic categorical variables $X_{i,j}$, here drawn in the standard case of $n = 3$, each with dimension $m = n^2 = 9$. Each basic categorical variables has n^2 corresponding atomization variables, which are further atomization variables to the row, column and squares constraints. Instead of depicting those constraints by hyperedges in a variable dependency graph, we here just indicate their existence through row, column and squares blocks.

and elementary activation cores as

$$\Lambda^{\mathcal{F},\mathrm{EL}} = \bigcup_{\tilde{\mathcal{F}}\subset\mathcal{F},\,\mu\in\{0,1\}^{|\mathcal{F}|}} \Gamma^{\mathcal{F}/\tilde{\mathcal{F}},\nu^{\tilde{\mathcal{F}},\mu}}$$

where we denote base measures

$$\nu^{\tilde{\mathcal{F}},\mu}\left[X_{[d]}\right] = \bigwedge_{f_l\in\tilde{\mathcal{F}}} \neg^{(1-\mu[L=l])} f_l\left[X_{[d]}\right].$$

The assumption of a satisfiable set $\mathcal{KB}$ is necessary, as we show next.

Theorem 6.20 *If any only if $\bigwedge_{f\in\mathcal{KB}} f$ is satisfiable, the tensor*

$$\left\langle \{f : f\in\mathcal{KB}\} \cup \{\exp\left[\theta_f\cdot f\right] : f\in\mathcal{F}\}\right\rangle\left[X_{[d]}\right]$$

is normable.

Proof. We need to show that

$$\left\langle \{f : f\in\mathcal{KB}\} \cup \{\exp\left[\theta_f\cdot f\right] : f\in\mathcal{F}\}\right\rangle[\varnothing] > 0\,. \tag{6.1}$$

Since the conjunction of $\mathcal{KB}$ is satisfiable we find a $x_{[d]}$ with $f\left[X_{[d]} = x_{[d]}\right] = 1$ for all $f\in\mathcal{KB}$.

Then

$$\begin{aligned}\left\langle \{f : f \in \mathcal{KB}\} \cup \{\exp\left[\theta_f \cdot f\right] : f \in \mathcal{F}\} \right\rangle \left[X_{[d]} = x_{[d]}\right] &= \left(\prod_{f \in \mathcal{KB}} f\left[X_{[d]} = x_{[d]}\right]\right) \cdot \left(\prod_{f \in \mathcal{F}} \exp\left[\theta_f \cdot f\right]\left[X_{[d]} = x_{[d]}\right]\right) \\ &= \left(\prod_{f \in \mathcal{F}} \exp\left[\theta_f \cdot f\right]\left[X_{[d]} = x_{[d]}\right]\right) \\ &> 0\,.\end{aligned}$$

Condition (6.1) follows from this and the Hybrid Logic Network is well-defined. ∎

6.3.1 Tensor Network Representation

We can employ the formula decompositions to represent both probabilistic facts of the MLN and hard facts (seen as the limit of large weights).

Theorem 6.21 *For any hybrid logic network we have*

$$\mathbb{P}^{(\mathcal{F},\theta,\mathcal{KB})}[X_{[d]}] = \left\langle \{\rho^f\left[Y_f, X_{[d]}\right] : f \in \mathcal{F} \cup \mathcal{KB}\} \cup \{e_1\left[Y_f\right] : f \in \mathcal{KB}\} \cup \{W^f[Y_f] : f \in \mathcal{F}\} \right\rangle \left[X_{[d]} | \varnothing\right]\,.$$

Proof. By Lem. 3.3. ∎

While the statistics computing cores in the relational encoding are shared to compute the soft and the hard logic formulas, their activation cores differ. While probabilistic soft formulas get activation cores (see Theorem 6.3)

$$W^f[Y_f] = \begin{bmatrix} 1 \\ \exp\left[\theta\left[L = l\right]\right] \end{bmatrix}\left[Y_l\right]$$

the hard formulas get activation cores by unit vectors

$$e_1\left[Y_f\right] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\left[Y_l\right]\,.$$

As shown in Sect. 6.2.1, the soft activation cores converge to these hard activation cores in the limit of large parameters, when imposing a local normation. We further notice, that the probabilistic activation cores are trivial tensors if and only if the corresponding parameter coordinate vanishes.

For an example see Figure 6.6.

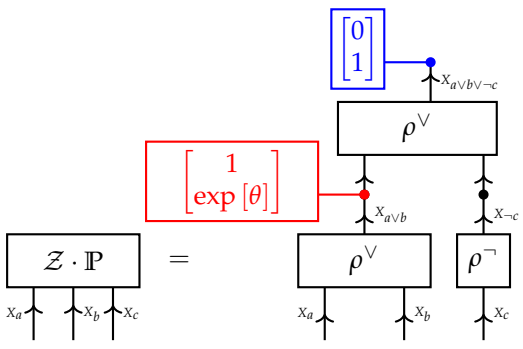


Figure 6.6: Diagram of a formula tensor with activated heads, containing hard constraint cores and probabilistic weight cores .

Remark 6.22 Probability interpretation using the Partition function The tensor networks here represent unnormalized probability distributions. The probability distribution can be normed by the quotient with the naive contraction of the network, the partition function. ♢

6.3.2 Reasoning Properties

Deciding probabilistic entailment (see Def. 4.10) with respect to Hybrid Logic Networks can be reduced to the hard logic parts of the network.

Theorem 6.23 *Let $(\mathcal{F}, \theta, \mathcal{KB})$ define a Hybrid Logic Network. Given a query formula f we have that*

$$\mathbb{P}^{(\mathcal{F},\theta,\mathcal{KB})} \models f$$

if and only if

$$\mathcal{KB} \models f\,.$$

Proof. This follows from Theorem 4.13 on the representation of Hybrid Logic Networks as Markov Networks in Theorem 6.21. ■

Formulas in $\mathcal{F}$, which are entailed or contradicted by $\mathcal{KB}$ are redundant, as we show next.

Theorem 6.24 *If for a formula f and $\mathcal{KB}$ we have*

$$\mathcal{KB} \models f \quad \textit{or} \quad \mathcal{KB} \models \neg f$$

then for any $(\mathcal{F}, \theta, \mathcal{KB})$

$$\mathbb{P}^{(\mathcal{F}/\{f\},\tilde{\theta},\mathcal{KB})}\left[X_{[d]}\right] = \mathbb{P}^{(\mathcal{F},\theta,\mathcal{KB})}\left[X_{[d]}\right]\,,$$

where $\tilde{\theta}$ denotes the tensor θ, where the coordinate to f is dropped, if $f \in \mathcal{F}$.

Proof. Isolate the factor to the hard formula, which is constant for all situations. ■

A similar statement holds for the hard formulas itself, as shown in Theorem 4.4. However, notice that if $\mathcal{KB}/\{f\} \models \neg f$, then $\mathcal{KB} \cup \{f\}$ is not satisfiable and a hybrid logic network cannot be defined for $\mathcal{KB} \cup \{f\}$ as hard logic formulas.

These results are especially interesting for the efficient implementation of Algorithm 6, which has been introduced in Chapter 4. By Theorem 6.23 only the hard logic parts of a Hybrid Logic Network are required in the ASK operation.

6.3.3 Expressivity

Hybrid Logic Networks extend the expressivity result of Theorem 6.5 to arbitrary probability tensors, dropping the positivity constraints for Markov Logic Networks.

Theorem 6.25 *Let $\mathbb{P}\left[X_{[d]}\right]$ a possibly not positive probability tensor we build a base measure*

$$\nu^{\mathcal{KB}} = \mathbb{I}_{\neq 0}\left(\mathbb{P}\left[X_{[d]}\right]\right)$$

and a parameter tensor

$$\theta\left[L_{[d]} = x_{[d]}\right] = \begin{cases} 0 & \text{if} \quad \mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0 \\ \ln\left[\mathbb{P}\left[X_{[d]} = x_{[d]}\right]\right] & \text{else} \end{cases}.$$

Then the probability tensor is the member of the minterm exponential family with base measure $\mathcal{KB}$ and parameter θ, that is

$$\mathbb{P}\left[(\mathcal{F}_\wedge, \theta, \nu^{\mathcal{KB}})\right]$$

Proof. It suffices to show that

$$\left\langle \nu^{\mathcal{KB}}, \exp\left[\left\langle \gamma^{\mathcal{F}_\wedge}\theta\right\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}\right] = \mathbb{P}\left[X_{[d]}\right].$$

For indices $x_{[d]}$ with $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0$ we have $\nu^{\mathcal{KB}}\left[X_{[d]} = x_{[d]}\right] = 0$ and thus also

$$\left\langle \nu^{\mathcal{KB}}, \exp\left[\left\langle \gamma^{\mathcal{F}_\wedge}\theta\right\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]} = x_{[d]}\right] = 0.$$

For indices $x_{[d]}$ with $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0$ we have $\nu^{\mathcal{KB}}\left[X_{[d]} = x_{[d]}\right] = 1$ and

$$\begin{aligned}\left\langle \nu^{\mathcal{KB}}, \exp\left[\left\langle \gamma^{\mathcal{F}_\wedge}, \theta\right\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]} = x_{[d]}\right] &= \prod_{l_{[d]}} \exp\left[\theta\left[L_{[d]} = l_{[d]}\right] \cdot Z^{\wedge}_{l_{[d]}}\left[X_{[d]} = x_{[d]}\right]\right] \\ &= \exp\left[\theta\left[L_{[d]} = x_{[d]}\right]\right] \\ &= \mathbb{P}\left[X_{[d]} = x_{[d]}\right].\end{aligned}$$

■

6.4 Applications

Hybrid Logic Networks as neuro-symbolic architectures:

- Neural Paradigm here by decompositions of logical formulas into their connectives. In more generality by decompositions of sufficient statistics into composed functions, using Basis Calculus. Deeper nodes as carrying correlations of lower nodes.
- Symbolic Paradigm by interpretability of propositional logics.

Hybrid Logic Networks as trainable Machine Learning models:

- Expressivity: Can represent any positive distribution, as shown by Theorem 6.5, with 2^d formulas.
- Efficiency: Can only handle small subsets of possible formulas, since their possible number is huge. Tensor networks provide means to efficiently represent formulas depending on many variables and reason based on contractions.
- Differentiability: Distributions are differentiable functions of their weights, see Parameter Estimation Chapter. The log-likelihood of data is therefore also differentiable function of their weights and we can exploit first-order methods in their optimization.
- Structure Learning: We need to find differentiable parametrizations of logical formulas respecting a chosen architecture. In Chapter 5 such representations are described based on Selector Tensor Networks.

Differentiability and structure learning will be investigated in more detail in the next chapter.

When understanding atoms as observed variables, and the computed as hidden, Hybrid Logic Networks are deep higher-order boltzmann machines: More generic correlations can be captured by a logical connective, calculated by a relational encoding and activated by an activation core.

Hybrid Logic Networks as bridging soft and hard logics within the formalism of exponential families.

A more general class of problems, which have natural representations by Hard Logic Networks are Constraint Satisfaction Problems (see Chapter 5 in [RN21]). Solving such problems is then equivalent to sampling from the worlds in a logical interpretation, and can be approached by the methods of Chapter 4. Among these classed, we have only discussed the Sudoku game in Example 6.18. Extensions by Hybrid Logic Networks can be interpreted as implementations of preferences among possible solutions by probabilities.

7

Logical Network Inference

In this chapter we investigate the inference properties of Hybrid Logic Networks starting with characterizations of its mean parameter polytopes. We first investigate unconstrained parameter estimated for Markov Logic Networks and Hybrid Logic Networks, which are special cases of the backward maps introduced in Chapter 1. We then motivate structure learning based on sparsity constraints on the parameters on the minterm family and present heuristic strategies leading to efficient structure learning algorithms.

7.1 Mean parameters of Hybrid Logic Networks

The convex polytope of realizable mean parameters (see Def. 1.34) is for a statistic $\mathcal{F}$ of propositional formulas and a base measure ν

$$\mathcal{M}_{\mathcal{F},\nu} = \left\{ \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle [L] \,:\, \mathbb{P} \in \Gamma \right\},$$

where by Γ we denote the set of all probability distributions. By The. 1.35 the convex polytope has a characterization as a convex hull

$$\mathcal{M}_{\mathcal{F},\nu} = \operatorname{conv}\left(\gamma^{\mathcal{F}} \left[X_{[d]} = x_{[d]}, L \right] \,:\, x_{[d]} \in \bigtimes_{k \in [d]} [2],\, \nu \left[X_{[d]} = x_{[d]} \right] = 1 \right). \tag{7.1}$$

The mean parameter polytopes are examples of $0/1$-polytopes [Zie00; Gil07], from which a few obvious properties follow. Since those are convex subsets of the cube $[0,1]^p$, which extreme points are all binary vectors, also each $\gamma^{\mathcal{F}} \left[X_{[d]} = x_{[d]}, L \right]$ (with $\nu \left[X_{[d]} = x_{[d]} \right] = 1$) is an extreme point. Further, if for any $l \in [p]$ we have $\mu [L = l] \in \{0,1\}$, then $\mu [L]$ is in the boundary of the cube and thus also of $\mathcal{M}_{\mathcal{F},\nu}$

To summarize some insights:

- If all mean parameters in $\{0,1\}$ then an extreme points, exactly those are hard logic networks
- If some mean params in $\{0,1\}$, then not in the interior. Back direction not correct: There are interior points where no coordinate in $\{0,1\}$.
- If in the interior after dropping the hard coordinates: Then a hybrid logic network

7.1.1 Extreme points by hard logic networks

We exploit this characterization to show, that the extreme points of $\mathcal{M}_{\mathcal{F},\nu}$ are exactly those realizable by Hard Logic Networks.

Theorem 7.1 *Any parameter $\mu [L] \in \mathcal{M}_{\mathcal{F},\nu}$ is an extreme point of $\mathcal{M}_{\mathcal{F},\nu}$, if and only $\mu [L]$ is*

boolean and the formula

$$\nu^{\mathcal{F},\mu}\left[X_{[d]}\right] := \bigwedge_{l\in[p]} \neg^{(1-\mu[L=l])} f_l\left[X_{[d]}\right]$$

is satisfiable. In that case, μ is the mean parameter of the Hard Logic Network with formulas

$$\mathcal{KB} = \left\{\neg^{(1-\mu[L=l])} f_l : l \in [p]\right\},$$

where we denote by $\neg^0$ the identity connective and by $\neg^1 = \neg$ the logical negation.

Proof. "⇒": Let μ be an extreme point of $\mathcal{M}_{\mathcal{F},\nu}$. Since by (7.1) $\mathcal{M}_{\mathcal{F},\nu}$ is a convex hull of vectors, there exists a $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ such that

$$\mu[L] = \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right].$$

By definition of $\gamma^{\mathcal{F}}$, $\mu[L]$ is a boolean vector and for any $l \in [p]$ we have

$$f_l\left[X_{[d]} = x_{[d]}\right] = \mu[L = l]$$

and thus

$$\neg^{(1-\mu[L=l])} f_l\left[X_{[d]} = x_{[d]}\right] = 1.$$

It follows that $x_{[d]}$ is also a model of $\nu^{\mathcal{F},\mu}$ and $\nu^{\mathcal{F},\mu}$ is satisfiable.

"⇐": To show the converse direction, let $\mu[L]$ be a boolean vector such that $\nu^{\mathcal{F},\mu}$ is satisfiable. Then there exists a model $x_{[d]}$ of $\nu^{\mathcal{F},\mu}$. We have for any $l \in [p]$

$$\gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L = l\right] = f_l\left[X_{[d]} = x_{[d]}\right] = \mu[L = l]$$

and thus $\mu[L] = \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]$. With the characterization (7.1) this establishes in particular $\mu[L] \in \mathcal{M}_{\mathcal{F},\nu}$. Since $\mu[L]$ is boolean and therefore an extreme point of the cube $[0,1]^p$, it is also an extreme point of the subset $\mathcal{M}_{\mathcal{F},\nu} \subset [0,1]^p$. ■

7.1.2 Mean parameters in the interior

By The. 2.13 the interior points are those realizable by a Hybrid Logic Network with statistics $\mathcal{F}$ and base measure ν, as we state in the following Corollary.

Corollary 7.2 *If $\mu[L] \in \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$, or equivalently the statistic is minimal and $\mu[L]$ is reproduceable by a distribution positive with respect to ν, then there is $\theta[L]$ such that $\mathbb{P}^{\mathcal{F},\theta,\nu}$ reproduces $\mu[L]$.*

7.1.3 Mean parameters outside the interior

By The. 2.14 mean parameter vectors outside the interior of $\mathcal{M}_{\mathcal{F},\nu}$ are not realizable by distributions, which are positive with respect to the base measure ν. Instead, we in this section construct refined base measures $\tilde{\nu}$ (refinement in the sense of $\tilde{\nu} \prec \nu$), such that there are distributions, which are positive with respect to $\tilde{\nu}$ and represent such mean parameters.

First of all, we can use the criterion of mean parameter coordinates in $\{0,1\}$ as a sufficient condition for $\mu[L] \notin \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$. In this case, the next theorem provides us with a procedure to refine the base measure in these cases.

Theorem 7.3 (Base measure refinement for mean coordinates in $\{0,1\}$) *To any $\mu[L] \in \mathcal{M}_{\mathcal{F},\nu}$ we build the base measure*

$$\nu^{\mathcal{F},\mu}\left[X_{[d]}\right] := \bigwedge_{l\in[p]\,:\,\mu[L=l]\in\{0,1\}} \neg^{(1-\mu[L=l])} f_l\left[X_{[d]}\right] .$$

Then any probability distribution reproducing $\mu[L]$ has a representation with respect to the base measure

$$\tilde{\nu}\left[X_{[d]}\right] = \left\langle \nu, \nu^{\mathcal{F},\mu}\right\rangle \left[X_{[d]}\right] .$$

Proof. Let $\mathbb{P}$ be a distribution representable with respect to ν and reproducing μ, that is

$$\left\langle \mathbb{P}, \gamma^{\mathcal{F}}\right\rangle [L] = \mu[L] .$$

For any $l \in [p]$, if $\mu[L = l] = 1$ we have

$$\left\langle \mathbb{P}\left[X_{[d]}\right], f_l\left[X_{[d]}\right]\right\rangle [\varnothing] = 1$$

and with The. 4.8 probabilistic entailment $\mathbb{P} \models f_l$ (see Def. 4.10). On the other hand for any $l \in [p]$, if $\mu[L = l] = 0$ we have

$$\left\langle \mathbb{P}\left[X_{[d]}\right], \neg f_l\left[X_{[d]}\right]\right\rangle [\varnothing] = 1$$

and with the same argumentation $\mathbb{P} \models \neg f_l$.

Together it holds that $\mathbb{P} \models \nu^{\mathcal{F},\mu}$ and

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \mathbb{P}\left[X_{[d]}\right], \nu^{\mathcal{F},\mu}\left[X_{[d]}\right]\right\rangle [\,] .$$

Thus, $\mathbb{P}$ is representable with respect to the base measure $\nu^{\mathcal{F},\mu}$ and also with respect to the base measure $\tilde{\nu}$. ∎

By The. 7.3 we can reduce the statistics to the set $\mathcal{F}^{\mu} = \{f_l \in \mathcal{F} : \mu[L = l] \notin \{0,1\}\}$ and continue searching for a reproducing distribution, now for the reduced mean parameter with coordinates not in $\{0,1\}$. The criterion of mean parameter coordinates in $\{0,1\}$ is, however, not a necessary condition for $\mu[L] \notin \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$, see Example 7.5 for minimal representations where mean parameters outside the interior exist with coordinates not in $\{0,1\}$.

Following a different approach, any mean parameter outside the interior of the mean parameter polytope is on a face of the polytope. We can use this insight, to refine base measures by face base measures (see Def. 2.16), which has been shown in more generality in The. 2.15. For hybrid logic network we characterize the face base measures in the next theorem.

Theorem 7.4 (Base measures by formula satisfaction) *Let $\theta[L = l]$ be a normal to a face of $\mathcal{M}$. If and only if the face formula*

$$f_{\mathcal{F},\theta} = \bigwedge_{l\in[p]\,:\,\theta[L=l]\neq 0} \neg^{(1-\mathbb{I}_{>0}(\theta[L=l]))} f_l$$

is satisfiable, then it is the base measure to the face with normal θ. Here $\mathbb{I}_{>0}(z)$ denotes the indicator of $z > 0$.

If the above is not satisfiable, then the base measure is

$$\bigvee_{v[L]\,:\,\langle\theta,v\rangle[\varnothing]\in\max_{\mu}\langle\theta,\mu\rangle[\varnothing]} f_{\mathcal{F},\langle\theta,v\rangle[L]}$$

where v are boolean vectors.

Proof. We show this lemma based on characterizations of

$$\text{argmax}_{x_{[d]}} \left\langle \theta\left[L\right], \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]\right\rangle\left[\varnothing\right] .$$

For the first claim: We have since $\gamma^{\mathcal{F}}$ is boolean that

$$\|\theta\left[L\right]\|_1 \geq \max_{x_{[d]}} \left\langle \theta\left[L\right], \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]\right\rangle\left[\varnothing\right] .$$

If and only if the formula $f_{\mathcal{F},\theta}$ is satisfiable, then we find a model $x_{[d]}$ and the inequalitiy is straight. The maximum is taken at any $x_{[d]}$ if and only if for any l in the support of θ we have $\gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L = l\right] = \mathbb{I}_{>0}\left(\theta\left[L = l\right]\right)$. This condition is equal to $x_{[d]}$ being a model of $\neg^{(1-\mathbb{I}_{>0}(\theta[L=l]))} f_l$ for any l in the support of θ and thus to being a model of $f_{\mathcal{F},\theta}$.

For the second claim: The auxiliary boolean vector v serves as an indicator fo the formulas $\neg^{(1-\mathbb{I}_{>0}(\theta[L=l]))} f_l$ being satisfied. In the worlds, for which $\max_{x_{[d]}} \left\langle \theta\left[L\right], \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]\right\rangle\left[\varnothing\right]$ is attained, we have

$$\langle\theta, v\rangle\left[\varnothing\right] \in \max_{\mu} \langle\theta, \mu\rangle\left[\varnothing\right] .$$

They are the models of $f_{\mathcal{F},\langle\theta,v\rangle[L]}$. Note that then, all formulas f_l with $v[L = l] = 0$ need to be contradicted from $f_{\mathcal{F},\langle\theta,v\rangle[L]}$, since otherwise the maximum would be larger. The mean parameters solving the optimization problem are convex combinations of these extreme points, which correspond with distributions being convex combinations of the one-hot encodings of these models. Such distributions are therefore representable by the base measure being the indicator of these models, which is

$$\bigvee_{v[L]\,:\,\langle\theta,v\rangle[\varnothing]\in\max_{\mu}\langle\theta,\mu\rangle[\varnothing]} f_{\mathcal{F},\langle\theta,v\rangle[L]} .$$

■

7.1.4 Expressivity of Hybrid Logic Networks

Let us recall, that the set $\Lambda^{\mathcal{F},\text{EL}}$ contains all Hybrid Logic Networks, which can be realized as tensor networks with the same structure of computation and activation cores. We now investigate, whether we can reduce the set of probability distributions in the definition of the convex polytope of mean parameters to the set $\Lambda^{\mathcal{F},\text{EL}}$ (see Def. 6.19), that is

$$\mathcal{M}_{\mathcal{F},\mathcal{V}}|_{\Lambda^{\mathcal{F},\text{EL}}} = \left\{\left\langle \mathbb{P}, \gamma^{\mathcal{F}}\right\rangle[L] \,:\, \mathbb{P} \in \Lambda^{\mathcal{F},\text{EL}}\right\} .$$

While $\mathcal{M}_{\mathcal{F},\mathcal{V}}|_{\Lambda^{\mathcal{F},\text{EL}}} \subset \mathcal{M}_{\mathcal{F},\mathcal{V}}$ is obvious, we pose the question, for which $\mathcal{F}$ there is an equivalence. We will refer to the equality of both sets as sufficient expressivity of $\Lambda^{\mathcal{F},\text{EL}}$. In the next example we provide a class of formulas, for which $\Lambda^{\mathcal{F},\text{EL}}$ does not have sufficient expressivity.

Example 7.5 (Insufficient expressivity of $\Lambda^{\mathcal{F},\text{EL}}$ in cases of disjoint models) To provide an example, where the set of hybrid logic networks does not suffice to reproduce all possible mean parameters, consider the formulas

$$f_0 = X_0 \wedge X_1 \quad , \quad f_1 = \neg X_0 \wedge \neg X_1 .$$

The probability distributions on the facet with normal $\theta = [1\ 1]$ are those with support on the models of $f_0 \vee f_1$. The Hybrid Logic Networks can only reproduce those with are supported on the model of f_0 or the model of f_1, but not their convex combinations.

More generally, we can construct similar examples by arbitrary sets of formulas with pairwise disjoint model sets. If they do not sum to $\mathbb{I}$, i.e. there is a world which is not a model to any

formula, the statistic is minimal. The vector $\theta[L] = \mathbb{I}[L]$ is then the normal of the facet with All probabilities supported on the models of the formulas have mean parameters on this facet. $\Diamond$

Before presenting the example class of atomic formulas as a case, where $\Lambda^{\mathcal{F},\mathrm{EL}}$ has sufficient expressivity, let us first proof a generic criterion.

Theorem 7.6 *The set of mean parameters realizable by $\Lambda^{\mathcal{F},\mathrm{EL}}$ coincides with the set of mean parameters realizable by any distribution, if for any boolean vector $\theta[L]$ the formula $f_{\mathcal{F},\theta}$ is satisfiable.*

Proof. If these formulas are satisfiable, the base measures to the facets coincides with those realizable by $\Lambda^{\mathcal{F},\mathrm{EL}}$. ∎

When the assumptions of The. 7.6 are not satisfied, there are mean parameters, which can not be reproduced by a distribution in $\Lambda^{\mathcal{F},\mathrm{EL}}$. In that case, we can flexibilize the distribution, to also represent the base measures used for refinement in The. 7.4. This can be done by adding activation cores with multiple variables, or further computation cores calculating the disjunctions of formulas.

To summarize, for any $\mu \in \mathcal{M}_{\mathcal{F},\nu}$ we have one of the following (see Figure **??**):

- $\mu[L = l] \in \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$: Then reproducible by a Hybrid Logic Network with base measure ν .
- $\mu[L = l] \notin \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$: Then on a facet and reproducible by a Hybrid Logic Network with refined base measure $\tilde{\nu}$. Whether the base measure can be realized by the computation network of $\mathcal{H}$ depends on the satisfiability of the face formula.

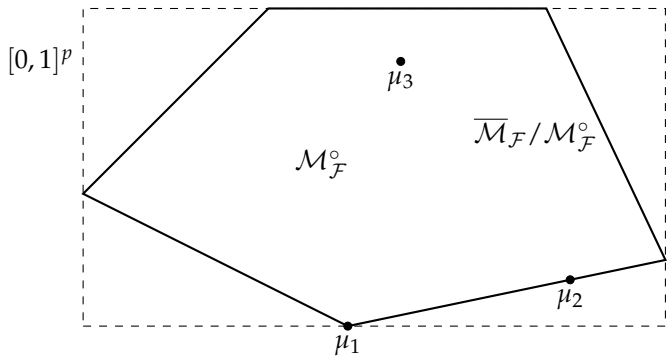


Figure 7.1: Sketch of the convex polytope $\mathcal{M}_{\mathcal{F},\nu}$ as a subset of the p-dimensional cube $[0,1]^p$ (here as a 2-dimensional projection) with example mean parameters μ_1, μ_2, μ_3. The boundary points $\mu_1, \mu_2 \in \overline{\mathcal{M}}_{\mathcal{F}}/\mathcal{M}_{\mathcal{F}}^{\circ}$ are examples of mean parameters, which can be realized by a Hard Logic Networks (respectively Hybrid Logic Network). Any extreme point $\mu_1 \in \mathcal{M}_{\mathcal{F},\nu} \cup \{0,1\}^p$ is realizable by a Hard Logic Network, while a non-extreme boundary point $\mu_2 \in \mathcal{M}_{\mathcal{F},\nu}/\{0,1\}^p$ is realizable by a Hybrid Logic Network. Any interior point $\mu_3 \in \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$ is realizable by a Markov Logic Network.

7.1.5 Case of tree computation networks

In this case, the mean polytope can be embedded into a markov network and characterized by local consistency of the mean parameters of the markov network.

7.1.6 Examples

We can relate our two standard examples of the atomic and the minterm formula sets to well-studied polytopes, namely the d-dimensional hypercube and the standard simplex (see Lecture 0 in [Zie13])

Example 7.7 (Atomic formulas) The assumption of The. 7.6 is satisfied in the case for atomic formulas, where the formulas $f_{\mathcal{F},\theta}$ are the minterms, which are always satisfiable in exactly one situation. The mean polytope in this case is the d-dimensional hypercube,

$$\mathcal{M}_{\mathcal{F}_{[d]},\mathbb{I}} = [0,1]^d$$

which is called a simple polytope, since each vertex is contained in the minimal number of d facets. ◊

Example 7.8 (Minterm formulas) The mean polytope is in this case the $2^d - 1$-dimensional standard simplex. In this case, $\Lambda^{\mathcal{F}_\wedge,\mathrm{EL}}$ contains any distribution and therefore trivially realizes any mean parameter in $\mathcal{M}_{\mathcal{F}_\wedge,\mathbb{I}}$. Set relation to the HLN Expressivity theorem! ◊

7.2 Entropic Motivation of unconstrained Parameter Estimation

Markov Logic Networks are exponential families with statistics by a set $\mathcal{F}$ of propositional formulas. We furthermore allow for propositional formulas as base measures, to also include the discussion of Hybrid Logic Networks. Based on this, we apply the theory of probabilistic inference, developed in Chapter 2 for the generic exponential families.

7.2.1 Maximum Likelihood in Hybrid Logic Networks

The Maximum Likelihood Problem on Markov Logic Networks is the M-projection

$$\mathrm{argmax}_{\theta[L]\in\mathbb{R}^p} \quad \mathbb{H}\left[\mathbb{P}, \mathbb{P}^{(\phi,\theta,\nu)}\right]$$

in the case $\mathbb{P} = \mathbb{P}^D$ for a data map D.

The M-projection coincides, after dropping constant terms in case of non-trivial base measure, with the backward map

$$\mathrm{argmax}_{\theta[L]\in\mathbb{R}^p} \quad \langle \theta[L], \mu[L] \rangle [\varnothing] - A^{(\phi,\nu)}(\theta[L])$$

where

$$\mu[L] = \left\langle \gamma^{\mathcal{F}}, \mathbb{P} \right\rangle [L] \quad \text{and} \quad A^{(\phi,\nu)}(\theta[L]) = \left\langle \exp\left[\left\langle \gamma^{\mathcal{F}}\left[X_{[d]}, L\right], \theta[L] \right\rangle \left[X_{[d]}\right] \right], \nu \right\rangle [\varnothing] .$$

We now extend to Hybrid Logic Networks

$$\mathrm{argmax}_{\tilde{\mathbb{P}}\in\Lambda^{\mathcal{F},\mathrm{EL}}} \quad \mathbb{H}\left[\mathbb{P}, \tilde{\mathbb{P}}\right]$$

Corollary 7.9 *Let* $\mu[L] = \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle [L]$ *and*

$$\tilde{\mathcal{F}} = \{ f_l : \mu[L = l] \in \{0,1\} \} \quad , \quad \nu^{\tilde{\mathcal{F}},\mu}\left[X_{[d]}\right] = \bigwedge_{f_l\in\tilde{\mathcal{F}}} \neg^{(1-\mu[L=l])} f_l\left[X_{[d]}\right] .$$

If $\mu[L]$ *is reproduceable by a positive distribution with respect to* $\nu^{\tilde{\mathcal{F}},\mu}\left[X_{[d]}\right]$*, then the solution of the M-projection of* $\mathbb{P}$ *onto the set of hybrid logic networks is representable by* $\tilde{\mathcal{F}}$ *then coincides with the projection of* $\mathbb{P}$ *onto* $\Gamma^{\mathcal{F}/\tilde{\mathcal{F}},\nu^{\tilde{\mathcal{F}},\mu}}$.

7.2.2 Maximum Entropy in Hybrid Logic Networks

The Maximum Entropy Problem for Markov Logic Networks is

$$\operatorname{argmax}_{\mathbb{P}} \quad \mathbb{H}[\mathbb{P}] \quad \text{subject to} \quad \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle [L] = \mu [L]$$

Corollary 7.10 (of The. 2.26) *Let $\mathbb{P}^D$ be a distribution such that there is a positive distribution $\mathbb{P}$ with $\left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle [L] = \left\langle \mathbb{P}^D, \gamma^{\mathcal{F}} \right\rangle [L]$. Among all positive distributions $\mathbb{P}$ of $\bigtimes_{k \in [d]} [2]$ satisfying this moment matching condition, the Markov Logic Network with formulas $\mathcal{F}$ and weights θ being the solution of the maximum likelihood problem has minimal entropy.*

We notice, that the solution of the maximum entropy problem is thus a Markov Logic Network. This is remarkable, because this motivates our restriction to Markov Logic Networks as those distributions with maximal entropy given satisfaction rates of formulas in $\mathcal{F}$.

When now extend to the situations $\mu [L = l] \in \{0, 1\}$ can appear. It that case the formula is entailed or contradicted by the facts, and dropping should be considered in both cases.

The max entropy - max likelihood duality still holds for hybrid logic networks as we show in the next theorem.

Theorem 7.11 *Given a set of formulas $\tilde{\mathcal{F}}$ and $\tilde{\mu}$, with coordinates $\tilde{\mu}_l \in [0, 1]$ in the closed interval $[0, 1]$. If the corresponding maximum entropy problem is feasible, its solution is a hybrid logic network with*

- $\mathcal{KB} = \{f_l : l \in [p], \mu [L = l] = 1\} \cup \{\neg f_l : l \in [p], \mu [L = l] = 0\}$
- $\mathcal{F} = \{f_l : l \in [p], \mu [L = l] \in (0, 1)\}$
- *θ being the backward map evaluated at the vector μ consisting of the coordinates of $\tilde{\mu}$ not in $\{0, 1\}$*

Proof. Feasible distributions have a density with base measure by $\mathcal{KB}$, we therefore reduce the set of distributions in the argmax to those with density to the base measure. The max entropy is a max entropy problem with respect to that base measure, where we only keep the constraints to the mean parameters different from $\{0, 1\}$ (those are trivially satisfied). The statement then follows from the generic property (see Sec3.1 in [WJ08]). ∎

7.3 Alternating Algorithms to Approximate the Backward Map

Let us now introduce an implementation of the Alternating Moment Matching Algorithm 5 in case of Markov Logic Networks. To solve the moment matching condition at a formula f_l we refine Lem. 2.27 in the following.

Lemma 7.12 *Let there be a base measure ν, a formula selecting map $\mathcal{F} = \{f_l : l \in [p]\}$ and weights θ, and choose $l \in [p]$ such that $f_l \notin \{\mathbb{I}\left[X_{[d]}\right], 0\left[X_{[d]}\right]\}$. The moment matching condition relative to θ, $l \in [p]$ and $\mu_D [L = l] \in (0, 1)$ is then satisfied, if*

$$\theta[L = l] = \ln \left[\frac{\mu_D [L = l]}{(1 - \mu_D [L = l])} \cdot \frac{T\left[X_{f_l} = 0\right]}{T\left[X_{f_l} = 1\right]} \right]$$

where by $T\left[X_{f_l}\right]$ *we denote the contraction*

$$T\left[X_{f_l}\right] = \left\langle \{\rho^{f_l} \,:\, l \in [p]\} \cup \{W^{\tilde{l}} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\} \right\rangle \left[X_{f_l}\right] .$$

Proof. Since $\mathrm{im}\,(f_l) \subset [2]$ we have

$$\mathrm{Id}|_{\mathrm{im}(f_l)} = e_1\left[X_{f_l}\right]$$

and the moment matching condition is by Lem. 2.27 satisfied if

$$\left\langle W^l, e_1, T \right\rangle [\varnothing] = \left\langle W^l, T \right\rangle [\varnothing] \cdot \mu_D [L = l] .$$

This is equal to

$$\exp[\theta [L = l]] \cdot T\left[X_{f_l} = 1\right] = \left(\exp[\theta [L = l]] \cdot T\left[X_{f_l} = 1\right] + T\left[X_{f_l} = 0\right]\right) \cdot \mu_D [L = l] .$$

Rearranging the equations this is equal to

$$T\left[X_{f_l}\right] = \left\langle \{\rho^{f_l}\} \cup \{W^{\tilde{l}} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\} \right\rangle [L] .$$

We notice that the right side is well defined, since we have by assumption $\mu_D [L = l], (1 - \mu_D [L = l]) \neq 0$ and $T\left[X_{f_l} = 0\right], T\left[X_{f_l} = 1\right] \neq 0$ since Markov Logic networks are positive distributions and $f_l \notin \{\mathbb{I}\left[X_{[d]}\right], 0\left[X_{[d]}\right]\}$. ∎

In the case $\mu_D [L = l] \in \{0, 1\}$ the moment matching conditions are not satisfiable for $\theta [L = l] \in \mathbb{R}$. But, we notice, that in the limit $\theta [L = l] \to \infty$ (respectively $-\infty$) we have

$$\mu [L = l] \to 1 \quad \text{(respectively } 0) ,$$

and the moment matching can be satisfied up to arbitrary precision. In Sect. 6.2 we will allow infinite weights and interpret the corresponding factors by logical formulas. As a consequence, we will able to fit graphical models, which we will call hybrid networks on arbitrary satisfiable mean parameters.

The cases $T\left[X_{f_l} = 1\right] = 0$, respectively $T\left[X_{f_l} = 1\right] = 0$ only appear for nontrivial formulas when the distribution is not positive. This is not the case for Markov Logic Networks, but will happen when formulas are added as cores of a Markov Network. This situation will has been investigated in Sect. 6.2.

Since the likelihood is concave (see [KF09]), there are not local maxima the coordinate descent could run into and coordinate descent will give a monotonic improvement of the likelihood.

We suggest an alternating optimization by Algorithm 8, solving the moment matching equation iteratively for all formulas $f \in \mathcal{F}$ and repeat the optimization until a convergence criterion is met. This is an coordinate ascent algorithm, when interpreted the loss $\mathcal{L}_D\left(\mathbb{P}^{(\phi,\theta,\nu)}\right)$ as an objective depending on the vector θ.

In the initialization phase of Algorithm 8, each parameters is initialized relative to a uniform distribution. The algorithm would be finished, if the variables X_f are independent. This would be the case, if the Markov Logic Network consists of atomic formulas only. When they fail to be independent, the adjustment of the weights influence the marginal distribution of other formulas and we need an alternating optimization. This situation corresponds with couplings of the weights by a partition contraction, which does not factorize into terms to each formula.

Solving Equation 7.12 requires inference of a current model by answering a query. This can be a bottleneck and circumvented by approximative inference, see e.g. CAMEL [**ganapathi_constrained_2008**].

Algorithm 8 Alternating Weight Optimization (AWO)

$\mathcal{KB} = \mathbb{I}$, $\tilde{\mathcal{V}} = \varnothing$
for $l \in [p]$ **do**
  if $\mu[L = l] = 1$ **then**

$$\mathcal{KB} \leftarrow \mathcal{KB} \cup \{f_l\}$$

  else if $\mu[L = l] = 0$ **then**

$$\mathcal{KB} \leftarrow \mathcal{KB} \cup \{\neg f_l\}$$

  else
    $\tilde{\mathcal{V}} \leftarrow \tilde{\mathcal{V}} \cup \{l\}$
  end if
end for
for $l \in \tilde{\mathcal{V}}$ **do**
  Compute

$$T\left[X_{f_l}\right] \leftarrow \left\langle \rho^{f_l} \right\rangle \left[X_{f_l}\right]$$

  Set

$$\theta[L = l] \leftarrow \ln\left[\frac{\mu_D[L = l]}{(1 - \mu_D[L = l])} \cdot \frac{T\left[X_{f_l} = 0\right]}{T\left[X_{f_l} = 1\right]}\right]$$

end for
if $\langle \mathcal{KB} \rangle [\varnothing] = 0$ **then**
  raise "Inconsistent Knowledge Base"
end if
while Convergence criterion is not met **do**
  for $l \in \tilde{\mathcal{V}}$ **do**
    Compute

$$T\left[X_{f_l}\right] = \left\langle \{\rho^{f_l} \,:\, l \in [p]\} \cup \{W^{\tilde{l}} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\} \right\rangle \left[X_{f_l}\right]$$

    Set

$$\theta[L = l] = \ln\left[\frac{\mu_D[L = l]}{(1 - \mu_D[L = l])} \cdot \frac{T\left[X_{f_l} = 0\right]}{T\left[X_{f_l} = 1\right]}\right]$$

  end for
end while

Remark 7.13 (Grouping of coordinates with trivial sum) When having a set of coordinates, such that the coordinate functions are binary and sum to the trivial tensor, one can find simultaneous updates to the canonical parameters, such that the partition function is staying invariant. Given a parameter θ^t we compute

$$\mu^t = \left\langle \mathbb{P}^{(\phi, \theta^t)}, \phi \right\rangle [L]$$

and build the update

$$\theta^{t+1} = \theta^t + \ln\left[\mu^D\right] \mu^t .$$

Then, θ^{t+1} satisfies the moment matching equations for all coordinates in the set.

The assumptions are met when taking all features to any hyperedge in a Markov Network seen

as an exponential family. In that case, the update algorithm is refered to as Iterative Proportional Fitting [WJ08]. Further, when activating both f and $\neg f$. $\Diamond$

7.4 Forward and backward mappings in closed form

We recall from Chapter 2, that while forward mappings are always in closed form by contractions, backward mapping in general do not have a closed form representation. Instead, the backward map is in general implicitly characterized by a maximum entropy problem constrained to matching expected sufficient statistics. We investigate in this section specific examples, where closed forms are available for both. In these cases, parameter estimation can thus be solved by application of the inverse on the expected sufficient statistics with respect to the empirical distribution, and iterative algorithms can be avoided.

7.4.1 Maxterms and Minterms

Minterms (respectively maxterms) are ways in propositional logics to get a syntactical formula representation based on a formula to each world which is a model (respectively fails to be a model). We have already studied in Sect. 6.1.4 how to represent any distribution as a MLN of maxterms (respectively minterms), see The. 6.5.

We use the tuple enumeration of the maxterms and minterms by $\bigtimes_{k\in[d]}[2]$ introduced in Sect. 3.3.3. With respect to this enumeration the canonical parameters and mean parameters are tensors in $\bigotimes_{k\in[d]}\mathbb{R}^2$. Since the statistic of the minterm family is the identity, the mean parameters for the minterm family are

$$\mu\left[L_{[d]} = x_{[d]}\right] = \mathbb{P}\left[x_{[d]}\right]$$

and therefore after a relabeling of categorical variables to selection variables $\mu = \mathbb{P}$. For maxterms we have analogously

$$\mu\left[L_{[d]} = x_{[d]}\right] = 1 - \mathbb{P}\left[x_{[d]}\right]$$

and $\mu = \mathbb{I}[] - \mathbb{P}$. We can use these insights to provide a characterization of the forward and backward maps of the minterm and maxterm family.

Theorem 7.14 *Given the Markov Logic Networks to the formula sets*

$$\mathcal{F}_\wedge := \{Z^\wedge_{x_0,\dots,x_{d-1}} : x_0,\dots,x_{d-1} \in \bigtimes_{k\in[d]}[2]\} \quad \textit{and} \quad \mathcal{F}_\vee := \{Z^\vee_{x_0,\dots,x_{d-1}} : x_0,\dots,x_{d-1} \in \bigtimes_{k\in[d]}[2]\}$$

of all minterms, respectively of all mapterms, the forward mapping are

$$F^\wedge(\theta) = \langle \exp[\theta]\rangle\left[X_{[d]}|\varnothing\right] \quad \textit{and} \quad F^\vee(\theta) = \langle \exp[-\theta]\rangle\left[X_{[d]}|\varnothing\right],$$

where in a slight abuse of notation we assigned the variables $X_{[d]}$ to the canonical parameters θ.

Possible choices of the backward mappings are

$$B^\wedge(\mu) = \ln[\mu] \quad \textit{and} \quad B^{\mathcal{F}_\vee}(\mu) = -\ln[\mu].$$

Proof. For the minterms we use that

$$\mathcal{F}_\wedge[X_{[d]}, X_{\mathcal{F}_\wedge}] = \delta\left[X_{[d]}, X_{\mathcal{F}_\vee}\right]$$

and get

$$F^\wedge(\theta) = \left\langle \exp\left[\langle\{\mathcal{F}_\wedge,\theta\}\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right] = \langle\exp[\theta]\rangle\left[X_{[d]}|\varnothing\right].$$

We notice that for any μ in the image of the forward map we have

$$F^{\wedge}(B^{\wedge}(\mu)) = \mu$$

Therefore, $B^{\mathcal{F}_{\wedge}}$ is indeed a backward mapping to the exponential family of minterms.

For the maxterms we use that

$$\mathcal{F}_{\vee}[X_{[d]}, X_{\mathcal{F}_{\vee}}] = \mathbb{I}\left[X_{[d]}, X_{\mathcal{F}_{\vee}}\right] - \delta\left[X_{[d]}, X_{\mathcal{F}_{\vee}}\right]$$

and get

$$\begin{aligned} F^{\vee}(\theta) &= \left\langle \exp\left[\langle\{\mathcal{F}_{\wedge}, \theta\}\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right] \\ &= \left\langle \{\exp\left[\langle\{\mathbb{I}, \theta\}\rangle\left[X_{[d]}\right]\right], \exp\left[-\langle\theta\rangle\left[X_{[d]}\right]\right]\}\right\rangle\left[X_{[d]}|\varnothing\right] \\ &= \langle \exp[-\theta]\rangle\left[X_{[d]}|\varnothing\right] \end{aligned}$$

where we used, that $\exp\left[\langle\{\mathbb{I}, \theta\}\rangle\left[X_{[d]}\right]\right]$ is a multiple of $\mathbb{I}\left[X_{[d]}\right]$ and is thus eliminated in the normation. For any $\mu \in \operatorname{im}(F^{\vee})$ we have

$$F^{\vee}(B^{\vee}(\mu)) = \mu$$

and $B^{\wedge}$ is thus a backward map for the exponential family of maxterms. ∎

Any positive probability distribution can thus be fitted by minterms when we choose $\theta = \ln[\mathbb{P}]$, respectively by maxterms when we choose $\theta = \mathbb{I} - \ln[\mathbb{P}]$. Thus, we have identified a subset of 2^d formulas, which is rich enough to fit any distribution.

7.4.2 Atomic formulas

Let us now derive a closed form backward mapping for the statistic

$$\mathcal{F}_{[d]} := \{X_k : k \in [d]\}.$$

The mean parameters coincide with the queries on the atomic formulas, that is the marginal

$$\mu[L = k] = \mathbb{P}[X_k = 1].$$

Theorem 7.15 *Given a Markov Logic Network with the statistic $\mathcal{F}_{[d]}$ of atomic formulas, the forward mapping from canonical parameters to mean parameters is the coordinatewise sigmoid, that is*

$$F^{[d]}(\theta[L]) = \frac{\exp[\theta[L]]}{\mathbb{I}[L] + \exp[\theta[L]]}$$

where the quotient is performed coordinatewise.

A backward mapping is the coordinatewise logit, that is

$$B^{[d]}(\mu[L]) = \ln\left[\frac{\mu[L]}{\mathbb{I}[L] - \mu[L]}\right].$$

Proof. We have for any $\theta[L] \in \mathbb{R}^d$

$$\mathbb{P}^{(\mathcal{F}_{[d]}, \theta)}\left[X_{[d]}\right] = \bigotimes_{k \in [d]} \langle \exp[\theta[L = k] \cdot X_k]\rangle[X_k|\varnothing].$$

For any $k \in [d]$ it therefore holds, that

$$\begin{aligned} F^{[d]}(\theta\,[L])[L=k] &= \left\langle X_k, \mathbb{P}^{(\mathcal{F}_{[d]},\theta)}\left[X_{[d]}\right]\right\rangle[\varnothing] \\ &= \langle X_k, \langle \exp\left[\theta\,[L=k]\cdot X_k\right]\rangle\,[X_k|\varnothing]\rangle\,[\varnothing] \\ &= \frac{\exp\left[\theta\,[L=k]\right]}{1+\exp\left[\theta\,[L=k]\right]}. \end{aligned}$$

Since the coordinatewise logit is the inverse function of the coordinatewise sigmoid the map

$$B^{[d]}(\mu\,[L])[L=k] = \ln\left[\frac{\mu\,[L=k]}{1-\mu\,[L=k]}\right]$$

satisfies for any μ in the image of the forward map

$$F^{[d]}(B^{[d]}(\mu)) = \mu$$

and is therefore a backward map. ∎

In a selection tensor networks they are represented by a single neuron with identity connective and variable selection to all atoms. We will investigate such examples in more detail in Chapter 12, where atomic formulas Markov Logic Networks are specific cases of monomial decomposition of order 1.

The maximum likelihood estimator of a positive probability distribution by the MLN of atomic formulas is therefore the tensor product of the marginal distributions. The Kullback-Leibler divergence between the distribution and its projection is the mutual information of the atoms, see for example Chapter 8 in [Mac03].

Remark 7.16 (Decomposition into systems of atomic networks) By Independence Decomposition we reduce to a system of atomic MLN. The minterms of such MLNs are the literals. By redundancy (literals sum up to $\mathbb{I}$), it suffices to take only the positive or the negative literal. ♢

7.5 Constrained parameter estimation in the minterm family

We approach structure learning as constrained parameter estimation in the naive exponential family (see Example 1.39), which coincides with the minterm family $\mathcal{F}_\wedge$. The minterm family is defined by the statistic $\phi = \delta\left[X_{[d]}, L_{[d]}\right]$ and has energy tensors coinciding with the canonical parameters.

For the minterm family, we have as mean parameter set the convex hull of one-hot encodings. Each basis vector is an extreme point is an extreme point.

By The. 6.5 all positive distributions are member of the minterm markov logic network family. This expressivity result was generalized to arbitrary distributions, when allowing for formulas as basemeasures by The. 6.25.

Finding the distribution maximizing the likelihood of data would then be the empirical distribution. In this case we would have $\mu_D\left[L_{[d]} = x_{[d]}\right] = \mathbb{P}^D\left[X_{[d]} = x_{[d]}\right]$ and the maximum likelihood distribution is found by the problem

$$\operatorname{argmax}_{\theta\in\bigotimes_{k\in[d]}\mathbb{R}^{m_k}}\left\langle\theta, \mathbb{P}^D\right\rangle[\varnothing] - A^{(\phi,\nu)}(\theta)$$

which is solved at $\theta = \ln\left[\mathbb{P}^D\right]$ with $\mathbb{P}^{(\delta,\ln\left[\mathbb{P}^D\right])} = \mathbb{P}^D$. This follows from $\mathcal{L}_D\left(\mathbb{P}^{(\delta,\theta)}\right) = \mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^D||\mathbb{P}^{(\delta,\theta)}\right]$, which is by Gibbs inequality minimized at $\mathbb{P}^{(\delta,\theta)} = \mathbb{P}^D$, which is the case for $\theta = \ln\left[\mathbb{P}^D\right]$.

We here allow for $\ln[0] = -\infty$, with the convention of $\exp[-\infty] = 0$, to handle datasets where specific worlds are not represented. Better: Use The. 6.25 with basemeasure dropping non appearing data.

To avoid this overfitting situation, we regularize by restricting the parameter to be a set $\Theta \subset \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ and state

$$\operatorname{argmax}_{\theta\in\Theta} \left\langle \theta, \mathbb{P}^D \right\rangle [\varnothing] - A^{(\phi,\nu)}(\theta) \,. \tag{$\mathrm{P}_{\Theta,\mathbb{P}^D}$}$$

Problem **??** has two important types of instantiation, which we discuss in the next sections.

7.5.1 Parameter Estimation

Projecting onto the markov logic family to the statistic $\mathcal{F}$ is the instance of Problem **??** with the hypothesis choice

$$\Theta^{\mathcal{F}} = \operatorname{span}(\{f : f \in \mathcal{F}\}) \,.$$

Then, the problem is the parameter estimation problem studied in Sect. 7.2. To see this, we reparametrize by the coefficient vectors of the elements in the span, which are then understood as the canonical parameter of the respective distribution in the markov logic family to $\mathcal{F}$.

Remark 7.17 (Overparametrization) Taking $\mathcal{F}$ to consist of all propositional formulas, we get a massive overparametrization: The essential statistics maps to a $2^{\left(2^d\right)}$ dimensional real vector space. All possible distributions of the d atomic variables are mapped to an $2^d - 1$ dimensional submanifold, where also the essential statistics maps to.

Thus, to identify probabilistic knowledge bases, we need to drastically restrict the shape of formulas allowed. It is in principle impossible to decide which formulas to be activated, based only on statistics and not on prior assumptions.

When having d atoms, there are 2^d states in the factored system. Since each state can either be a model of a formula or not, there are

$$|\mathcal{F}| = 2^{\left(2^d\right)}$$

formulas. Having, for example, $d = 10$, then $|\mathcal{F}| > 10^{308}$.

One regularization is by allowing only a small number of formulas to be active. This corresponds with regularization with $\ell_0(\theta)$. The problem is then non-convex.

A further regularization strategy is the restriction of the size of the possible formulas to maintain interpretability. Thus, we choose small formula selection networks. ♢

7.5.2 Structure Learning

The problem of structure learning arises, when the set of parameters in Problem **??** is choosen as

$$\Theta^{\mathcal{H}} = \bigcup_{\mathcal{F}\in\mathcal{H}} \operatorname{span}(\mathcal{F}) \,.$$

In this case, the problem in general fails to be convex.

Each formula set $\mathcal{F}$ represents a subspace in the parameters of the minterm family, which is spanned by the propositional formulas $f \in \mathcal{F}$.

7.6 Greedy Structure Learning

It can be impracticle to learn all formulas at once, since the set $\mathcal{H}$ often grows combinatorically, for example when choosing as a powerset of formulas. Further, we need to avoid overfitting and carefully choose a hypothesis. To avoid intractabilities and overfitting, one can choose a greedy approach and learn in addition formulas f when already having learned a set $\mathcal{F}$ of formulas. We

in this section assume a current model $\tilde{\mathbb{P}}$, which is a generic positive distribution not necessarily a Markov Logic Network.

We will use the effective selection tensor network representation of exponentially many formulas described in Chapter 5 and select from them a small subset.

7.6.1 Greedy formula inclusions

Having a current set of formulas $\mathcal{F}$ we want to choose the best $f \in \mathcal{H}$ to extend the set of formulas to $\mathcal{F} \cup \{f\}$ in a way minimizing the cross entropy. Given this, add each step we solve the greedy cross entropy minimization

$$\text{argmin}_{f \in \mathcal{H}} \text{argmin}_{\theta \in \mathbb{R}^{|\mathcal{F}|+1}} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F} \cup \{f\}, \theta, \nu)}\right] .$$

A brute force solution would require parameter estimation for each candidate in $\mathcal{H}$. We provide two more efficient approximative heuristics in the following (see Chapter 20 in [KF09]).

7.6.2 Gain Heuristic

In the gain heuristic, only the parameters of the new formula are optimized and the others left unchanged. This amounts to

$$\text{argmin}_{f \in \mathcal{H}} \left(\min_{\theta[|\mathcal{F}|] \in \mathbb{R}} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F} \cup \{f\}, \theta, \nu)}\right] \right) . \qquad (\mathrm{P}^{\text{gain}}_{D,\mathcal{F},\mathcal{H}})$$

Here we denote by θ the first $|\mathcal{F}|$ coordinates of the M-projection $\tilde{\mathbb{P}}$ of $\mathbb{P}^D$ onto $\mathcal{F}$ and the variable new coordinate at position $\theta\,[|\mathcal{F}|]$.

Lemma 7.18 *The gain heuristic objective is an upper bound on the true greedy objective.*

Proof. Since

$$\text{argmin}_{f \in \mathcal{H}} \left(\text{argmin}_{\theta \in \mathbb{R}^{|\mathcal{F}|+1}} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F} \cup \{f\}, \theta, \nu)}\right] \right) \leq \text{argmin}_{f \in \mathcal{H}} \left(\text{argmin}_{\theta[|\mathcal{F}|] \in \mathbb{R}} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F} \cup \{f\}, \theta, \nu)}\right] \right) .$$

■

Further, this is Problem ($\mathrm{P}_{\Theta,\mathbb{P}^D}$) in the case

$$\Theta = \ln\left[\tilde{\mathbb{P}}\right] + \cup_{f \in \mathcal{F}} \text{span}\,(f) .$$

Let us choose a formula $f \in \mathcal{F}$ and consider Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ in the case

$$\Theta^f = \ln\left[\tilde{\mathbb{P}}\right] + \text{span}\,(f) .$$

This is parameter estimation on the exponential family with the single feature f and the base measure $\tilde{\mathbb{P}}$. Therefore we can apply the theory of Chapter 2 and characterize the solution by the θ satisfying the moment matching condition

$$\left\langle \tilde{\mathbb{P}}, \langle \exp\,[\theta] \rangle \left[X_{[d]} | \varnothing \right] \right\rangle [\varnothing] = \left\langle \mathbb{P}^D, f \right\rangle [\varnothing] .$$

We state the solution of this condition in the next theorem.

Theorem 7.19 *Problem* ($\mathrm{P}^{\mathrm{gain}}_{D,\mathcal{F},\mathcal{H}}$) *is solved at any*

$$\hat{\theta} = \theta_{\hat{f}} \cdot \hat{f}$$

where the formula $\hat{f}$ *is in*

$$\hat{f} \in \mathrm{argmax}_{f \in \mathcal{F}} \mathrm{D}_{\mathrm{KL}} \left[\left\langle \mathbb{P}^D, f \right\rangle [\varnothing] \,||\, \langle \tilde{\mathbb{P}}, f \rangle [\varnothing] \right]$$

and $\theta_{\hat{f}}$ *is the weight of* $\hat{f}$ *in the solution of Problem* $\mathrm{P}_{\Theta,\mathbb{P}^D}$ *with* $\Gamma = \tilde{\mathbb{P}} + \mathrm{span}(f)$. *Here we denote by* $\mathrm{D}_{\mathrm{KL}}[p_1||p_2]$ *the Kullback-Leibler divergence between Bernoulli distributions with parameters* $p_1, p_2 \in [0,1]$, *that is*

$$\mathrm{D}_{\mathrm{KL}}[p_1||p_2] = p_1 \cdot \ln\left[\frac{p_1}{p_2}\right] + (1-p_1) \cdot \ln\left[\frac{(1-p_1)}{(1-p_2)}\right]$$

Proof. For any formula f, the inner minimum of Problem ($\mathrm{P}^{\mathrm{gain}}_{D,\mathcal{F},\mathcal{H}}$) is by Lem. 7.12 taken at

$$\theta_f = \ln\left[\frac{\mu_D}{(1-\mu_D)} \cdot \frac{(1-\tilde{\mu})}{\tilde{\mu}}\right]$$

where

$$\tilde{\mu} = \langle \tilde{\mathbb{P}}, f \rangle [\varnothing]$$

and

$$\mu_D = \left\langle \mathbb{P}^D, f \right\rangle [\varnothing] \,.$$

The difference of the likelihood at the current distribution and the optimum is

$$\mathbb{H}\left[\mathbb{P}^D, \tilde{\mathbb{P}}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F}\cup\{f\},\tilde{\theta}\cup\{\theta_f\},\nu)}\right] = \mu_D \cdot \theta_f - A^{\mathcal{F}\cup\{f\},\nu}\left(\tilde{\theta} \cup \{\theta_f\}\right) \,.$$

We use the representation scheme of Theorem 6.21 and get

$$\begin{aligned}
\left\langle \tilde{\mathbb{P}}, \exp\left[\theta_f \cdot f\right] \right\rangle [\varnothing] &= \left\langle \tilde{\mathbb{P}}, \rho^f\left[X_f\right], W^f[X_f] \right\rangle [\varnothing] \\
&= (1-\tilde{\mu}) + \tilde{\mu} \cdot \exp\left[\theta_f\right] \\
&= (1-\tilde{\mu}) + \frac{\mu_D \cdot (1-\tilde{\mu})}{(1-\mu_D)} \\
&= (1-\tilde{\mu}) \cdot \frac{1}{(1-\mu_D)} \,.
\end{aligned}$$

It follows, that

$$\begin{aligned}
A^{\mathcal{F}\cup\{f\},\nu}\left(\tilde{\theta} \cup \{\theta_f\}\right) &= \ln\left[\left\langle \tilde{\mathbb{P}}, \exp\left[\theta_f \cdot f\right] \right\rangle [\varnothing]\right] \\
&= \ln[1-\tilde{\mu}] - \ln[1-\mu_D] \,.
\end{aligned}$$

We further have

$$\mu_D \cdot \theta_f = \mu_D \cdot \left[\ln\left[\frac{\mu_D}{(1-\mu_D)} \cdot \frac{(1-\tilde{\mu})}{\tilde{\mu}}\right]\right] = \mu_D \ln[\mu_D] - \mu_D \ln[1-\mu_D] + \mu_D \ln[1-\tilde{\mu}] - \mu_D \ln[\tilde{\mu}]$$

and arrive at

$$\mathbb{H}\left[\mathbb{P}^D, \tilde{\mathbb{P}}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(f,\theta_f,\tilde{\mathbb{P}})}\right] = \mu_D \ln[\mu_D] - \mu_D \ln[1-\mu_D] + \mu_D \ln[1-\tilde{\mu}] - \mu_D \ln[\tilde{\mu}] - \ln[1-\tilde{\mu}] - \ln[1-\mu_D]$$

$$= (-\mu_D \ln[\tilde{\mu}] - (1-\mu_D)\ln[1-\tilde{\mu}]) - (-\mu_D \ln[\mu_D] - (1-\mu_D)\ln[1-\mu_D]) \ .$$

By definition, this is the Kullback-Leibler divergence between Bernoulli distributions with parameters μ_D and $\tilde{\mu}$. Since the gain in the likelihood loss when restricting to $\Theta = \text{span}(f)$ is thus given by $\mathrm{D}_{\mathrm{KL}}\left[\langle \mathbb{P}^D, f\rangle[\varnothing] \,||\, \langle \tilde{\mathbb{P}}, f\rangle[\varnothing]\right]$, we have that Problem **??** in the case $\Theta = \bigcup_{f\in\mathcal{F}} \text{span}(f)$ is solved at $\hat{\theta} = \theta_{\hat{f}} \cdot \hat{f}$ where

$$\hat{f} = \mathrm{D}_{\mathrm{KL}}\left[\left\langle \mathbb{P}^D, f\right\rangle[\varnothing] \,||\, \langle \tilde{\mathbb{P}}, f\rangle[\varnothing]\right] \ .$$

■

Thus, we solve the grain heuristic with a coordinatewise transform of the mean parameter tensors to $\mathbb{P}^D$ and $\tilde{\mathbb{P}}$, using the Bernoulli Kullback-Leibler divergence as transform function.

One therefore takes the formula, which marginal distribution in the current model and the targeted distribution are differing at most, measured in the KL divergence.

One optimization method would thus be the computation of the mean parameters to both distribution, building the coordinatewise KL divergence and choosing the maximum. Since we need to evaluate each coordinate, this can be intractable for large sets of formulas.

Further improvement of the model can be achieved by iteratively optimizing the other weights as well, since their corresponding moment matching conditions might be violated after the integration of a new formula. This would require the computation of backward mappings for each candidate formula, for which we only have an alternating approach in general.

7.6.3 Gradient heuristic and the proposal distribution

Advantage: Might avoid formulawise calculus, when sampling from proposal distribution. Brute force solution of gain heuristic require formulawise approach.

We now derive a heuristic of choosing features based on the maximal coordinate of the gradient when differentiating the canonical parameter in the minterm family. To prepare for this, we build the gradient of the loss

$$\mathcal{L}_D\left(\mathbb{P}^{(\delta,\tilde{\theta})}\right) = \left\langle \mathbb{P}^D, \gamma^\delta, \tilde{\theta}\right\rangle[\varnothing] - \ln\left[\left\langle \exp\left[\left\langle \gamma^\delta, \tilde{\theta}\right\rangle\left[X_{[d]}\right]\right]\right\rangle[\varnothing]\right]$$

as

$$\begin{aligned}\nabla_{\tilde{\theta}[L]}\mathcal{L}_D\left(\mathbb{P}^{(\delta,\tilde{\theta})}\right) &= \left\langle \gamma^\delta, \mathbb{P}^D\right\rangle[L] - \left\langle \gamma^\delta, \mathbb{P}^{(\delta,\tilde{\theta})}\right\rangle[L] \\ &= \mathbb{P}^D - \mathbb{P}^{(\delta,\tilde{\theta})} \ .\end{aligned}$$

The gradient shows the typical decomposition into a positive and a negative phase. While the positive phase comes from the data term and prefers directions of large data support, the negative phase originates in the partition function and draws the gradient away from directions already supported by the current model $\mathbb{P}^{(\delta,\tilde{\theta})}$. The negative phase is a regularization, by comparing with what has already been learned. When nothing has been learned so far, we can take the current model to be the uniform distribution, which is the naive exponential family with vanishing canonical parameters.

Given a set $\mathcal{H}$ of features we vary $\tilde{\theta}$ by the function

$$f(\theta) = \tilde{\theta} + \left\langle \theta, \gamma^{\mathcal{H}}\right\rangle\left[X_{[d]}\right] \ .$$

At $\theta = 0$ we have the gradient of the loss of the parametrized formula by

$$\nabla_\theta|_0 \mathcal{L}_D\left(\mathbb{P}^{(\delta,f(\theta),\nu)}\right) = \left\langle \nabla_{f(\theta)}|_{\tilde{\theta}}\mathcal{L}_D\left(\mathbb{P}^{(\delta,f(\theta),\nu)}\right), \nabla_\theta|_0 f(\theta)\right\rangle[\varnothing]$$

$$= \left\langle \mathbb{P}^D, \gamma^\phi \right\rangle [L] - \left\langle \mathbb{P}^{(\delta,\tilde{\theta},\nu)}, \gamma^\phi \right\rangle [L] \ .$$

We want to choose the formula, which is best aligned with the gradient of the log-likelihood, that is using a formula selecting map $\mathcal{H}$

$$\text{argmax}_{l \in [p]} \left\langle \mathbb{P}^D, \mathcal{H} \right\rangle [L = l] - \left\langle \mathbb{P}^{(\delta,\tilde{\theta},\nu)}, \mathcal{H} \right\rangle [L = l] \ . \qquad (\mathrm{P}^{\text{grad}}_{D,\mathcal{F},\mathcal{H}})$$

This method is known as the gradient heuristic or grafting. The objective of Problem ($\mathrm{P}^{\text{grad}}_{D,\mathcal{F},\mathcal{H}}$) has another interpretation by the difference of the mean parameter μ_D and $\tilde{\mu}$ of the projections of the empirical and current distributions on the family to $\mathcal{H}$.

Problem ($\mathrm{P}^{\text{grad}}_{D,\mathcal{F},\mathcal{H}}$) is further equivalent to the formula alignment

$$\text{argmax}_{f \in \mathcal{H}} \left\langle f, \mathbb{P}^D - \tilde{\mathbb{P}} \right\rangle [\varnothing] \ .$$

The objective can be interpreted as the difference of the satisfaction probability of the formula with respect to the empirical distribution and the current distribution.

7.6.4 Iterations

Let us now iterate the search for a best formula at a current model with the optimization of weights after each step. The result is Algorithm 9, which is a greedy algorithm adding iteratively the currently best feature.

Algorithm 9 Greedy Structure Learning

Initialize

$$\tilde{\mathbb{P}} \leftarrow \frac{1}{\prod_{k \in [d]} m_k} \cdot \mathbb{I}\left[X_{[d]}\right] \quad , \quad \mathcal{F} = \varnothing$$

while Stopping criterion is not met **do**

Structure Learning: Compute a (approximative) solution $\hat{f}$ to Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ and add the formula to $\mathcal{F}$, i.e.

$$\mathcal{F} \leftarrow \mathcal{F} \cup \{\hat{f}\}$$

Extend dimension of L by one, by $f_p = \hat{f}$ and $\theta[p] = 0$

Weight Estimation: Estimate the best weights for the added formula and recalibrate the weights of the previous formulas, by calling Algorithm 8.

$$\tilde{\mathbb{P}} \leftarrow \mathbb{P}^{\mathcal{F},\theta}$$

end while

When having used the same learning architecture multiple times, the energy of the corresponding formulas are all representable by a formula selecting architecture. Their energy term is therefore a contraction of the selecting tensor with a parameter tensor θ in a basis CP decomposition with rank by the number of learned formulas. When mutiple selection architectures have been used, the energy is a sum of such contractions. Let us note, that this representation is useful after learning, when performing energy-based inference algorithms on the result. During learning, one needs to instantiate the proposal distribution, which requires instantiation of the probability tensor. However, one could alternate data energy-based and use this as a particle-based proxy for the probability tensor.

Remark 7.20 (Sparsification by Thresholding) To maintain a small set of active formulas, one could combine greedy learning approaches with thresholding on the coordinates of θ. This is a

standard procedure in Iterative Hard Thresholding algorithms of Compressed Sensing, but note that here we do not have a linear in θ objective. $\lozenge$

7.7 Proposal distribution

Let us now understand the likelihood gradient as the energy tensor of a probability distribution, which we call the proposal distribution.

Definition 7.21 (Proposal Distribution) Let there be a base distribution $\tilde{\mathbb{P}}$, a targeted distribution $\mathbb{P}^D$ and a formula selecting map $\mathcal{H}[X_{[d]}, L]$. The proposal distribution at inverse temperature $\beta > 0$ is the distribution of L defined by

$$\left\langle \exp\left[\left\langle \beta \cdot (\mathbb{P}^D - \tilde{\mathbb{P}}), \mathcal{H} \right\rangle [L]\right] \right\rangle [L|\varnothing] \ .$$

The proposal distribution is the member of the exponential family with statistics $\mathcal{H}$ and parameter $\beta \cdot (\mathbb{P}^D - \tilde{\mathbb{P}})$.

The proposal distribution is in the exponential family with sufficient statistic by the formula selecting map $\mathcal{H}$, namely the member with the canonical parameters $\theta = \mathbb{P}^D - \tilde{\mathbb{P}}$. Of further interest are tempered proposal distributions, which are in the same exponential family with canonical parameters $\beta \cdot (\mathbb{P}^D - \tilde{\mathbb{P}})$ where $\beta > 0$ is the inverse temperature parameter.

As Markov Logic Networks, the proposal distributions are in exponential families with the sufficient statistic defined in terms of formula selecting maps. While Markov Logic Networks contract the maps on the selection variables L, the proposal distributions contract them along the categorical variables X to define energy tensors.

The grafting Problem ($\mathrm{P}^{\mathrm{grad}}_{D,\mathcal{F},\mathcal{H}}$) is the search for the mode of the proposal distribution. To solve grafting, we thus need to answer a MAP query, for which we can apply the methods introduced in Chapter 2, such as Gibbs Sampling or Mean Field Approximations in combination with annealing.

7.7.1 Mean parameter polytope

The mean parameter polytope of the proposal distribution with statistic $\mathcal{H}^T$ is the convex hull of the formulas in $\mathcal{F}$, that is

$$\mathcal{M}_{\mathcal{H}^T} = \mathrm{conv}\left(\gamma^{\mathcal{H}^T} L = l, X_{[d]} \ : \ l \in [p]\right) = \mathrm{conv}\left(f\left[X_{[d]}\right] \ : \ f \in \mathcal{H}\right)$$

As it was the case for Markov Logic Networks, the mean parameter polytopes are instances of a 0/1-polytopes [Zie00; Gil07].

The extreme points are the formulas selectable by the formula selecting map $\mathcal{H}$.

7.8 Discussion

Remark 7.22 (Bayesian approach) We only treated the estimation of a single resulting distribution by the data, while in a Bayesian approach one typically considers an uncertainty over possible distributions. When treating θ as a random tensor, which prior distribution is given and posteriori distribution wanted, we have a more involved Bayesian approach. When having a prior $\mathbb{P}[\mathcal{F}, \theta]$ over the Markov Logic Networks we alternatively want to find the parameters $\mathcal{F}, \theta$ solving the maximum a posteriori problem

$$\mathrm{argmax}_{\mathcal{F},\theta} \mathbb{P}^{\mathcal{F},\theta}[\{D(j)\}_{j\in[m]}] \cdot \mathbb{P}[\mathcal{F}, \theta] \ .$$

$\lozenge$

The polytopes of mean parameters to hybrid logic networks and proposal distributions are an interesting connection between the fields of combinatorical optimization and the study of expressivity of tensor networks. This is of special interest, when the computation cores of a hybrid logic network are minimally connected, the mean parameters are captured by local consistencies. Similar investigations have been made in the field of tensor networks, where minimal connected tensor networks are refered to by Hierarchical Tucker formats (HT). Minimal connection is exploited in the tensor network community to show numerical properties of the format, such as closedness and existence of best approximators.

8

Probabilistic Guarantees

When drawing data independently from a random distribution, we are limited by random effects. We in this chapter derive guarantees, that the learning methods introduced in Chapter 2 and Chapter 7 are robust against such effects.

8.1 Fluctuations of random data

A random tensor is a random element of a tensor space $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$, drawn from a probability distribution on $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$. In contrast to the discrete distributions investigated previously in this work, the random tensors are in most generality continuous distributions.

8.1.1 Fluctuation of the empirical distribution

When drawing random states $D(j) \in \bigtimes_{k\in[d]}[m_k]$ by a distribution $\mathbb{P}^*$, we use the one-hot encoding to forward each random state to the random tensor

$$e_{D(j)}\left[X_{[d]}\right] .$$

The expectation of this random tensor is

$$\mathbb{E}\left[e_{D(j)}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \mathbb{P}^*\left[X_{[d]} = x_{[d]}\right] e_{x_{[d]}}\left[X_{[d]}\right] = \mathbb{P}^*\left[X_{[d]}\right] .$$

The empirical distribution is then the average of independent random one-hot encodings, namely the random tensor

$$\mathbb{P}^D = \frac{1}{m}\sum_{j\in[m]} e_{D(j)}\left[X_{[d]}\right] .$$

To avoid confusion let us strengthen, that in this chapter we interpret $\mathbb{P}^D$ as a random tensor taking values in $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$, whereas each supported value of $\mathbb{P}^D$ is an empirical distribution taking values in $\bigtimes_{k\in[d]}[m_k]$. The forwarding of $\bigtimes_{k\in[d]}[m_k]$ under the one-hot encoding is a multinomial random variable, see The. 8.10.

When the marginal of each datapoint is $\mathbb{P}^*$, the expectation of the empirical distribution is

$$\mathbb{E}\left[\mathbb{P}^D\right] = \frac{1}{m}\sum_{j\in[m]} \mathbb{E}\left[e_{D(j)}\right] = \mathbb{P}^* .$$

From the law of large numbers it follows, that in the limit of $m \to \infty$ at any coordinate $x \in \bigtimes_{k\in[d]}[m_k]$ almost everywhere

$$\mathbb{P}^D\left[X_{[d]} = x_{[d]}\right] \to \mathbb{E}\left[\mathbb{P}^D\left[X_{[d]} = x_{[d]}\right]\right] = \mathbb{P}^*\left[X_{[d]} = x_{[d]}\right] .$$

At finite m the empirical distribution differs from the by the difference

$$\mathbb{P}^D - \mathbb{P}^*$$

which we call a fluctuation tensor.

8.1.2 Mean parameter of the empirical distribution

We now investigate the empirical mean parameter

$$\mu_D[L] = \left\langle \gamma^\phi \left[X_{[d]}, L \right], \mathbb{P}^D \left[X_{[d]} \right] \right\rangle [L] \ .$$

Each coordinate of μ_D is decomposed as

$$\mu_D[L = l] = \frac{1}{m} \sum_{j \in [m]} \phi_l [D(j)]$$

and thus stores the empirical average of the feature ϕ_l on the dataset $\{D(j)\}_{j \in [m]}$.

Since the mean parameter depends linearly on the corresponding distribution, we can show the following correspondence between the empirical and the expected mean parameter.

Theorem 8.1 *When drawing data independently from $\mathbb{P}^*$, we have $\mathbb{E}[\mu_D[L]] = \mu^*[L]$, where we call*

$$\mu^*[L] = \left\langle \gamma^\phi \left[X_{[d]}, L \right], \mathbb{P}^D \left[X_{[d]} \right] \right\rangle [L]$$

the expected mean parameter.

Proof. Since the expectation commutes with linear functions. ∎

For each $l \in [p]$ the law of large numbers guarantees that $\mu^*[L = l]$ converges almost surely against $\mu^*[L = l]$ when $m \to \infty$. To utilize these we need to approach the following issues:

- We need non-asymptotic convergence bounds, since one has access to finite data when learning
- The convergence has to happen uniformly for all $l \in [p]$
- Guarantees on the result of an estimated model are more accessible when provided for quantities like the canonical parameter and KL-divergences of the learning result. Those, however, depend nonlinearly on $\mu_D[L]$ and therefore require further investigation.

8.1.3 Noise tensor and its width

Motivated by The. 8.1, we build our derivation of probabilistic guarantees on non-asymptotic and uniform convergence bounds for $\mu_D[L]$. Let us first define the fluctuations of the empirical mean parameter, when drawing the data independently from a random distribution, as the noise tensor.

Definition 8.2 Given a statistic ϕ, $m \in \mathbb{N}$ and a distribution $\mathbb{P}^*$, we call

$$\eta^{\phi, \mathbb{P}^*, D} = \left\langle (\mathbb{P}^D - \mathbb{P}^*), \gamma^\phi \right\rangle [L]$$

the *noise tensor*, where D is a collection of m independent samples of $\mathbb{P}^*$.

The fluctuation of the empirical distribution around the generating distribution corresponds in this notation with the minterm exponential family, taking the identity as statistics. Besides this, fluctuation tensors appears in Markov Logic Networks as fluctuations of random mean parameters and in proposal distributions as fluctuation of random energy tensor. We will discuss these examples in the following sections.

We notice, that the fluctuation tensor $\eta^{\phi,\mathbb{P}^*,D}$ is the centered mean parameter to the empirical distribution, that is

$$\mu_D - \mathbb{E}\left[\mu_D\right] = \left\langle \gamma^{\phi}, \mathbb{P}^D - \mathbb{P}^* \right\rangle [L] \ .$$

In the following we will use the supremum of contractions with random tensors in the derivation of success guarantees for learning problems. Such quantities are called widths.

Definition 8.3 Given a set $\Gamma \subset \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ and $\eta^{\phi,\mathbb{P}^*,D}$ a random tensor taking values in $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we define the width as the random variable

$$\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right) = \sup_{\theta\in\Gamma} \left| \left\langle \theta, \eta^{\phi,\mathbb{P}^*,D} \right\rangle [\varnothing] \right| \ .$$

Bounds on the widths are also called uniform concentration bounds [Goe21] and generic probabilistic bounds will be provided in Sect. 8.4.

8.2 Error bounds based on the noise width

We now derive error bounds for parameter estimation and structure learning, as introduced in Chapter 7. When combined with probabilistic bounds on the noise width, they are probabilistic success guarantees.

8.2.1 Parameter Estimation

We in this section always assume, that $\mathbb{P}^D$ is representable by the base measure ν of the respective exponential families.

Parameter Estimation is the M-projection of the empirical distribution onto an exponential family. In Chapter 2 we have characterized those by the backward map acting on the mean parameter. Thus, while we are interested in the expected canonical parameter

$$\theta_*\left[L\right] = B^{(\phi,\nu)}(\mu^*[L])$$

we get an estimation by the empirical canonical parameter

$$\theta_D\left[L\right] = B^{(\phi,\nu)}(\mu_D\left[L\right]) \, .$$

Unfortunately, since the backward map is not linear, we in general do not have that $\mathbb{E}\left[B^{(\phi,\nu)}(\mu_D)\right]$ coincides with $B^{(\phi,\nu)}(\mu^*)$. To build intuition on the concentration we recall the expression of the backward map as

$$B^{(\phi,\nu)}(\mu) = \mathrm{argmax}_\theta - \mathbb{H}\left[\mathbb{P}^\mu, \mathbb{P}^{(\phi,\theta,\nu)}\right]$$

where $\mathbb{P}^\mu$ is any distribution reproducing the mean parameter. We want to compare the solutions $B^{(\phi,\nu)}(\mu_D)$ and $B^{(\phi,\nu)}(\mu^*)$, in which case $\mathbb{P}^\mu$ can be chosen as $\mathbb{P}^D$ and $\mathbb{P}^*$. It is common to call the objectives $\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta,\nu)}\right]$ and $\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right]$ empirical and expected risk [SB14] Since the

empirical risk has a linear dependence on μ_D, we have at each θ

$$\begin{aligned}\mathbb{E}\left[\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta,\nu)}\right]\right] &= \mathbb{E}\left[\langle \mu_D, \theta\rangle\left[\varnothing\right] - A^{(\phi,\nu)}(\theta)\right] \\ &= \langle \mathbb{E}\left[\mu_D\right], \theta\rangle\left[\varnothing\right] - A^{(\phi,\nu)}(\theta) \\ &= \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right]\end{aligned}$$

By the law of large numbers, in the limit $m \to \infty$ we thus have at each θ a convergence of the empirical risk to the expected risk. However, since the backward map is defined by the minima of these risks, we need a uniform and non-asymptotical concentration guarantee to get more useful bounds. To this end, we now consider constrained parameter estimation and relate the supremum on the differences between expected and empirical risks with the width of the noise tensor.

Lemma 8.4 *For any Γ and D we have*

$$\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right) = \sup_{\theta\in\Gamma}\left|\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right]\right| .$$

Proof. For any $\theta \in \Gamma$ and by $\mathbb{P}^\mu$ realizable mean parameter μ we have

$$\mathbb{H}\left[\mathbb{P}^\mu, \mathbb{P}^{(\phi,\theta,\nu)}\right] = -\langle \mu, \theta\rangle\left[\varnothing\right] + A^{(\phi,\nu)}(\theta) .$$

It follows that

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right] = -\langle (\mu_D - \mu^*), \theta\rangle\left[\varnothing\right]$$

and the claim follows from comparison with Def. 8.2 and Def. 8.3. ∎

As a direct consequence, we have at any $\theta \in \Gamma$

$$\left|\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right]\right| \leq \omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right) .$$

Thus, the absolute difference of the expected risk and the empirical risk is bounded by the width of the noise tensor. This is especially useful for the solution μ_D of the empirical risk minimization, where we can state

$$\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta_D,\nu)}\right] \leq \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta_D,\nu)}\right] + \omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right) .$$

At the solution of a empirical risk minimization problem over Γ, the expected risk exceeds the empirical risk at most by the noise tensor width.

When the generating distribution is in the hypothesis, we can further show the following KL-divergence bound for the estimated distribution.

Theorem 8.5 *Let us assume that for $\theta_* \in \Gamma$ we have $\mathbb{P}^* = \mathbb{P}^{(\phi,\theta_*,\nu)}$. Then for any solution θ_D of the empirical problem we have*

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\phi,\theta_*,\nu)} || \mathbb{P}^{(\phi,\theta_D,\nu)}\right] \leq 2\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right) .$$

Proof. For the solution θ_D of the empirical risk minimization on Γ we have since $\theta_* \in \Gamma$ that

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta_D,\nu)}\right] \leq \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta_*,\nu)}\right] .$$

It follows that

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\phi,\theta_*,\nu)}||\mathbb{P}^{(\phi,\theta_D,\nu)}\right] &\leq \mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\phi,\theta_*,\nu)}||\mathbb{P}^{(\phi,\theta_D,\nu)}\right] + \mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_*,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_D,\nu)}\right] \\ &= \left(\mathbb{H}\left[\mathbb{P}^{(\phi,\theta_*,\nu)},\mathbb{P}^{(\phi,\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_D,\nu)}\right]\right) \\ &\quad - \left(\mathbb{H}\left[\mathbb{P}^{(\phi,\theta_*,\nu)},\mathbb{P}^{(\phi,\theta_*,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_*,\nu)}\right]\right),\end{aligned}$$

where we expanded the KL-divergence as a difference of cross entropies. We apply Lem. 8.4 to estimate the terms in brackets and get

$$\left(\mathbb{H}\left[\mathbb{P}^{(\phi,\theta_*,\nu)},\mathbb{P}^{(\phi,\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_D,\nu)}\right]\right) - \left(\mathbb{H}\left[\mathbb{P}^{(\phi,\theta_*,\nu)},\mathbb{P}^{(\phi,\theta_*,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_*,\nu)}\right]\right) \leq 2\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right).$$

Combined with the above inequality we arrive at

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\phi,\theta_*,\nu)}||\mathbb{P}^{(\phi,\theta_D,\nu)}\right] \leq 2\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right)\blacksquare$$

One technical issue arises from the fact, that when we allow for $\Gamma = \mathbb{R}^p$, then $\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right)$ vanishes or is infinity. To apply the result on the unconstrained parameter estimation, we therefore need to argue on bounded sets for the canonical parameter. When restricting to the sphere $\mathcal{S} \subset \mathbb{R}^p$ we have

$$\|\mu_D - \mu^*\|_2 = \omega_{\mathcal{S}}\left(\eta^{\mathcal{F},\mathbb{P}^*,D}\right),$$

We apply this insight to state the following guarantee for unconstrained parameter estimation.

Theorem 8.6 *Let* θ

$$\left|\mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^{*},\mathbb{P}^{(\phi,\theta_D,\nu)}\right]\right| \leq \omega_{\mathcal{S}}\left(\eta^{\mathcal{F},\mathbb{P}^*,D}\right)\cdot\|\theta_D\|_2.$$

Proof. As in the proof of Lem. 8.4 we use that

$$\mathbb{H}\left[\mathbb{P}^{D},\mathbb{P}^{(\phi,\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^{*},\mathbb{P}^{(\phi,\theta_D,\nu)}\right] = \langle\mu_D - \mu^*,\theta_D\rangle\left[\varnothing\right].$$

By Cauchy-Schwartz we further have

$$|\langle\mu_D - \mu^*,\theta_D\rangle\left[\varnothing\right]| \leq \|\mu_D - \mu^*\|_2\cdot\|\theta_D\|_2.$$

Using that $\|\mu_D - \mu^*\|_2 = \omega_{\mathcal{S}}\left(\eta^{\mathcal{F},\mathbb{P}^*,D}\right)$ we arrive at the claim. ∎

8.2.2 Structure Learning

In the gradient heuristic of structure learning, one selects the statistic to the maximal coordinate of the energy tensor of the proposal distribution. This tensor coincides with the mean parameter of a markov logic network and has thus a fluctuation by the noise tensor. We now use these insights to show a guarantee, that the formula chosen by grafting with respect to the empirical proposal distribution coincides with the formula chosen with respect to the expected proposal distribution. To this end, we need to define the max gap, which is the difference between the maximal coordinate of a tensor to the second maximal coordinate.

Definition 8.7 The max gap of a tensor $T\left[X_{[d]}\right]$ is the quantity

$$\Delta(T) = \left(\max_{x_{[d]}} T\left[X_{[d]} = x_{[d]}\right]\right) - \left(\max_{x_{[d]} \notin \operatorname{argmax}_{x_{[d]}} T\left[X_{[d]} = x_{[d]}\right]} T\left[X_{[d]} = x_{[d]}\right]\right).$$

When comparing the gap with the noise width, we get the following guarantee.

Theorem 8.8 *Whenever*

$$\Delta(\mu^*) > 2 \cdot \omega_{\{e_{x_{[d]}} : x_{[d]} \in \times_{k \in [d]}[m_k]\}}\left(\eta^{\phi, \mathbb{P}^*, D}\right),$$

then any mode $x_{[d]}$ of the empirical proposal distribution is a mode of the expected proposal distribution.

Proof. Let us assume that for a mode $l^D \in \operatorname{argmax}_{l \in [p]} \mu_D[L = l]$ of the empirical mean parameter we have

$$l^D \notin \operatorname{argmax}_{l \in [p]} \mu^*[L = l].$$

For a mode $l^* \in \operatorname{argmax}_{l \in [p]} \mu^*[L = l]$ of the expected mean parameter we then have

$$\mu^*[L = l^D] \leq \mu^*[L = l^*] - \Delta(\mu^*)$$

and

$$\mu_D\left[L = l^D\right] \geq \mu_D[L = l^*].$$

Comparing both intequalities we get

$$\left(\mu_D\left[L = l^D\right] - \mu^*[L = l^D]\right) + \left(-\mu_D[L = l^*] + \mu^*[L = l^*]\right) \geq \Delta(\mu^*).$$

Estimating the terms in the bracket by the width of the noise tensor with respect to basis vectors, we get

$$2 \cdot \omega_{\{e_{x_{[d]}} : x_{[d]} \in \times_{k \in [d]}[m_k]\}}\left(\eta^{\phi, \mathbb{P}^*, D}\right) \geq \Delta(\mu^*),$$

which is a contradiction to the assumption. Thus, any mode of the empirical mean parameter is also a model of the expected mean parameter. ∎

8.3 Fluctuations in Logic Networks

In case of logical formulas being statistics, the coordiantes of the mean parameter are satisfaction rates to the formulas.

For Logic Networks we have statistics consistent of boolean statistics f_l, which are logical formulas. In this case the marginal distributions of the coordinates of $\eta^{\phi, \mathbb{P}^*, D}$ are scaled and centered binomials, which we show now.

Theorem 8.9 *For any $\mathcal{F}$ the marginal distribution of the coordinate $\eta^{\mathcal{F}, \mathbb{P}^*, D}[L = l]$ is the scaled*

and centered binomial distribution

$$\frac{1}{m}\left(B\left(m, \mu\left[L=l\right]\right)-\mu\left[L=l\right]\right)$$

with parameters m and $\mu\left[L=l\right]$.

Proof. We notice that when forwarding a random sample $D(j)$ of $\mathbb{P}^*$ is the random tensor

$$e_{D(j)}\left[X_{[d]}\right]$$

and since $\operatorname{im}\left(\phi_l\right) \subset \{0,1\}$ the contraction

$$\left\langle \phi_l, e_{D(j)}\left[X_{[d]}\right]\right\rangle [\varnothing]$$

is a random variable taking values in $\{0,1\}$. The variable therefore follows a Bernoulli distribution with mean parameter

$$\mu\left[L=l\right]=\mathbb{E}\left[\left\langle \phi_l, e_{D(j)}\left[X_{[d]}\right]\right\rangle [\varnothing]\right]=\left\langle \phi_l, \mathbb{P}^*\right\rangle [\varnothing] \qquad \blacksquare$$

The mean parameter of the M-projection of the empirical distribution on the family of Markov Logic Networks with statistic $\mathcal{H}$ is the random tensor

$$\mu_D\left[L\right]=\left\langle \gamma^{\mathcal{F}}, \mathbb{P}^D\right\rangle [L]\ .$$

The expectation of this random tensor is

$$\mathbb{E}\left[\mu_D\right]=\left\langle \gamma^{\mathcal{F}}, \mathbb{E}\left[\mathbb{P}^D\right]\right\rangle [L]=\left\langle \gamma^{\mathcal{F}}, \mathbb{P}^*\right\rangle [L]=\mu^*\,,$$

where we used that the expectation and contraction operation can be commuted due to the multilinearity of contractions.

8.3.1 Energy tensor in proposal distributions

The fluctuation tensor appears as a fluctuation of the energy of the proposal distribution. The expectation of the energy of the proposal distribution is

$$\mathbb{E}\left[E^{\mathcal{H}^T, \mathbb{P}^D-\tilde{\mathbb{P}}}\right]=\mathbb{E}\left[\left\langle \gamma^{\mathcal{H}^T}, \mathbb{P}^D-\tilde{\mathbb{P}}\right\rangle [L]\right]=\left\langle \gamma^{\mathcal{H}^T}, \mathbb{E}\left[\mathbb{P}^D-\tilde{\mathbb{P}}\right]\right\rangle [L]=\left\langle \gamma^{\mathcal{H}^T}, \mathbb{P}^*-\tilde{\mathbb{P}}\right\rangle [L]=\mathbb{E}\left[E^{\mathcal{H}^T, \mathbb{P}^*-\tilde{\mathbb{P}}}\right]\,.$$

The fluctuation of this random tensor is

$$\mathbb{E}\left[E^{\mathcal{H}^T, \mathbb{P}^D-\tilde{\mathbb{P}}}\right]-\mathbb{E}\left[E^{\mathcal{H}^T, \mathbb{P}^*-\tilde{\mathbb{P}}}\right]=\mathbb{E}\left[E^{\mathcal{H}^T, \mathbb{P}^D-\mathbb{P}^*}\right]$$

and coincides with $\eta^{\mathcal{F}, \mathbb{P}^*, D}$.

8.3.2 Minterm Exponential Family

In case of the minterm exponential family, we have $\phi=\delta\left[X_{[d]}, L\right]$ and the noise tensor is

$$\eta^{\delta, \mathbb{P}^*, D}=\mathbb{P}^D-\mathbb{P}^*\,.$$

This noise tensor follows a multinomial distribution as we show next. To this end, we notice that a multinomial distribution can be defined as the average of one-hot encodings of

independently and identically distributed datapoints. When drawing $\{D(j)\}_{j\in[m]}$ independently from $\mathbb{P}^*$ we denote

$$\sum_{j\in[m]} e_{D(j)}\left[X_{[d]}\right] \sim \underline{B}\left(m, \mathbb{P}^*\right) .$$

Theorem 8.10 *The noise tensor $\eta^{\delta,\mathbb{P}^*,D}$ is a by $\frac{1}{m}$ rescaled centered multinomial random tensor with parameters $\mathbb{P}^*$ and m, that is*

$$\eta^{\delta,\mathbb{P}^*,D} \sim \frac{1}{m}\left(\underline{B}\left(m, \mathbb{P}^*\right) - \mathbb{P}^*\right) .$$

Proof. By the above construction we have

$$\mathbb{P}^D - \mathbb{P}^* = \frac{1}{m}\sum_{j\in[m]}\left(e_{D(j)}\left[X_{[d]}\right] - \mathbb{E}\left[e_{D(j)}\left[X_{[d]}\right]\right]\right)$$

We further have

$$\mathbb{E}\left[e_{D(j)}\left[X_{[d]}\right]\right] = \mathbb{P}^*\left[X_{[d]}\right] .$$

■

The noise tensor characterization by multinomial distributions, which holds for minterm statistics, is a more detailed characterization compared to the characterization of its marginals by binomial distribution in The. 8.9, which holds for generic statistics $\mathcal{F}$.

8.3.3 Guarantees for Mode of the Proposal Distribution

Let us now derive probabilistic guarantees, that the mode of the proposal distribution at the empirical and the generating distribution are equal.

Theorem 8.11 *Whenever the energy tensor of the expected proposal distribution has a gap of Δ, then for every $\epsilon > 0$ any mode of the empirical proposal distribution coincides is also a mode of the expected proposal distribution with probability at least $1 - \exp\left[-\frac{1}{\epsilon^2}\right]$, provided that*

$$m > C\frac{(1 + \ln[p])}{\Delta^2}$$

where C is a universal constant.

Proof. To proof the theorem we combine the deterministic guarantee The. 8.8 with the width bound of The. 8.16, which we show in the next section. Given the assumed bound, the sub-gaussian norm of the width is upper bounded by $C_2 \cdot \Delta$, thus for any $\epsilon > 0$ we have

$$\omega_{\{e_l[L]:l\in[p]\}}\left(\eta^{\mathcal{F},\mathbb{P}^*,D}\right) < 2\Delta$$

with probability at least $1 - \exp\left[-\frac{1}{\epsilon^2}\right]$. The claim thus follows with The. 8.8. ■

Example 8.12 (Gap of a MLNs with single formulas) Let there be the MLN of a maxterm f with d variables, and let $\mathcal{F}$ be the maxterm selecting tensor, then

$$\Delta\left(E^{(\mathcal{F},\mathbb{P}^{(\{f\},\theta)}-\langle\mathbb{I}\rangle[X_{[d]}|\varnothing])}\right) = \frac{1}{2^d - 1 + \exp[-\theta]}$$

If $\theta > 0$ we have an exponentially small gap. Thus, for the above Lemma to apply, the width needs to be exponentially in d small.

Let there be the MLN of a minterm f with d variables, then

$$\Delta(E^{(\mathcal{F},\mathbb{P}^{(\{f\},\theta)}-\langle\mathbb{I}\rangle\left[X_{[d]}|\varnothing\right])}) = \frac{1}{1+(2^d-1)\cdot\exp\left[-\theta\right]}$$

For large θ and d, the gap tends to 1. ◊

8.3.4 Guarantees for Unconstrained Parameter Estimation

We here the sphere bounds and combine with The. 8.6.

Theorem 8.13 *For any $\epsilon \in (0,1)$ we have the following with probability at least $1-\epsilon$. Let $\hat{\theta}$ and $\tau > 0$, then*

$$\left|\mathbb{H}\left[\mathbb{P}^*,\mathbb{P}^{(\mathcal{F},\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^D,\mathbb{P}^{(\mathcal{F},\theta_D,\nu)}\right]\right| \leq \tau\cdot\|\theta_D\|_2$$

provided that

$$m \geq \frac{\langle\mu^*\rangle\left[\varnothing\right] - \left\langle(\mu^*)^2\right\rangle\left[\varnothing\right]}{\epsilon\tau^2}.$$

Proof. The claim follows from the deterministic guarantee The. 8.6 with the probabilistic width bound The. 8.17 to be shown in the next section. ■

8.4 Width bounds for the noise tensor

We here provide width bounds on the noise tensors $\eta^{\mathcal{F},\mathbb{P}^*,D}$ to logic networks, which coordinates have marginal distributions by Binomials, as shown in The. 8.9. All bounds hold for arbitrary statistics $\mathcal{F}$ of propositional formulas and number m of data and the appearing constants are universal, that is independent of particular choices of $\mathcal{F}$ and m.

8.4.1 Basis Vectors

We first introduce the sub-Gaussian Norm and show how we can exploit it to state concentration inequalities.

Definition 8.14 (Sub-Gaussian Norm, see Def. 2.5.6 in [Ver18]) The sub-Gaussian norm of a random variable X is defined as

$$\|X\|_{\psi_2} = \inf\left\{C > 0 : \mathbb{E}\left[\exp\left[\frac{X^2}{C^2}\right]\right] \leq 2\right\}.$$

The moment bound used to define the sub-Gaussian norm can then be combined with Markovs inequality to state concentration bounds. Before showing the utility of these norm, let us first connect with the contraction formalism of this work. When X is a random coordinate of $T\left[X_{[d]}\right]$, selected by a probability tensor $\mathbb{P}\left[X_{[d]}\right]$ we have

$$\mathbb{E}\left[\exp\left[\frac{X^2}{C^2}\right]\right] = \left\langle\mathbb{P}\left[X_{[d]}\right],\exp\left[\frac{1}{C^2}\cdot T\left[X_{[d]}\right]\right]\right\rangle\left[\varnothing\right]$$

and thus

$$\|X\|_{\psi_2} = \inf\left\{C > 0 : \left\langle \mathbb{P}\left[X_{[d]}\right], \exp\left[\frac{1}{C^2} \cdot T\left[X_{[d]}\right]\right]\right\rangle [\varnothing] \leq 2\right\}.$$

We now show a sub-Gaussian norm bound on the coordinates of the noise tensor.

Lemma 8.15 *The marginal distribution of any coordinate of $\eta^{\mathcal{F},\mathbb{P}^*,D}[L]$ is sub-Gaussian with*

$$\left\|\eta^{\mathcal{F},\mathbb{P}^*,D}[L = l]\right\|_{\psi_2} \leq C_0 \frac{1}{\sqrt{m}}$$

where $C_0 > 0$ is a universal constant .

Proof. Any centered Bernoulli variable is bounded and therefore sub-Gaussian with

$$\left\|\left\langle f_l\left[X_{[d]}\right], e_{D(j)}\left[X_{[d]}\right]\right\rangle [\varnothing] - \left\langle f_l\left[X_{[d]}\right], \mathbb{P}^*\right\rangle [\varnothing]\right\|_{\psi_2} \leq \frac{1}{\sqrt{\ln[2]}}.$$

Binomial variables are sums of independent Bernoulli variables. We apply the sub-Gaussian norm bound for sums from Proposition 2.6.1 in [Ver18], which states that for a universal constant $C > 0$ we have

$$\left\|\left\langle f_l\left[X_{[d]}\right], \left(\sum_{j\in[m]} e_{D(j)}\left[X_{[d]}\right] - \mathbb{P}^*\right)\right\rangle [\varnothing]\right\|_{\psi_2} \leq \frac{C \cdot \sqrt{m}}{\sqrt{\ln[2]}}.$$

We therefore have

$$\left\|\eta^{\mathcal{F},\mathbb{P}^*,D}[L = l]\right\|_{\psi_2} = \frac{1}{m}\left\|\left\langle f_l\left[X_{[d]}\right], \left(\sum_{j\in[m]} e_{D(j)}\left[X_{[d]}\right] - \mathbb{P}^*\right)\right\rangle [\varnothing]\right\|_{\psi_2} \leq \frac{C}{\sqrt{\ln[2] \cdot m}}.$$

We arrive at the claimed bound with a transform of the universal constant to $C_0 = \frac{C}{\sqrt{\ln[2]}}$. ∎

Based on this norm bound, we now show a bound on the sub-Gaussian norm of the width with respect to basis vectors.

Theorem 8.16 *For the set of basis vectors*

$$\Gamma = \{e_l[L] : l \in [p]\}$$

we have

$$\left\|\omega_\Gamma\left(\eta^{\mathcal{F},\mathbb{P}^*,D}\right)\right\|_{\psi_2} \leq C_1\sqrt{\frac{1+\ln[p]}{m}},$$

where $C_1 > 0$ is a universal constant.

Proof. We first notice, that

$$\omega_\Gamma(\eta) = \max_{l\in[p]}\left|\eta^{\mathcal{F},\mathbb{P}^*,D}[L = l]\right|$$

By a generic bound on the supremum of sub-Gaussian variables (see Exercise 2.5.10 in [Ver18])

we have for a universal constant $C > 0$

$$\left\| \max_{l \in [p]} \left| \eta^{\mathcal{F},\mathbb{P}^*,D} [L = l] \right| \right\|_{\psi_2} \leq C \left(\max_{l \in [p]} \left\| \eta^{\mathcal{F},\mathbb{P}^*,D} [L = l] \right\|_{\psi_2} \right) \sqrt{1 + \ln [p]} .$$

We now apply Lem. 8.15 and get with $C_1 = C \cdot C_0$ that

$$\left\| \omega_\Gamma \left(\eta^{\mathcal{F},\mathbb{P}^*,D} \right) \right\|_{\psi_2} \leq C_1 \sqrt{\frac{1 + \ln [p]}{m}} .$$

■

The bound in The. 8.16 is furthermore sharp, see the construction of an identically scaling lower bound in Exercise 2.5.11 in [Ver18]. Note that the binomials used here tend to normal distributed variables used in the construction therein.

8.4.2 Sphere

For any tensor $\eta [L]$ and the sphere $\mathcal{S} \subset \mathbb{R}^p$ we have

$$\omega_{\mathcal{S}} (\eta [L]) = \| \eta [L] \|_2 .$$

To show probabilistic width bounds with respect to the sphere, we therefore apply in the following Chebyshevs inequality on the norm of random tensors.

Theorem 8.17 *Let $\mu [L]$ be a deterministic vector with coordinates in $[0, 1]$ and $\eta [L]$ a random vector, which coordiantes are for $l \in [p]$ marginally distributed as*

$$\eta [L = l] \sim B (m, \mu [L = l]) .$$

Then we have for any $\epsilon > 0$, $\tau > 0$ and $m \in \mathbb{N}$ with probability at least $1 - \epsilon$

$$\left\| \frac{\eta - \mathbb{E} [\eta]}{m} \right\|_2 \leq \tau$$

provided that

$$m \geq \frac{\langle \mu [L] , (\mathbb{I} [L] - \mu [L]) \rangle [\varnothing]}{\epsilon \cdot \tau^2} .$$

Proof. Since the squared norm of the noise is the sum of squared centered and averaged Binomials, we have

$$\mathbb{E} \left[\| \eta [L] - \mathbb{E} [\eta [L]] \|_2^2 \right] = m \cdot \left(\sum_{l \in [p]} \mu [L = l] (1 - \mu [L = l]) \right)$$

Here we used that the variance of a variable distributed by $B (m, \mu [L = l])$ is $m \cdot \mu [L = l] (1 - \mu [L = l])$.

If follows, that

$$\mathbb{E} \left[\left(\left\| \frac{\eta - \mathbb{E} [\eta]}{m} \right\|_2 \right)^2 \right] = \frac{\langle \mu [L] , (\mathbb{I} [L] - \mu [L]) \rangle [\varnothing]}{m} .$$

Then we apply a Chebyshev Bound to get for any $\tau > 0$

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 > \tau\right] = \mathbb{P}\left[\left(\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2\right)^2 > \tau^2\right] \leq \frac{\langle \mu[L], (\mathbb{I}[L] - \mu[L])\rangle[\varnothing]}{m \cdot \tau^2}$$

For a $\epsilon > 0$ we choose any m with

$$m \geq \frac{\langle \mu[L], (\mathbb{I}[L] - \mu[L])\rangle[\varnothing]}{\tau^2 \epsilon}$$

and get

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 > \tau\right] \leq \epsilon .$$

Thus, we have

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 \leq \tau\right] = 1 - \mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 > \tau\right] \geq 1 - \epsilon .$$

■

For the minterm family where $\mathcal{F} = \delta$ the noise tensor is a rescaled and centered multinomial. In that case, the bound of The. 8.17 can be simplified by

$$\langle \mu[L], (\mathbb{I}[L] - \mu[L])\rangle[\varnothing] = 1 - \left\langle \mu[L]^2 \right\rangle[\varnothing] .$$

8.5 Discussion

We in this chapter only provided probabilistic width bounds for logic networks, that are exponential families with boolean statistics. Similar recovery bounds for parameter estimation and structure learning for more general exponential families would require width bounds in these generic cases. A general approach towards width bounds are chaining techniques on stochastic processes, see [Tal14]. While we showed bounds based on the sub-Gaussian norm, more general sub-exponential bounds could be used, see [Wai19].

We further assumed that our random tensors to be projected are empirical distributions. More general random tensor networks and corresponding width bounds have been developed in [Goe21].

9

First Order Logic

We now extend the tensor representation from to structured representations, whereas we previously focused on factored representation of systems.

We observe that the more expressive first-order logic bears another tensor structure: The representation of each world is a boolean tensor.

9.1 World Tensors

Since first-order logic follows structured representations of a system, a first-order logic world consists in objects and relations between them. To each world there is a world domain $\mathcal{U}$ of objects, which we assume to be finite (this is a restrictive assumption). We exploit the set-encoding formalism discussed in more detail in Chapter 11 and use bijective index interpretation maps

$$I : [r] \to \mathcal{U} .$$

A so-called term variable O takes states $o \in [r]$, which represent objects

$$I(o) \in \mathcal{U} .$$

The relations between objects are described by n-ary predicates g. Given a specific world x_W the truth of relations is represented by boolean tensors

$$g|_{x_W} : \bigtimes_{l \in [n]} [r] \to \{0, 1\} .$$

Given a tuple $o_0, \ldots, o_{n-1} \in \bigtimes_{l \in [n]} [r]$ the boolean

$$g|_{x_W} [O_0 = o_0, \ldots, O_{n-1} = o_{n-1}] \in \{0, 1\}$$

is called a grounding and encodes, whether the relation g is satisfied in the world x_W for the objects $I^{-1}(o_0), \ldots, I^{-1}(o_{n-1})$.

Let us assume, that we have a function-free theory with d predicates, where are predicates all of the same arity n. We then formalize a world in the following based on a selection variable L selecting a specific predicate and term variables $O_{[n]} = O_0, \ldots, O_{n-1}$ representing choices of objects from a given set $\mathcal{U}$.

Definition 9.1 (FOL World) Given a set of objects $\mathcal{U}$ enumerated by an index interpretation function $I : [r] \to \mathcal{U}$ and a finite set $\{g_0, \ldots, g_{d-1}\}$ of n-ary predicates a world is a boolean tensor

$$x_W[L, O_{[n]}] : [d] \times \left(\bigtimes_{l \in [n]} [r] \right) \to [2] .$$

We interpret the world tensor as encoding in the coordinate $x_W[L = k, O_{[n]} = o_{[n]}]$, whether the k-th predicate is satisfied on the object tuple $I^{-1}(o_0), \dots, I^{-1}(o_{n-1})$.

When the assumptions of function-free and constant variable order are not met, we can do the following tricks. Functions are turned to predicates by their relation interpretation. If there are predicates of different arity in the theory, we can trivially extend them to $n$ary predicates by tensor products with the trivial tensor $\mathbb{I}$. This can be done by a tensor product with $e_0\,[O]$, where we add an object a_0 as a placeholder for predicates with smaller arity.

While in first order logics, depending on the chosen semantics, worlds can have infinite sets of objects, we here only treat worlds with finite objects.

9.1.1 Case of Propositional Logics

Before continuing with the one-hot encoding of first-order logic worlds, let us show that the previously discussed formalism of propositional logics (see Chapter 3) is a special case of first-order logics, namely when demanding $n = 0$. Consistent with Def. 9.1 we have a propositional logic world by

$$x_W : [d] \to [2]\,,$$

which we have in Chapter 3 represented by the assignments $x_k = x_W[L = k]$ to the categorical variables X_k.

To represent logical formulas as sets of possible worlds, and distributions of worlds, we applied in Part I one-hot encodings of possible worlds. For the case of propositional logics, this is

$$e_{x_W}\left[X_{[d]}\right] = \bigotimes_{k\in[d]} e_{x_W[L=k]}\left[X_k\right]\,.$$

9.1.2 One-hot encoding of worlds

Let us now generalize the one-hot encodings of propositional logic worlds to worlds of first-order logic. To encode the boolean tensors x_W describing first order logics as basis elements of a tensor space, we take the one-hot encoding

$$e : \bigtimes_{k\in[d]} \bigtimes_{o_0\in[r_0]} \cdots \bigtimes_{o_{n-1}\in[r_{n-1}]} [2] \to \bigotimes_{k\in[d]} \bigotimes_{o_0\in[r_0]} \cdots \bigotimes_{o_{n-1}\in[r_{n-1}]} \mathbb{R}^2$$

defined by

$$e_{x_W}\left[X_{[d]\times[r]^n}\right] = \bigotimes_{k\in[d]} \bigotimes_{o_0\in[r_0]} \cdots \bigotimes_{o_{n-1}\in[r_{n-1}]} e_{x_W[L=k,O_{[n]}=o_{[n]}]}\left[X_{k,o_{[n]}}\right]\,.$$

This is a tensor of order $d \cdot r^n$, in a tensor space of dimension $2^{(d\cdot r^n)}$. Storage of such tensors in naive formats would not be possible. However, the basis CP format discussed in Chapter 12 still provides storage with demand linear in the order $d \cdot r^n$.

Another issue when comparing different first-order logic worlds arises in potentially different world domains. As we have explored, the cardinality of the domain influences the order of the one-hot encoding tensors. To avoid such issues we here enumerate worlds coinciding in their domains. This restriction is called database semantics (see e.g. Section 8.2.8 in [RN21]), where only those worlds are considered, which domains have a one-to-one map to the constant symbols appearing in a respective knowledge base. When restricting to worlds coinciding in their domain, we still have a factored representation of the system, since we can enumerate the possible worlds by a cartesian product. However, the number of categorical variables representing the world is $d \cdot r^n$ and tensor representations, even in sparse formats, are not feasible due to the large order

required. These techniques to restrict to comparable factored representations are often refered to propositionalization of a first-order logic knowledge base.

9.1.3 Probability distributions

Having established the formalism of one-hot encodings also in the case of first-order logic worlds, we can now proceed with the definition of distributions and formulas, analogously to the development in Part I. Probability distributions over worlds coinciding on their domain are then non-negative and normed tensors

$$\mathbb{P}\left[X_{[d]\times[r]^n}\right] \in \bigotimes_{k\in[d],o_{[n]}\in[r]^n} \mathbb{R}^2 .$$

where each coordinate of a world x_W is captured by a boolean random variable $X_{k,o_{[n]}}$, indicating whether the k-th predicate holds on the object tuple indexed by $o_{[n]}$.

We notice, that by definition these probability distributions are distributions of $d \cdot r^n$ Booleans with $2^{(d\cdot r^n)}$ many states. Unfortunately, it is not possible to design encoding spaces of smaller dimension, when our aim is to get any distribution over possible worlds by an element in the encoding space. This is due to the fact, that one-hot encodings provide a basis in the tensor space, as will be shown in Chapter 10. The reason for the large encoding space dimension is therefore rooted in the equal number of possible worlds and not in an overhead in the dimension of the one-hot encoding space. We will later in this chapter investigate methods to handle such high-dimensional distributions in the formalism of exponential families.

9.1.4 Semantics of formulas

Following the development of Chapter 3, we can choose a semantic approach to the definition of formulas, under the assumption of database semantics. Since the semantic of a logical formula is the set of its models, we again have a one-to-one correspondence between logical formulas and the boolean tensors in the one-hot encoding space

$$\bigotimes_{k\in[d],o_{[n]}\in[r]^n} \mathbb{R}^2 .$$

This correspondence between the semantics and boolean tensor is through a subset encoding (see Def. 11.1) of the respective formulas. However, due to the large state dimensions, we will in the following sections choose a syntactical approach to the construction of formulas, which will naturally provide efficient tensor network decompositions.

9.1.5 Two levels of tensor representation

In comparison with propositional logics, first-order logic bears two levels of natural tensor representations. In the first level, each world (see Def. 9.1) has a natural structure by a tensor, since it encodes relations between objects chosen by assignments to term variables. This is different to the worlds of a propositional logic theory, which are represented by a boolean vector instead of a tensor. The second level arises as in propositional logics, by understanding each world as a uncertain state and studying distributions over states, which are understood themself as a tensor (see Def. 1.1). As argued above, the assumption of database semantics is central to exploit the tensor structure of the first level. Under this assumption, representation of an uncertain state, or a collection of possible states, is done in the tensor space

$$\bigotimes_{k\in[d],o_{[n]}\in[r]^n} \mathbb{R}^2$$

where the enumeration of the 2-dimensional axes contains the tensor structure of the first level.

9.2 Formulas in a fixed first-order logic world

Following the argumentation above, we in this section restrict to the exploitation of the first level tensor structure, namely a fixed world represented as a tensor $x_W[L, O_{[n]}]$, see Def. 9.1. We are specifically interested in the tensor network decomposition of first order formulas, which contain in full generality variables and therefore also have a tensor. The evaluation of a first-order formula on a specific world is therefore different to the case in propositional logics, where the evaluation was a boolean in $\{0, 1\}$ indicating whether the world is a model.

9.2.1 Grounding tensors

Given a first-order logic world $x_W[L, O_{[n]}]$, arbitrary formulas are interpreted in terms of the satisfactions of their groundings. We define their semantic first, and then relate their syntactical decomposition to tensor networks, similar to our approach to propositional logics in Chapter 3.

Definition 9.2 (Grounding of a first-order formula given a world) Given a specific world x_W, with an domain $\mathcal{U}$ enumerated by $[r]$, the grounding of a formula q with variables O_q is the tensor

$$q|_{x_W}\left[O_q\right] : \bigtimes_{l \in [O_q]} [r] \to \{0, 1\}\,.$$

Each coordinate represents thereby the boolean, whether the substitution of the variables in the formula is satisfied given a world x_W, that is

$$q|_{x_W}\left[O_q = o_q\right] = \begin{cases} 1 & \text{if the substitution of } q \text{ with the variables } O_q \text{ replaced by the objects } I(o_l) \text{ is satisfied on th} \\ 0 & \text{else} \end{cases}$$

The grounding tensor formalism can be used to define formulas as a map

$$q : \left(\bigotimes_{k \in [d], o_{[n]} \in [r]^n} \mathbb{R}^2 \right) \to \left(\bigotimes_{k \in [d], o_q \in [r]^{|O_q|}} \mathbb{R}^2 \right)$$

where each world x_W is mapped to a grounding tensor

$$q(x_W) = q|_{x_W}\,.$$

This would involve the second level of tensor interpretation, namely

9.2.2 Predicates

The predicates itself are the simplest cases of first-order formulas with term variables. They are stored in the slices to the first axis of x_W and we have

$$g_k|_{x_W} = \left\langle x_W[L, O_{[n]}], e_k\left[L\right] \right\rangle \left[O_{[n]}\right]\,.$$

What is more abstract, we can understand the predicate itself as an object, then take the first-order world as a grounding tensor of a more abstract formula. We will follow this thought in the ternary representation of Knowledge Graphs in Sect. 9.3.2.

9.2.3 Substitution by slicing

Slicing the grounding tensor of a formula a first-order formula amounts to substitution of the respective variable by the constant at the enumeration index.

9.2.4 Formuly synthesis by connectives

In order to have a sound semantic, the grounding of FOL formulas is determined by the syntax of the formula, i.e. a decomposition of the formula into connectives and quantifiers acting on atomic formulas.

Quantifier-free formulas are connectives acting on atomic formulas. We can describe them as in the case of propositional logics in the ρ-formalism. While the atomic formulas where delta tensors copying states, they are more involved here.

Theorem 9.3 *For any connective $\circ$ and formulas q_1 and q_2 we have*

$$(q_1 \circ q_2)|_{x_W}\left[O_{q_1} \cup O_{q_2}\right] = \left\langle \rho^{q_1|_{x_W}}\left[Y_{q_1}, O_{q_1}\right], \rho^{q_2|_{x_W}}\left[Y_{q_2}, O_{q_2}\right], \rho^{\circ}\left[Y_{q_1\circ q_2}, Y_{q_1}, Y_{q_2}\right], e_1\left[Y_{q_1\circ q_2}\right]\right\rangle \left[O_{[n]}\right] .$$

Proof. This directly follows from The. 11.10. ∎

Here, variables can be shared by the connected formulas, therefore the variables in the combined formula are unions of the possible not disjoint variables of the connected formulas.

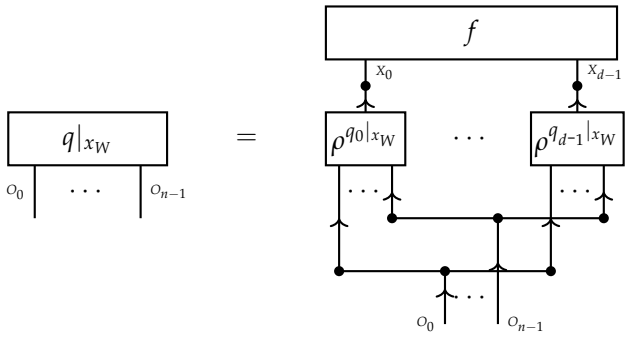
When interpreting the head variables of relational encoded atomic formulas as the atoms of a propositional theory, we find a propositional formula f associated with any decomposable first order logic formula.

Definition 9.4 Given a formula q in first order logic, we say that a propositional formula $f\left[X_{[d]}\right]$ is the propositional equivalent to q given atomic formulas q_k in first order logic, when for any world x_W we have

$$q|_{x_W}\left[O_q\right] = \left\langle \{\rho^{q_k|_{x_W}}\left[X_k, O_{q_k}\right] : k \in [d]\} \cup \{f\left[X_{[d]}\right]\}\right\rangle \left[O_q\right] .$$

We here denote the head variables of the relational encoding to $\rho^{q_k|_{x_W}}$ by X_k to highlight their interpretation as propositional atoms.

We depict the relation of a grounding tensor to a propositional formula as:



9.2.5 Quantifiers

Existential and universal quantifiers appear in generic first order logic and are besides substitutions further means to reduce the number of variables in a formula.

The semantics of existential quantification consists in a formula being true, if at least one state of the quantified variable is true, as we define next.

Definition 9.5 Given a grounding tensor

$$q|_{x_W}[O_0,\ldots,O_{n-1}]$$

the existential and universal quantification with respect to the first variable are the tensors

$$(\exists_{o_0} q)|_{x_W}[O_1,\ldots,O_{n-1}] \quad \text{and} \quad (\forall_{o_0} q)|_{x_W}[O_1,\ldots,O_{n-1}]$$

with coordinates as follows. For an assignment $o_1,\ldots,o$ to the non-quantified variables we have

$$(\exists_{o_0} q)|_{x_W}[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1$$

if and only if there is an assignment $o_0 \in [r_0]$ such that

$$q|_{x_W}[O_0 = o_0, O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1\,.$$

Conversely, we have for the universal quantification that

$$(\forall_{o_0} q)|_{x_W}[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1$$

if and only if for any assignment $o_0 \in [r_0]$ we have

$$q|_{x_W}[O_0 = o_0, O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1\,.$$

Let us now show, that existential and universal quantification are coordinatewise transforms (see Def. 10.5) of contracted grounding tensors. To this end, let us introduce the greater-z indicator $\mathbb{I}_{>z}$, where $z \in \mathbb{R}$, as the function

$$\mathbb{I}_{>:}\mathbb{R} \to \{0,1\} \quad , \quad \mathbb{I}_{>z}(x) = \begin{cases} 1 & \text{if} \quad x > z \\ 0 & \textit{else} \end{cases}\,.$$

Theorem 9.6 *For any formula q with variables $O_{[n]}$ we have*

$$(\exists_{o_0} q)|_{x_W}[O_1,\ldots,O_{n-1}] = \mathbb{I}_{>0}\left(\langle q|_{x_W}\rangle[O_1,\ldots,O_{n-1}]\right)[O_1,\ldots,O_{n-1}]$$

and

$$\left(\forall_{a_{o_0}} q\right)|_{x_W}[O_1,\ldots,O_{n-1}] = \mathbb{I}_{>r-1}\left(\langle q|_{x_W}\rangle[O_1,\ldots,O_{n-1}]\right)[O_1,\ldots,O_{n-1}]$$

Proof. We proof the claimed equalities to arbitrary slices of the remaining variables, which amount to arbitrary substitutions of the formulas. For any indices $o_1 \in [r_1],\ldots,o_{n-1} \in [r_{n-1}]$ we notice, that

$$\begin{aligned}\langle q|_{x_W}\rangle[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] &= \sum_{o_0 \in [r_0]} q|_{x_W}[O_0 = o_0,\ldots,O_{n-1} = o_{n-1}] \\ &= |o_0 \in [r_0] \,:\, q|_{x_W}[O_0 = o_0,\ldots,O_{n-1} = o_{n-1}] = 1|\,.\end{aligned}$$

We can thus understand the contracted grounding tensor as storing in its coordinates the count of the coordinate extensions to the zeroth variable, such that the grounding tensor is satisfied. This is analogous to our interpretation of contracted propositional formulas as world counts. From

this it is obvious, that the existential quantification is satisfied, if the count is different from zero, which is captured by the coordinatewise transform with $\mathbb{I}_{>0}$. We therefore arrive at

$$(\exists_{o_0} q)|_{x_W} [O_1 = o_1, \ldots, O_{n-1} = o_{n-1}] = \mathbb{I}_{>0} \left(\langle q|_{x_W} \rangle [O_1, \ldots, O_{n-1}] \right) [O_1 = o_1, \ldots, O_{n-1} = o_{n-1}] \, .$$

The first claim follows, since the assignment to the non-quantified variables was arbitrary. The universal quantification is satisfied, when all extensions are satisfied, and the count is r. Since r is the maximal count, this is captured by the coordinatewise transform with $\mathbb{I}_{>r-1}$ and we get

$$(\forall_{o_0} q)|_{x_W} [O_1 = o_1, \ldots, O_{n-1} = o_{n-1}] = \mathbb{I}_{>r-1} \left(\langle q|_{x_W} \rangle [O_1, \ldots, O_{n-1}] \right) [O_1 = o_1, \ldots, O_{n-1} = o_{n-1}] \, .$$

With the same argument, the second claim is established. ∎

We can extend this discussion towards more generic counting quantifiers, of which the existential and the universal quantifier are extreme cases. One can define quantifiers by demanding that at least $z \in \mathbb{N}$ compatible groundings are satisfied, and show that they amount to coordinatewise transforms with $\mathbb{I}_{>z}$. What is more, quantifiers demanding that at most $z \in \mathbb{N}$ are satisfied would be representable by transforms with an analogously defined function $\mathbb{I}_{\leq z}$. Such customized quantifiers appear for example in the OWL 2 standard of description logics (see [**rudolph_foundations_2011**] and Sect. 9.3).

As will be discussed in Chapter 11, any coordinatewise transform can be performed by a contraction of a relational encoding of the tensor with a head vector prepared by the transform function (see The. 11.11). In the case here, a direct implementation would require a dimension of these head variables by r, which can be infeasible when having large object sets.

9.2.6 Storage in basis CP decomposition

In many situations, grounding cores are sparse and representations as single tensor cores comes with a drastic overhead. We often encounter sparse grounding tensors, where the number of non-zero coordinates (to be investigated by basis CP ranks in Chapter 12) satisfies

$$\ell_0 \left(q|_{x_W} \right) << r^{|O_q|} \, .$$

In this case, since most coordinates vanish, the basis CP decomposition (see Sect. 12.1.2) enables a representation of the grounding with significantly lower storage demand, see The. 12.3. This is particularly useful for representing large relational databases, where each object has only a few relations with others, while the majority of possible relations remains unsatisfied. We depict such CP decomposition of a formula grounding in The. 9.1.

Most logical syntaxes exploit ℓ_0-sparsity, explicitly storing only known assertions. The interpretation of unspecified assertions depends on the underlying assumptions. Under the Closed World Assumption, for example, all unspecified assertions are assumed to be false.

9.2.7 Queries

A database is understood as a specific fist order logic world, and are operations on such a single world. Queries are described by a formula p, which are asked against a specific world x_W to retrieve the grounding $p|_{x_W}$. The variables of such formulas are called projection variables. The answer $p|_{x_W}$ of a query is most conveniently represented as a list of solution mappings from the projection variables to objects in the world, such that the query formula is satisfied. Answering a query by solution mappings corresponds with finding the basis CP Decomposition (see Sect. 12.1.2) of $p|_{x_W}$. We can understand these solution mappings as stored in the leg-matrices $V^{q,l}$ (see Figure **??**).

Let us give with the outer join an example of a popular operation to define queries, which efficient execution and storage can be improved based on considerations in the tensor network formalism.

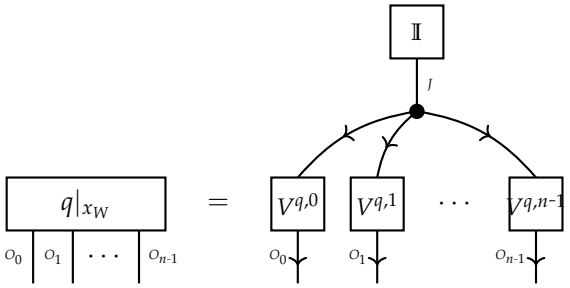


Figure 9.1: Basis CP Decomposition of the grounding of q, following the scheme of The. 12.3. Instead of direct storage of the grounding tensor $q|_{x_W}$, the non-zero coordinates are enumerated by a variable J and the corresponding coordinates stored in leg-matrices $V^{q,l}$.

Definition 9.7 (Outer join) Let there be a world x_W and formulas q_l depending on variables $O_{\mathcal{V}^l}$, which have grounding tensors by

$$q_l|_{x_W}[O_v] \ : \ \bigtimes_{v\in\mathcal{V}^l}[r_v] \to \{0,1\}\,.$$

Then their (outer) JOIN is defined as the grounding of their conjunctions, as

$$\text{JOIN}\left(q_0,\ldots,q_{p-1}\right)|_{x_W}\left[\bigcup_{l\in[p]} O_{\mathcal{V}^l}\right] = \langle q_l|_{x_W}[O_{\mathcal{V}^l}] \ : \ l\in[p]\rangle\left[\bigcup_{l\in[p]} O_{\mathcal{V}^l}\right]\,.$$

We can understand the JOIN of groundings by a factor graph, where each grounding tensor decorates the hyperedge to the node set $\mathcal{V}^l$. The projection variable assignment to each formula combined in a JOIN operation provide a basic tensor network format to store the output of the operation. There are thus situations, in which the solution map storage corresponding with a CP Decomposition comes with unnecessary overheads compared with other formats.

We can also understand the JOIN operation as a coordinatewise transform (see Def. 10.5) with the product as transform function. To make this connection solid, one would need to extend each joined formula trivially to the variables appearing in other formulas.

The efficiency of evaluating the contraction to a JOIN operation might be improved by understanding it as an Constraint Satisfaction Problem (see Chapter 4). When applying efficient Message Passing algorithms such as Knowledge Propagation (see Algorithm 7), the groundings can be sparsified by local constraint propagation operations before turning to more global and more demanding contraction operations. Here the groundings $q_l|_{x_W}$ would be used to initialize Knowledge Cores K^e and sequentially sparsified during the algorithm.

9.3 Representation of Knowledge Graphs

Let us now represent a specific fragment of first-order logic, namely Description Logics which Knowledge Bases are often refered to as Knowledge Graphs. We here use the OWL 2 standard, which encodes the syntax of the description logic $\mathcal{SROIQ}(\mathcal{D})$ **[rudolph_foundations_2011]**.

9.3.1 Representation as unary and binary predicates

Predicates in knowledge graphs are binary (owl:ObjectProperties) and unary (owl:Class). We enumerate the predicates by $[d]$, the objects in the domain $\mathcal{U}$ by $[r]$, and extend the unary predicates to binaries by tensor product with $e_0[O_1]$. A Knowledge Graph on the set $\mathcal{U}$ of

constants (owl:NamedIndividuals) is then the tensor

$$\mathrm{KG}|_{x_W}\,[L, O_0, O_1] : [d] \times [r] \times [r] \to \{0, 1\}\,.$$

9.3.2 Representation as ternary predicate

It has been particulary convenient to represent a Knowledge Graph instead as a grounding of a single ternary predicate RDF. To this end, the predicates g_k and another object rdftype are added to a domain $\mathcal{U}$, by extending the r and the index interpretation function accordingly.

Following our notation we understand a Knowledge Graph as a grounding of the rdf triple relation RDF (being a formula of order 3) on a specific world $\mathrm{KG}|_{x_W}$ with individuals $\mathcal{U}$

We then construct a grounding tensor $\mathrm{RDF}|_{x_W}$ out of the world $\mathrm{KG}|_{x_W}\,[L, O_0, O_1]$ by

$$\mathrm{RDF}|_{x_W} : [r] \times [r] \times [r] \to \{0, 1\}$$

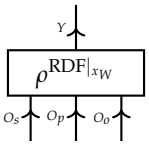
where

$$\mathrm{RDF}|_{x_W}\,\big[O_s = o_s, O_p = o_p, O_o = o_o\big] = \begin{cases} \mathrm{KG}|_{x_W}\,[L = o_s, O_0 = o_o, O_1 = 0] & \text{if} \quad o_p = I^{-1}(\text{rdftype}) \\ \mathrm{KG}|_{x_W}\,\big[L = o_p, O_0 = o_s, O_1 = o_o\big] & \text{if} \quad o_p = I^{-1}(g_k) \quad \text{for some} \quad k \\ 0 & \text{else} \end{cases}.$$

Slicing the tensor $\mathrm{RDF}|_{x_W}$ along the predicate axis retrieves specific information about roles and can be efficiently be performed on these formats. The role rdftype has a specific meaning, since it contains from a DL perspective classifications (memberships of named concepts). Further slicing the tensor along object axis therefore results in membership lists for specific classes (concepts). One can thus regard rdftype as a placeholder for unitary formulas in a space of binary formulas.

Exploiting the ℓ_0-sparsity now leads to a so-called triple store, where $\mathrm{RDF}|_{x_W}$ is stored by a listing of those triples o_s, o_p, o_o such that $\mathrm{RDF}|_{x_W}\,\big[O_s = o_s, O_p = o_p, O_o = o_o\big] = 1$ A recent implementation of a triple store exploiting these intuitions is TENTRIS, see [Big+20]. In this work, such decompositions are generalized into more generic CP formats, see Chapter 12. Approximations of grounding tensors by decompositions leads to embeddings of the individuals such as Tucker, ComplEx and RESCAL (see [Nic+16]).

For our purposes of evaluating logical formulas such as SPARQL queries we use the relational encoding of the groundings, which are depicted by



9.3.3 SPARQL Queries

The SPARQL query language is a syntax to express first-order logic formulas q and intended to be evaluated given a Knowledge Graph. We here consider tensor network representations of the WHERE· block. Given a specific knowledge graph $\mathrm{RDF}|_{x_W}$, the execution of query is the interpretation $q|_{x_W}$, typical represented in a sparse basis CP format where each slice represents a solution mapping.

9.3.3.1 Triple Patterns

Central to SPARQL queries are triple patterns, which we understand as slicings of the tensor $\mathrm{RDF}|_{x_W}$. To each so-called triple pattern we define a corresponding pattern tensor. The triple pattern is then evaluated by contraction of the pattern tensor with $\mathrm{RDF}|_{x_W}$.

Let us now provide examples of such pattern tensors. A unary triple patterns contains a single projection variable, typically related with the subject variable O_s of $\mathrm{RDF}|_{x_W}$. The corresponding pattern tensor is then

$$\psi_{\langle Z,\mathrm{rdftype},g_k\rangle}\left[O_s,O_p,O_o,Z\right]=\delta\left[O_s,Z\right]\otimes e_{I^{-1}(\mathrm{rdftype})}\left[O_p\right]\otimes e_{I^{-1}(g_k)}\left[O_o\right].$$

Binary triple patterns come with two projection variables, typically related with the subject and the object variables O_s and O_o. The pattern tensor to the k-th predicate is then

$$\psi_{\langle Z_0,g_k,Z_1\rangle}\left[O_s,O_p,O_o,Z_0,Z_1\right]=\delta\left[O_s,Z_0\right]\otimes e_{I^{-1}(g_k)}\left[O_p\right]\otimes\delta\left[O_o,Z_1\right].$$

Contraction with these pattern tensor evaluated the specific triple pattern, and outputs in a boolean tensor the indicator, which objects are members of a specific class (for unary patterns) or which pair of objects are related by a specific relation. Again, the output of such contractions is a subset encodings of the set of solutions (see Def. 11.1).

Examples of triple patterns, drawn in Figure 9.2 are

- Unary triple pattern with one variable, representing a formula with a single projection variable. For the example $\langle Z,\mathrm{rdftype},C\rangle$ see Figure 9.2a.

  $$\psi_{\langle Z,\mathrm{rdftype},g_k\rangle}\left[O_s,O_p,O_o,Z\right]=\delta\left[O_s,Z\right]\otimes e_{I^{-1}(\mathrm{rdftype})}\left[O_p\right]\otimes e_{I^{-1}(C)}\left[O_o\right]$$

  If and only if the output slice is e_1, then the corresponding object encoded by the input indices is of class C.

- Binary triple pattern with two variables, representing a formula with two projection variables. For the example $\langle Z_0,R,Z_1\rangle$ see Figure 9.2b. If and only if the output slice is e_1, then the corresponding object tuple encoded by the input indices has a relation R.

The composition $\psi(\psi^T)$ of the matrification of the tensor ψ is an orthogonal projection. That means that applying $\psi(\psi^T)$ is the same map as applying once.

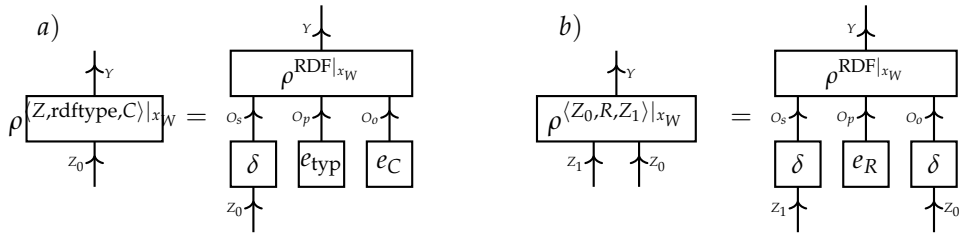


Figure 9.2: Triple patterns of SPARQL as tensor networks. a) Example of unary triple pattern $\langle Z,\mathrm{rdftype},C\rangle$ specifying whether an individual a_1 is a member of class C. Here by 0 we denote the element $I^{-1}(\mathrm{rdftype})$ b) Example of a binary triple pattern $\langle Z_0,R,Z_1\rangle$ specifying whether individuals a_1 and a_2 have a relation R. By e_0,e_C,e_R we denote the one-hot encodings of the enumeration of the resources $rdf:type,C$ and R.

9.3.3.2 Basic Graph Patterns

Generic SPARQL queries are compositions of triple patterns by logical connectives. These triple patterns possibly share projection variables. Statements in SPARQL can be translated into Propositional Logics combining the triple patterns:

SPARQL	Propositional Logics	Tensor Representation
$\{f_1, f_2\}$	$f_1 \wedge f_2$	$\rho^{\wedge}\left[Y_{f_1 \wedge f_2}, Y_{f_1}, Y_{f_2}\right]$
UNION$\{f_1, f_2\}$	$f_1 \vee f_2$	$\rho^{\vee}\left[Y_{f_1 \vee f_2}, Y_{f_1}, Y_{f_2}\right]$
FILTERNOTEXISTS{f}	$\neg f$	$\rho^{\neg}\left[Y_{\neg f}, Y_f\right]$

If a SPARQL query consists of these keywords, we find a straight forward corresponding network of triple patterns and encoded logical connectives, by applying our findings of Sect. 9.2.4. To this end, we prepare for each appearing triple pattern the corresponding pattern tensor, and a copy of $\mathrm{RDF}|_{x_W}$. Here we also copy the term variables O_s, O_p and O_o, to ensure that each copy of $\mathrm{RDF}|_{x_W}$ shares variables with a single pattern tensor. Projection variables are not copied, since we need to keep track of them shared among triple patterns. Then we prepare the relational encoding of logical connectives according to the hierarchy specified in the SPARQL query. Finally we add a e_1-vector to the final head variable representing the complete SPARQL query, to restrict the support to coordiantes corresponding with solution mappings. We then contract the resulting tensor network, leaving all projection variables open.

If a projection variable is not appearing in the SELECT statement in front of the WHERE$\{\cdot\}$-block, we simply exclude it from the open variables of the described contraction. Note that in that case, the coordinates contain solution counts, i.e. how many assignments to the dropped variable have been a 1 coordinate. We can drop this additional information simply by performing a coordinatewise transform with the greater zero indicator $\mathbb{I}_{>0}$.

Here we represented a SPARQL query p consistent of multiple triple pattern by instantiating a head variables to each triple pattern. Alternatively, the more direct effective calculus developed in Sect. 11.5 can be applied and the additional head variables avoided. This is especially compelling, when the WHERE$\{\cdot\}$-block does not contain further keywords, i.e. it is the conjunction of all triple patterns. In that case, we avoid the instantiation of head variables (i.e. close the head variables separately by e_1-vectors) and represent the query by a contraction of all triple pattern tensors.

We further notice, that any propositional formula acting on the head variables of the triple patterns can be expressed by a hierarchical combination of the key words in the above table. To find the expression, one can transform a given formula into its conjunctive or disjunctive normal form and apply the statements according to the apperaing operations $\wedge, \vee$ and $\neg$.

9.4 Probabilistic Relational Models

So far we have studied Markov Logic Networks in Propositional Logics as probability distributions over worlds. In FOL they define probability distributions over relations in worlds with a fixed set of objects. More generally, such models are probabilistic relational models (see for an overview [GT19].

We in this section treat random worlds in first-order logics with fixed domains $\mathcal{U}$

We in this section show, when and how we can interpret likelihoods of Markov Logic Networks in First Order Logic in terms of samples of a Markov Logic Network in Propositional Logics.

9.4.1 Hybrid First-Order Logic Networks

Following [RD06] Markov Logic Networks in first-order logics are templates for distributions, which instantiate random worlds when choosing a set of objects $\mathcal{U}$. Given a fixed set of constants, they then define a distribution over the worlds, which objects correspond with the constants. This applies database semantics, where only those worlds are considered, where the unique name and domain closure assumptions given a set of constants are satisfied. Here we directly define them as exponential families distributing X_W for a given set of objects $\mathcal{U}$. To avoid a similar discussion as in Chapter 6 we directly allow for boolean base measures and call the distributions Hybrid First-Order Logic Networks.

Definition 9.8 (Hybrid First-Order Logic Networks (HFLN)) Let there be a set $\mathcal{Q}$ of first-order logic formulas with maximal arity n, which is enumerated by a selection variable L of dimension p. Further, let there be a set of objects $\mathcal{U}$ and a boolean base measure $\nu\left[O_{[n]}\right]$. The family of Hybrid First-Order Logic Networks $\Gamma^{\mathcal{Q}|_{\mathcal{U}},\nu}$ defined by the tuple $(\mathcal{Q},\mathcal{U},\nu)$ is the exponential family of joint distributions to the variables X_W with the statistics

$$\phi_l^{\mathcal{Q}|_{\mathcal{U}}}\left[X_W = x_W\right] = \langle q_l|_{x_W}\rangle\left[\varnothing\right]$$

and the base measure ν.

Each element of the family $\Gamma^{\mathcal{Q}|_{\mathcal{U}},\nu}$ is represented by a canonical parameter $\theta\left[L\right]$.
The mean parameter polytope is the convex hull of the vectors

$$\gamma^{\mathcal{Q}}\left[X_W = x_W, L\right]$$

to the worlds x_W with $\nu\left[X_W = x_W\right] = 1$. These vectors store are the counts of satisfied groundings to each formula, that is

$$\gamma^{\mathcal{Q}}\left[X_W = x_W, L\right] = \left|o_{q_l} \,:\, q_l|_{x_W}\left[O_{q_l} = o_{q_l}\right] = 1\right| .$$

Each substitution of the variables in q_l by objects in $\mathcal{U}$, which satisfies the formula in the world x_W, therefore provides a factor of $\exp\left[\theta\left[L = l\right]\right]$ to the probability of x_W.

Let us notice, that different to the case of Hybrid Logic Networks treated in Chapter 6, the statistic does not consist of boolean features, when formulas contain variables and we have multiple objects. One could, however, replace each q_l by the set of the possible groundings, i.e. substitutions of the formulas variables by any tuple of objects in $\mathcal{U}$. The resulting distribution would be an Hybrid Logic Network with boolean statistic, which coincides with the HFLN when posing certain weight sharing conditions on θ. The downside of this construction is the increase in the number of features from p to $\sum_{l\in[p]}\left|\mathcal{U}\right|^{\left|O_{q_l}\right|}$. This polynomial in the cardinality of the domain set increase poses significant computational challenges, see **[domingos_markov_2006]**. We will in the next sections explore an alternative way to apply the theory of Chapter 6 and Chapter 7, namely based on importance formulas.

9.4.2 Base measures by importance formulas

The boolean base measure ν of a Hybrid First-Order Logic Network is the subset encoding of the possible worlds which have a non-vanishing probability with respect to any member of the family. We now construct specific base measures based on a fixed grounding tensor of an importance formula. This will reduce the number of object tuples influencing the probability distribution in order to arrive at an interpretation of FOL MLNs as likelihoods to datasets of propositional MLNs.

To this end, we mark pairs of term indices relevant to the distributions by an auxiliary index $j \in [m]$. Given a set $\{o^j_{[n]} \,:\, j \in [m]\}$ of indices to the important tuples we build a set encoding (see Def. 11.1)

$$\underline{p}\left[O_p\right] = \sum_{j\in[m]}\left(\bigotimes_{l\in[n]} e_{o_l^j}\left[O_l\right]\right) .$$

We interpret the tensor $\underline{p}\left[O_p\right]$ as the grounding of a formula, which we call the importance formula.

To have a constant importance formula we define a syntactic representation and restrict the support of the HFLN to those world coinciding with groundings of the importance formula

coinciding with $\underline{p}\left[O_p\right]$ by designing a base measure

$$\nu^{\underline{p}[O_p]}\left[X_W\right] = \begin{cases} 1 & \text{if} \quad p|_{x_W}\left[O_p\right] = \underline{p}\left[O_p\right] \\ 0 & \text{else} \end{cases} .$$

The base measure restricts the HFLN to be those worlds, where $p|_{x_W}$ is coincides with the fixed tensor $\underline{p}\left[O_p\right]$. Intuitively, $p|_{x_W}$ represents certain evidence about a first-order logic world, whereas other formulas are uncertain.

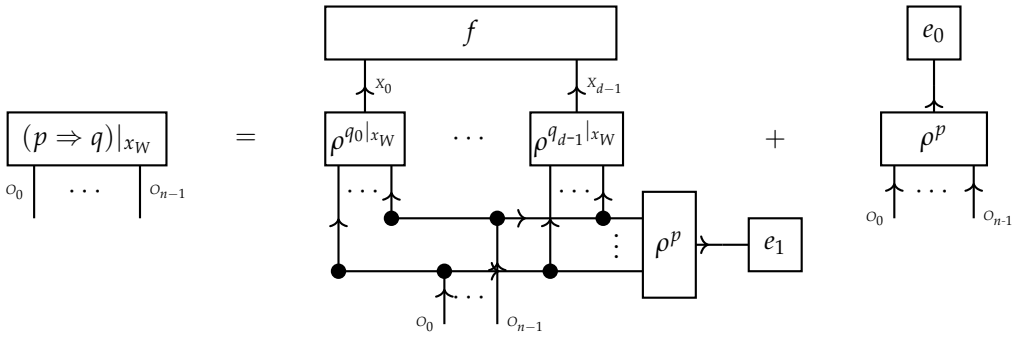
9.4.3 Decomposition of the log likelihood

To reduce the likelihood of a world to we make the assumption that all formulas in a HFLN are of the form

$$q_l\left[O_{q_l}\right] = \left(p\left[O_{[n]}\right] \Rightarrow h_l\left[O_{q_l}\right]\right) \tag{9.1}$$

that is a rule with the importance formula being the premise. In particular, we assume, that they depend on all term variables $O_{[n]}$. If this is not the case, we extend the formula trivially on the missing term variables.

When this assumption holds, we can think of the importance formula as a conditions on individuals to satisfy a statistical relation given by h. We depict the assumption, that any formula is of the form (9.1) in the diagram



where the second summand depends only on the query p and therefore does not appear in the likelihood.

Let us now show, how to decompose the probability of a first-order logic world to a HFLN under the above assumptions. Given a HFLN $\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}[O_p]}}$, the probability of a world x_W with $p|_{x_W} = \underline{p}\left[O_p\right]$ is

$$\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}[O_p]}}\left[X_W = x_W\right] = \frac{1}{\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}[O_p]}\right)} \exp\left[\sum_{l\in[p]} \theta\left[L=l\right]\left\langle (p \Rightarrow h)|_{x_W}\right\rangle[\varnothing]\right]$$

where the partition function is

$$\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}[O_p]}\right) = \sum_{x_W\,:\,p|_{x_W}\left[O_{[n]}\right]=\underline{p}[O_p]} \exp\left[\sum_{l\in[p]} \theta\left[L=l\right]\left\langle (p \Rightarrow h)|_{x_W}\right\rangle[\varnothing]\right] .$$

Let us now decompose the statistics into constant and varying terms. We have

$$\left\langle (p \Rightarrow h)|_{x_W}\right\rangle[\varnothing] = \left\langle p \wedge h|_{x_W}\right\rangle[\varnothing] + \left\langle \neg p|_{x_W}\right\rangle[\varnothing] ,$$

where the the second term is constant among the supported worlds and the first can be enumerated by the satisfied substitutions of p, that is

$$\langle p \wedge h|_{x_W} \rangle [\varnothing] = \sum_{j \in [m]} h|_{x_W} \left[O_{[n]} = o^j_{[n]} \right] .$$

Using these insights we decompose a normalized log likelihood as

$$\frac{1}{m} \ln \left[\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}[O_p]}} [X_W = x_W] \right] = \frac{1}{m} \sum_{j \in [m]} \sum_{l \in [p]} \theta [L = l] \, h|_{x_W} \left[o_{[n]} = o^j_{[n]} \right] - \frac{1}{m} \ln \left[\frac{\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}[O_p]} \right)}{\exp \left[\langle \theta \rangle [\varnothing] \cdot \langle \neg p|_{x_W} \rangle [\varnothing] \right]} \right] \tag{9.2}$$

We notice a similarity with the likelihood in the case of MLNs in propositional logic. When we interpret each tuple $o_{[n]} \in (\mathcal{U})^n$ satisfying $p \left[O_{[n]} = o_{[n]} \right] = 1$ as a datapoint, and choose the formulas

$$\mathcal{F} = \{ h_l \, : \, l \in [p] \}$$

from the propositional equivalents to the formulas $\mathcal{Q}$, the fist term in (9.2) coincides with the first term of the likelihood

$$\mathbb{H} \left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F}, \theta, \mathbb{I})} \right] = \frac{1}{m} \sum_{j \in [m]} \sum_{l \in [p]} \theta [L = l] \, f_l [D(j)] - \ln [\mathcal{Z}(\mathcal{F}, \theta)]$$

However, the partition function couples multiple samples, with possible couplings, and prevents a straight forward interpretation as an empirical dataset. We in the next section present assumptions on the tuples satisfying p, which lead to a factorization of the partition function.

9.4.4 Decomposition of the Partition function

We now make additional assumptions to decompose the partition function of an HFLN as a product of HLN partition functions.

Assumption 9.9 asf

Theorem 9.10 *Let $\nu^{\underline{p}[O_p]} [X_W]$ be a base measure of worlds such that the vectors*

$$\left(q_0|_{\tilde{x_W}} \left[O_0 = o^j_0, \ldots, O_{n-1} = o^j_{n-1} \right], \ldots, q_{d-1}|_{\tilde{x_W}} \left[O_0 = o^j_0, \ldots, O_{n-1} = o^j_{n-1} \right] \right) \tag{9.3}$$

for $j \in [m]$ are independent and identical distributed by a distribution ν, when distributing $\tilde{x_W}$ by $\nu^{\underline{p}[O_p]} [X_W]$.

Let there further be a set of formulas $\mathcal{Q}$, which formulas $q \in \mathcal{Q}$ are representable by

$$q = (p \Rightarrow h)$$

and we find a propositional formula f with

$$h|_{x_W} = \left\langle \{ \rho^{q_k|_{x_W}} \, : \, k \in [d] \} \cup \{ f \} \right\rangle \left[O_{[n]} \right] .$$

We then have for the likelihood of any by $\nu^{\underline{p}[O_p]}[X_W]$ supported world x_W that

$$\frac{1}{m}\ln\left[\mathbb{P}^{(\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}[O_p]})}[x_W]\right] = \frac{1}{m}\ln\left[\mathbb{P}^{\mathcal{F},\theta,\nu}[D]\right] - \frac{1}{m}\sum_{j\in[m]}\ln\left[\nu[X_{[d]} = D(j)]\right]$$

where ν is the distribution of the random vectors (9.3) and by D we denote the dataset

$$D = \left\{\left(q_0|_{x_W}\left[O_0 = o_0^j, \ldots, O_{n-1} = o_{n-1}^j\right], \ldots, q_{d-1}|_{x_W}\left[O_0 = o_0^j, \ldots, O_{n-1} = o_{n-1}^j\right]\right) : j \in [m]\right\}$$

and $\mathcal{F}$ is the set of propositional formulas to $q \in \mathcal{Q}$.

To show the theorem, we show first in the following lemma the factorization of the partition function of the FOL MLN.

Lemma 9.11 *Given the assumptions of The. 9.10, we have*

$$\frac{\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}[O_p]}\right)}{\exp\left[\langle\theta\rangle\left[\varnothing\right]\cdot\langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right]} = \left(\mathcal{Z}(\mathcal{F},\theta,\nu)\right)^m .$$

Proof. We have

$$\begin{aligned}
\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}[O_p]}\right) &= \mathbb{E}_{x_W\sim\nu^{\underline{p}[O_p]}[X_W]}\left[\exp\left[\sum_{q\in\mathcal{Q}}\theta_q\,\langle(p\Rightarrow h)|_{x_W}\rangle\left[\varnothing\right]\right]\right] \\
&= \exp\left[\langle\theta\rangle\left[\varnothing\right]\cdot\langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right]\cdot\mathbb{E}_{x_W\sim\nu^{\underline{p}[O_p]}[X_W]}\left[\exp\left[\sum_{q\in\mathcal{Q}}\theta_q\sum_{j\in[m]}h|_{x_W}\left[O_{[n]} = o_{[n]}^j\right]\right]\right] \\
&= \exp\left[\langle\theta\rangle\left[\varnothing\right]\cdot\langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right]\cdot\mathbb{E}_{x_W\sim\nu^{\underline{p}[O_p]}[X_W]}\left[\prod_{j\in[m]}\exp\left[\sum_{q\in\mathcal{Q}}\theta_q\cdot h|_{x_W}\left[O_{[n]} = o_{[n]}^j\right]\right]\right] \\
&= \exp\left[\langle\theta\rangle\left[\varnothing\right]\cdot\langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right]\cdot\prod_{j\in[m]}\mathbb{E}_{x_W\sim\nu^{\underline{p}[O_p]}[X_W]}\left[\exp\left[\sum_{q\in\mathcal{Q}}\theta_q\cdot h|_{x_W}\left[O_{[n]} = o_{[n]}^j\right]\right]\right]
\end{aligned}$$

Here we used, that since the substitutions of the atom formulas at the respective object tuples are independent, also the variables

$$\exp\left[\theta_q\cdot h|_{x_W}\left[O_{[n]} = o_{[n]}^j\right]\right]$$

for $j \in [m]$ are independent.

Since each $h|_{x_W}\left[O_{[n]}\right]$ depends only on the random value $x_k^j = q_k[O_{[n]}]$ we have

$$\begin{aligned}
\mathbb{E}_{x_W\sim\nu^{\underline{p}[O_p]}[X_W]}\left[\exp\left[\sum_{q\in\mathcal{Q}}\theta_q\cdot h|_{x_W}\left[O_{[n]} = o_{[n]}^j\right]\right]\right] &= \sum_{x_{[d]}\in[2]^d}\mathbb{E}_{x_W\sim\nu^{\underline{p}[O_p]}[X_W]}\left[\forall k\in[d] : q_k[O_{[n]}] = x_k\right] \\
&\qquad\cdot\exp\left[\sum_{f\in\mathcal{F}}\theta_f\cdot f\left[X_{[d]} = x_{[d]}\right]\right] \\
&= \sum_{x_{[d]}\in[2]^d}\nu[X_{[d]} = x_{[d]}]\cdot\exp\left[\sum_{f\in\mathcal{F}}\theta_f\cdot f\left[X_{[d]} = x_{[d]}\right]\right]
\end{aligned}$$

$$= \mathcal{Z}(\mathcal{F}, \theta, \nu)\,.$$

Combining the above, we arrive at the claim. ∎

Proof of The. 9.10. Use Equation 9.2 and the Lem. 9.11. Note that we need to correct the likelihood by the averalge log basemeasure on the data, since that term is appearing in the likelihood of a MLN. ∎

Let us now investigate, when the assumptions of independent data can be matched.

Lemma 9.12 *Let p and $q_0, \ldots, q_{d-1}$ be quantor and constant free and let the index tuples of the support of $\underline{p}\left[O_p\right]$ be pairwise disjoint. Then the vectors (9.3) are pairwise independent.*

Proof. Then we can reduce each sample as dependent only on an independent random world with domain by the respective objects. Quantor and constant-free is needed that this reductions is possible. ∎

Situations where atom base ν measures are not uniform:

- extraction formula being a) conjunctions of predicates: Probability that they are satisfied decreases b) disjunctions of predicates: Probability that they are satisfied increases
- extraction formula coinciding with importance formula: Always satisfied
- extraction formulas contradicting each other, more general not independent from each other

9.4.5 Approximation by Independent Samples

In general, we cannot assume that the $p, q_0, \ldots, q_{d-1}$ do not influence each other.

We approximate the partition function into factors to each solution map of p. This amounts to the assumption, that the atoms created to each tuple are independent.

For example, when solution maps share the resources, which form the arguments of an atom extraction query, they coincide on this atom and are thus dependent.

If the expectations of each sample with respect to the marginalized distributions coincide, the average of empirical distribution also coincides with these (by linearity). When the creation of samples has sufficient mixing properties, the empirical distribution converges to this expectation in the asymptotic case of large numbers of samples.

9.4.6 Extraction of Samples from FOL Knowledge Bases

The decomposition of the likelihood suggests the following approach to generate samples from groundings:

- Define for $k \in [d]$ queries q_k generating the the atoms X_k: Predicates along with assignment of variables / constants to its positions.
- Define a query formula p, which we decompose in the basis CP decomposition for later interpretation of datapoints
- Contract the groundings of each formula f^k with the grounding of p to build a data core

9.4.7 Representation by Tensor Networks

Let us now understand the extraction process as a relation between a tuple of individuals and the extracted world in the factored system of atoms X_k.

$$\mathcal{R} = \{(a_{o_0}, \ldots, a_{o_{n-1}}, X_0, \ldots, X_{d-1}) \,:\, p(a_{o_0}, \ldots, a_{o_{n-1}}) = 1\,,\, \forall_{k\in[d]} :\, X_k = q_k(a_{o_0}, \ldots, a_{o_{n-1}})\}$$

Towards constructing the encoding of this we enumerate the individuals in the set $\mathcal{U}$ and use in the following the respective one-hot encoding

$$e : \mathcal{U} \to \mathbb{R}^{|\mathcal{U}|}\,.$$

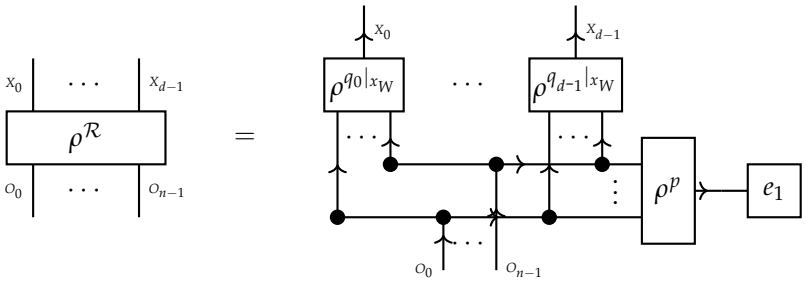
The combination of the queries p and $\{q_0, \ldots, q_{d-1}\}$ by the relational encoding

$$\rho^{\mathcal{R}} \subset \left(\bigotimes_{l\in[n]} \mathbb{R}^r\right) \otimes \left(\bigotimes_{k\in[d]} \mathbb{R}^2\right)$$

defined by

$$\rho^{\mathcal{R}} = \left\{(e_{a_{o_0},\ldots,a_{o_{n-1}}}, X_0, \ldots, X_{d-1}) \,:\, p(a_{o_0}, \ldots, a_{o_{n-1}}) = 1\,,\, \forall_{k\in[d]} :\, X_k = q_k(a_{o_0}, \ldots, a_{o_{n-1}})\right\}\,.$$

We can represent the encoding $\rho^{\mathcal{R}}$ by the diagram



Here the contraction of ρ^p with the truth vector e_1 represents the matching condition posed by p when extracting pairs of individuals.

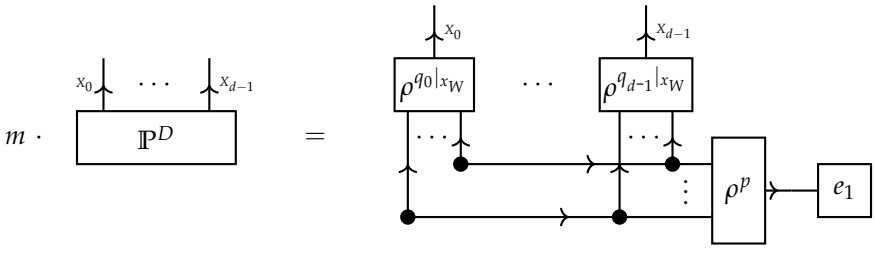
The empirical distribution is then the normalized contraction leaving only the legs to the extracted atomic formulas open, that is

$$\mathbb{P}^D = \frac{\left\langle \rho^{\mathcal{R}} \right\rangle \left[X_{[d]}\right]}{\left\langle \rho^{\mathcal{R}} \right\rangle [\varnothing]}\,.$$

Here the number of extracted data is the denominator

$$m = \left\langle \rho^{\mathcal{R}} \right\rangle [\varnothing]\,.$$

We depict this by



9.4.8 Basis CP Decomposition of extracted data

To connect with the empirical distribution introduced in Sect. 1.8 we now show how the dataset D extracted from the interpretations of the formulas $p, q_0, \ldots, q_{d-1}$ on a FOL world x_W can be represented by tensor networks.

Each datacore is then a contraction with the grounding of a formula, which is contracted with the grounding of the extraction query in the basis CP decomposition,

$$\rho^{D_k} = \left\langle \rho^{q_k|_{x_W}}, p|_{x_W} \right\rangle [J, X_k]$$

see Figure 9.3.

Theorem 9.13 *We have*

$$\left\langle \rho^{\mathcal{R}} \right\rangle \left[X_{[d]} \right] = \left\langle \{ \rho^{D_k}[J, X_k] \,:\, k \in [d] \} \right\rangle \left[X_{[d]} \right]$$

and thus

$$\mathbb{P}^D = \left\langle \{ \rho^{D_k}[J, X_k] \,:\, k \in [d] \} \right\rangle \left[X_{[d]} | \varnothing \right] .$$

Proof. Using that $p|_{x_W}$ is binary and the core can be copied (ref to theorem in binary calculus). ∎

Towards efficient calculation of the data cores, we build a basis CP decomposition of $p|_{x_W}$, where we further demand $\sigma = \mathbb{I}$. This is a collection of basis leg cores $V^{\underline{p}[O_p],l}$ such that

$$\underline{p}\,[O_p]\,[O_{[n]}] = \left\langle \left\{ V^{\underline{p}[O_p],l}[J, O_l] \,:\, l \in [n] \right\} \right\rangle \left[O_{[n]} \right] .$$

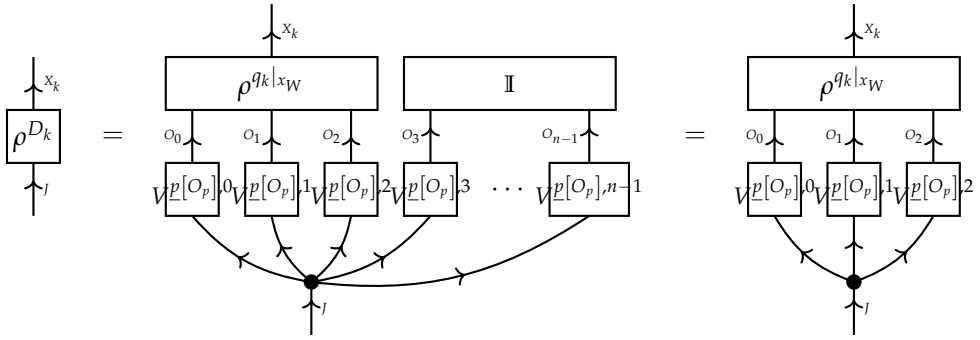


Figure 9.3: Generation of Datacores given a formula f, whose variables (here a_1, a_2, a_3) are selected from an extraction query p. Variables, which are not appearing in the formula f are trivialized over (here $a_4, \ldots, a_n$). For consistency we denoted the index $x_{f(k)}$ by x_k.

The log-likelihood of a MLN is then the contraction of the data cores with the representation of the MLN wrt the atomic formulas q_k.

9.4.9 Representation with auxiliary term variables

When many atom extraction formulas differ only by a constant, we can replace the constant by an auxiliary term variable. The atoms are then the atomizations of this variable (see Sect. 6.2.4), treated as a categorical variable, with respect to the constant in the extraction query. The advantages are that we can avoid the ρ-formalims and directly model the categorical distributions.

9.4.10 Generic Tensor Network Decomposition of Extracted Data

More naturally: Formulas are compositions of predicates. When quantor free then a tensor network.

Remark 9.14 (Alternative Representation of empirical distributions) In many applications such as the computation of log-likelihoods we can use any representation of the empirical distribution by tensor networks. It is thus not necessary to compute the data cores as above, unless one requires a list of the extracted samples. ◊

9.4.11 Design of the Formulas

Most intuitive when labeling individuals by classes. Extraction formulas $q_0, \ldots, q_{d-1}$ can then be defined by subclasses of the member of a class and relations between objects of different classes. We then choose $\mathcal{F}$ as more involved formulas decomposed into connectives acting on these atoms. The extraction query p is then designed based on class memberships to ensure, that the arguments of the formulas are always of specific classes.

We propose to

- Execute an extraction query to get pairs of individuals (the pairDf).
- Propositionalize the FOL Formulas independently on each tuple taking the individuals as a set of constant and filtering on the possible properties of each individuals. (Can understand as adding knowledge that most of the relations do not hold)
- Understand each such generated knowledge base as datapoint and average over them to get the empirical distribution to be fit.
- Fit a MLN describing the statistical relations of unseen results of the extraction query, based on likelihood maximation.

9.5 Generation of FOL worlds

Definition 9.15 (Reproduction of Empirical Distributions) Given an empirical distribution $\mathbb{P}^D \in \bigotimes_{k\in[d]} \mathbb{R}^2$, we say that a triple $(x_W, p, q_{[d]})$ of a FOL world x_W an importance formula p and extraction formulas q_d reproduces $\mathbb{P}^D$, when

$$\mathbb{P}^D = \left\langle \{p|_{x_W}\} \cup \{\rho^{q_k|_{x_W} : k\in[d]}\} \right\rangle \left[X_{[d]}|\varnothing\right] .$$

9.5.1 Samples by single objects

The first reproduction scheme identifies the datapoints with objects $\mathcal{U} = [m]$

Theorem 9.16 *Given a dataset D the world $x_W[L, O]$*

$$x_W[L, O] = \sum_{k\in[d]} \sum_{j\in[m]\,:D_k(j)=1} e_k[L] \otimes e_j[O]$$

reproduces with the trivial importance query and extraction queries coinciding with the predicates the dataset D.

9.6 Samples by pairs of objects

We instantiate multiple objects for each datapoint, one for each variable of the importance formula, i.e. $\mathcal{U} = [m] \times [n]$

9.7 Example: Generation of Knowledge Graphs

So far we have discussed, how MLNs for FOL Knowledge Bases such as Knowledge Graphs can be built by extracting data. Conversely, any binary tensor can be interpreted as a Knowledge Graph. To be more precise, we follow the intuition that the ones coordinates mark possible worlds compatible with the knowledge about a factored system. Each possible world can then be encoded in a subgraph of the Knowledge Graph representing the world.

This amounts to an "inversion" of the data generation process described in the subsection above.

Definition 9.17 (Reproduction of Empirical Distributions) Given an empirical distribution $\mathbb{P}^D \in \bigotimes_{k\in[d]} \mathbb{R}^2$, we say that a tuple $(\mathrm{KG}|_{x_W}, p, \{q_0, \ldots, q_{d-1}\})$ of a Knowledge Graph $\mathrm{KG}|_{x_W}$ and queries p, q_k reproduces $\mathbb{P}^D$, when

$$\mathbb{P}^D = \left\langle \{p|_{x_W}\} \cup \{\rho^{q_k|_{x_W}} : k \in [d]\} \right\rangle \left[X_{[d]}|\varnothing\right] .$$

Any distribution with rational coordinates is an empirical distribution, since each coordinate can be interpreted as the frequency of the respective world in the data D.

9.7.1 Samples by single resources

TBox: The categorical variables of the factored system are the coarsest classes. We define atomic formulas by the state indicators of each categorical variable as in Sect. 6.2.4. Each such atomic formula corresponds with a sub-class of the classes. By definition, each collection of state indicators define thus pairwise disjoint subclasses.

ABox: The samples are represented by single individuals in the Knowledge Graph. Their subclass memberships corresponding with the categorical variables of the system are instantiated whenever the atom is true in the sample.

Theorem 9.18 *Let there any empirical distribution* $\mathbb{P}^D \in \bigotimes_{k\in[d]} \mathbb{R}^2$ *and* $m \in \mathbb{N}$ *such that* $\mathrm{im}\left(m \cdot \mathbb{P}^D\right) \subset \mathbb{N}$. *Then the tuple* $(\mathrm{KG}|_{x_W}, p, \{q_0, \ldots, q_{d-1}\})$ *defined by*

$$\begin{aligned}\mathrm{KG}|_{x_W} = &\bigcup_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \{(s_{j,x_0,\ldots,x_{d-1}} \quad \text{rdf:type} \quad C) : j \in [m \cdot \mathbb{P}^D(x_0, \ldots, x_{d-1})]\} \\ &\bigcup_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \{(s_{j,x_0,\ldots,x_{d-1}} \quad \text{rdf:type} \quad C_k) : j \in [m \cdot \mathbb{P}^D(x_0, \ldots, x_{d-1})], k \in [d], x_k = 1\}\end{aligned}$$

p= SELECT { ?x } WHERE { ?x rdf:type *C .}*

q_k = SELECT { ?x } WHERE { ?x rdf:type *C_k .}*

reproduces $\mathbb{P}^D$.

Proof. With respect to any enumeration of the resources of $\mathrm{KG}|_{x_W}$ we have

$$p|_{x_W} = \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \sum_{j\in[m\cdot\mathbb{P}^D(x_0,\ldots,x_{d-1})]} e_{s_{j,x_0,\ldots,x_{d-1}}}$$

and

$$q_k|_{x_W} = \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2] : x_k=1} \sum_{j\in[m\cdot\mathbb{P}^D(x_0,\ldots,x_{d-1})]} e_{s_{j,x_0,\ldots,x_{d-1}}} .$$

Summing over the resource variables of these tensors in a contraction we get

$$\left\langle \{p|_{x_W}\} \cup \{\rho^{q_k|_{x_W}} : k \in [d]\} \right\rangle \left[X_{[d]}\right] = \sum_{k \in [d]} m \cdot \mathbb{P}^D(x_0, \ldots, x_{d-1}) \cdot e_{x_0,\ldots,x_{d-1}} = m \cdot \mathbb{P}^D$$

and therefore

$$\left\langle \{p|_{x_W}\} \cup \{\rho^{q_k|_{x_W}} \ : \ k \in [d]\} \right\rangle \left[X_{[d]}|\varnothing\right] = \mathbb{P}^D \, .$$

■

In this simple Knowledge Graph, Description Logic is expressive enough to represent any formula q composed of the formulas $q_0, \ldots, q_{d-1}$.

9.7.2 Samples by pairs of resources

Remark 9.19 (Refinement of the Samples) We can split each sample node into a pair of individuals. For this we need to specify, which each class membership will be encoded in a unary or binary attribute of the splitted individuals. This specification is possible based on the extraction query and the atomic formulas. ♢

Taking any importance query p, which has no permutation symmetries, we can instantiate each projection variable for each sample and prepare the links according to the triple patterns. When the atom queries $q_0, \ldots, q_{d-1}$ have different triple patterns compared with p, we instantiate those in cases where $x_k = 1$.

Label individuals $a_{j,l}$ by dataindex and variable index and

Theorem 9.20 *Let there be queries $p, q_0, \ldots, q_{d-1}$, a Knowledge Graph $\mathrm{KG}|_{x_W}$ containing individuals $a_{j,l}$ and a data map D. If*

$$p|_{x_W} = \sum_j \bigotimes_{l \in [n]} e_{j,l}$$

and for any $k \in [d]$

$$q_k|_{x_W} = \sum_{j : D^k(j) = 1} \bigotimes_{l \in q_k} e_{j,l} \, .$$

Then the tuple $(\mathrm{KG}|_{x_W}, p, \{q_0, \ldots, q_{d-1}\})$ reproduces $\mathbb{P}^D$.

Proof. It can be shown that the contraction is

$$\sum_j \bigotimes_{k \in [d]} e_{D^k(j)} \, .$$

■

Theorem 9.21 *The Knowledge Graph*

$$\mathrm{RDF}|_{x_W} = \sum_{t \in p} \sum_j t(a_{j,l}) + \sum_{k \in [d]} \sum_{t \in q_{k \in [d]}} \sum_{j : D^k(j) = 1} t(a_{j,l})$$

reproduces the data, if the summed tensors are pairwise orthogonal. Here we denote by t(..) the one-hot encoding of the tuple when mapping the corresponding individuals on the projection variables of the triple pattern t.

9.7.3 General orthogonal projection approach

Given any triples, such that their composed projections to pairwise different triple patterns vanish. One reproducing KG is then the sum of the projection of any reproducing KG.

Theorem 9.22 *Let $p, q_0, \dots, q_{d-1}$ contain of triple patterns such that for $t \neq \tilde{t}$*

$$P_t \circ P_{\tilde{t}} = 0 \,.$$

For any KG reproducing the data D also the KG

$$\sum_t P_t \mathrm{RDF}|_{x_W}$$

is a reproducing KG.

Proof. By coinciding on all triple patterns t since

$$P_t \left(\sum_{\tilde{t}} P_{\tilde{t}} \mathrm{RDF}|_{x_W} \right) = P_t P_t \mathrm{RDF}|_{x_W} = P_t \mathrm{RDF}|_{x_W} \,.$$

Then the extraction contraction on $\mathrm{KG}|_{x_W}$ and $\sum_t P_t \mathrm{RDF}|_{x_W}$ coincides. ∎

From this perspective, it is obvious that any triple in the orthogonal complement of all projections can be added and the KG stays reproducing.

9.8 Discussion

Models of FOL are called Probabilistic Relational Models. Extensions are models that also handle structural uncertainty, i.e. distributions of worlds with varying $\mathcal{U}$.

Statistical Models of Knowledge Graphs going beyond the typical single edge type perspective of network science [Bar16; Gio17].

Part III

Contraction Calculus

10

Coordinate Calculus

In the previous chapters, information to states has been stored in coordinates of a tensor. To distinguish from other schemes of calculus such as the basis calculus (see Chapter 11), we call this scheme of storing and retrieving information the coordinate calculus.

10.1 One-hot encodings as basis

Let us first state, that the one-hot encodings, which we have used to motivate tensor representations, build an orthonormal basis of the respective tensor spaces.

Lemma 10.1 *The image of the one-hot encoding map is an orthonormal basis of the tensor space* $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$*, that is for any* $x_{[d]}, \tilde{x}_{[d]} \in \bigtimes_{k\in[d]} [m_k]$ *we have*

$$\left\langle e_{x_{[d]}}\left[X_{[d]}\right], e_{\tilde{x}_{[d]}}\left[X_{[d]}\right]\right\rangle [\varnothing] = \delta_{x_{[d]},\tilde{x}_{[d]}} := \begin{cases} 1 & \text{if} \quad x_{[d]} = \tilde{x}_{[d]} \\ 0 & \text{else} \end{cases} .$$

Any element $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ *has a decomposition*

$$T\left[X_{[d]}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} T\left[X_{[d]} = x_{[d]}\right] \cdot e_{x_{[d]}}\left[X_{[d]}\right] .$$

We notice that the coordinates are the weights to the basis elements in the one-hot decomposition.

Proof. The first claim follows from an elementary decomposition of one-hot encodings and the orthogonality of basis vectors as

$$\left\langle e_{x_{[d]}}\left[X_{[d]}\right], e_{\tilde{x}_{[d]}}\left[X_{[d]}\right]\right\rangle [\varnothing] = \prod_{k\in[d]} \left\langle e_{x_k}[X_k], e_{\tilde{x}_k}[X_k]\right\rangle [\varnothing] = \prod_{k\in[d]} \delta_{x_k,\tilde{x}_k} = \delta_{x_{[d]},\tilde{x}_{[d]}} .$$

To show the second claim, it is enough to notice that for any $\tilde{x}_{[d]} \in \bigtimes_{k\in[d]} [m_k]$ we have

$$\begin{aligned} \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} T\left[X_{[d]} = x_{[d]}\right] \cdot e_{x_{[d]}}\left[X_{[d]} = \tilde{x}_{[d]}\right] &= \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} T\left[X_{[d]} = x_{[d]}\right] \cdot \delta_{x_{[d]},\tilde{x}_{[d]}} \\ &= T\left[X_{[d]} = \tilde{x}_{[d]}\right] . \end{aligned}$$

■

Any tensor can be understood as a coordinate encoding of a real-valued function, as we define next.

Definition 10.2 Given any real-valued function

$$f : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R}$$

we define the coordinate encoding by

$$T^f \left[X_{[d]}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} f(x_{[d]}) \cdot e_{x_{[d]}} \left[X_{[d]}\right] .$$

In Part I and Part II we did not distinguish between a real-valued function f and its coordinate encoding T^f. Based on coordinate encodings, we now show, that function evaluation can be performed by contractions.

Theorem 10.3 (Function evaluation in Coordinate Calculus) *Given any real-valued function*

$$f : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R}$$

and any input state $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$, we have

$$f(x_{[d]}) = \left\langle T^f \left[X_{[d]}\right], e_{x_{[d]}} \left[X_{[d]}\right] \right\rangle [\varnothing] .$$

Proof. We use the decomposition in Lem. 10.1 and have by linearity of contractions for any index tuple $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$

$$\begin{aligned}
\left\langle T^f \left[X_{[d]}\right], e_{x_{[d]}} \left[X_{[d]}\right] \right\rangle [\varnothing] &= \sum_{\tilde{x}_{[d]}\in\bigtimes_{k\in[d]}[m_k]} T^f \left[X_{[d]} = \tilde{x}_{[d]}\right] \cdot \left\langle e_{\tilde{x}_{[d]}} \left[X_{[d]}\right], e_{x_{[d]}} \left[X_{[d]}\right] \right\rangle [\varnothing] \\
&= \sum_{\tilde{x}_{[d]}\in\bigtimes_{k\in[d]}[m_k]} f(\tilde{x}_{[d]}) \cdot \delta_{\tilde{x}_{[d]}, x_{[d]}} \\
&= f(x_{[d]})
\end{aligned}$$

where we used that one-hot encodings are orthonormal. ∎

Coordinate calculus is the representation of real-valued functions as tensors, from which its evaluations can be retrieved by the scheme of The. 10.3. This is in contrast to the basis calculus scheme to be discussed (see The. 11.9), where the contraction-based evaluations of functions outputs one-hot encodings.

Tensors of large orders often admit a decomposition by tensor networks. We in the next theorem show, how such a decomposition can be exploited for efficient contractions and in particular coordinate retrieval.

Theorem 10.4 *Given a tensor network $\mathcal{T}^{\mathcal{G}}$ on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, disjoint subsets $A, B \subset \mathcal{V}$ and $x_B \in [m_B]$, we have*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_A, X_B = x_B] = \langle \{ \langle T^e \rangle [X_{e/B}, X_{e\cap B} = x_{e\cap B}] \,:\, e \in \mathcal{E} \} \rangle [X_A] .$$

Proof. By definition of contractions we have for any x_A

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_A = x_A, X_B = x_B] = \sum_{x_{\mathcal{V}/(A\cup B)}\in\bigtimes_{v\in\mathcal{V}/(A\cup B)}[m_v]} \prod_{e\in\mathcal{E}} T^e [X_{e/B} = x_{e/B}, X_{e\cap B} = x_{e\cap B}]$$

$$= \left\langle \{ \langle T^e \rangle [\varnothing] X_{e/(A\cup B)}, X_{e\cap A} = x_{e\cap A}, X_{e\cap B} = x_{e\cap B} : e \in \mathcal{E} \} \right\rangle [\varnothing]$$
$$= \langle \{ \langle T^e \rangle [\varnothing] X_{e/B}, X_{e\cap B} = x_{e\cap B} : e \in \mathcal{E} \} \rangle [X_A = x_A]$$

and the claim follows. ∎

If we retrieve a single coordinate of a tensor, we have the situation $A = \varnothing$, $B = \mathcal{V}$. In that case, Theorem 10.4 shows, that the coordinate is the product of the coordinates of the cores.

10.2 Coordinatewise Transforms

Let us now discuss a scheme to perform transformations of tensors. We call them coordinatewise, when the target tensor has the same variables as the input tensors, and each coordinate of the target tensor depends only on the respective coordinates of the input tensors.

Definition 10.5 Let $h : \mathbb{R}^p \to \mathbb{R}$ be a function. Then the coordinatewise transform of tensors $T^l \left[X_{[d]} \right]$, where $l \in [p]$, under f is the tensor

$$h \left(T^0, \ldots, T^{p-1} \right) \left[X_{[d]} \right]$$

with coordinates

$$h \left(T^0, \ldots, T^{p-1} \right) \left[X_{[d]} = x_{[d]} \right] = h \left(T^0 \left[X_{[d]} = x_{[d]} \right], \ldots, T^{p-1} \left[X_{[d]} = x_{[d]} \right] \right) .$$

Coordinatewise transforms in case of $p = 1$ have been indicated by ellipses in the diagrammatic depiction of contractions. We will provide a generic tensor network representation in Chapter 11, see The. 11.11.

Example 10.6 (Hadamard products as coordinatewise transforms) Hadamard products of tensors (see Example 0.14) are a special way of coordinate calculus, where the transform is the product and thus

$$\cdot \left(T^0, \ldots, T^{p-1} \right) \left[X_{[d]} \right] = \left\langle \{ T^l \left[X_{[d]} \right] : l \in [p] \} \right\rangle \left[X_{[d]} \right] .$$

These hadamard products are applied in the effective computation of conjunctions, as we will discuss in more detail in Sect. 11.5. ♢

Example 10.7 (Exponentiation of energies) In Def. 1.27 we introduced exponential families, based on the exponentiation of energies. For a statistic ϕ, a base measure ν and a canonical parameter θ we defined

$$\mathbb{P}^{(\phi,\theta,\nu)} = \frac{\left\langle \exp \left[\langle \gamma^\phi, \theta \rangle \left[X_{[d]} \right], \nu \left[X_{[d]} \right] \right] \right\rangle \left[X_{[d]} \right]}{\left\langle \exp \left[\langle \gamma^\phi, \theta \rangle \left[X_{[d]} \right], \nu \left[X_{[d]} \right] \right] \right\rangle [\varnothing]} .$$

Both the nominator and the denominator involve a coordinatewise transform of the energy tensor $E^{(\phi,\theta,\nu)}[X_{[d]}]$ by the exponentiation. The. 1.30 provided a transform-free contraction expression by relational encodings, which is the central tool of basis calculus (see Chapter 11). ♢

10.3 Directed Tensors

Directionality as defined in Def. 0.17 is a constraint on the structure of a tensor, namely that the contraction leaving only incoming variables open trivializes the tensor. We have motivated such constraints by conditional distributions, see Def. 1.13, and refered to Markov Networks (see

Def. **??**) satisfying these by Bayesian Networks (see Def. 1.24). To support our findings therein, we now discuss in more detail the connection between directed hypergraphs and directed tensors.

Definition 10.8 (Directed Hypergraph) A directed hyperedge is a hyperedge, which node set is split into disjoint sets of incoming and outgoing nodes. We say a hypercore T^e decorating a directed hyperedge respects the direction, when it is a conditional probability tensor with respect to the direction of the hyperedge. The hypergraph is acyclic, when there is no nonempty cycle of node tuples (v_1, v_2), such that v_1 is an incoming node and v_2 an outgoing node of the same hyperedge.

There can be multiple ways to direct a tensor, with an extreme example being Diracs Delta Tensors to be introduced in the next example. More general examples are relational encodings of invertible functions.

Example 10.9 (Dirac Delta Tensors) Given a set of variables $X_{[d]} = X_0, \ldots, X_{d-1}$ with a constant dimension m, Diracs Delta Tensor is the element

$$\delta^{[d],m}\left[[d], m\right] \in \bigotimes_{k\in[d]} \mathbb{R}^m$$

with coordinates

$$\delta^{[d],m}\left[[d], m\right] = \begin{cases} 1 & \text{if} \quad x_0 = \ldots = x_{d-1} \\ 0 & \text{else} \end{cases} .$$

The contractions with respect to subsets $\tilde{\mathcal{V}} \subset [d]$ are

$$\left\langle \delta^{[d],m} \right\rangle [X_{\tilde{\mathcal{V}}}] = \begin{cases} m & \text{if} \quad \tilde{\mathcal{V}} = \varnothing \\ \mathbb{I}\,[X_{\tilde{\mathcal{V}}}] & \text{if} \quad \left|\tilde{\mathcal{V}}\right| = 1 \\ \delta^{\tilde{\mathcal{V}},m}\left[\tilde{\mathcal{V}}, m\right] & \text{else} \end{cases} .$$

Thus are directed for any orientation of the respective edge with exactly one incoming variable. ◊

We can use Diracs Delta Tensors to represent any contraction of a tensor network on a hypergraph by a tensor network on a graph, as we show next.

Lemma 10.10 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a hypergraph and $\mathcal{T}^{\mathcal{G}}$ a tensor network on $\mathcal{G}$. We build a graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}} \cup \Delta^{\mathcal{G}})$ and a tensor network $\mathcal{T}^{\tilde{\mathcal{G}}}$ by*

- *Recolored Edges $\tilde{\mathcal{E}} = \{\tilde{e} \,:\, e \in \mathcal{E}\}$ where $\tilde{e} = \{v^e \,:\, v \in e\}$, which decoration tensor $T^{\tilde{e}}$ has same coordinates as T^e*
- *Nodes $\tilde{\mathcal{V}} = \bigcup_{e\in\mathcal{E}} \tilde{e}$*
- *Delta Edges $\Delta^{\mathcal{G}} = \left\{\{v\} \cup \{v^e \,:\, e \ni v\} \,:\, v \in \mathcal{V}\right\}$ each of which decorated by a delta tensor $\delta^{\{v^e : e\ni v\}}$*

Then we have

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}] = \left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\mathcal{V}}] .$$

Proof. For any $x_{\mathcal{V}}$ we have

$$\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = \left\langle \{T^{\tilde{e}}[X_{\{v^e : v\in e\}}] : e \in \mathcal{E}\} \cup \{\delta^{\{v\}\cup\{v^e : e\ni v\}}[X_{\{v^e : e\ni v\}}, X_v = x_v] : v \in \mathcal{V}\} \right\rangle [\varnothing]$$

$$= \left\langle \{T^{\tilde{e}}[X_{\{v^e : v \in e\}} = x_{\{v : v \in e\}}] : e \in \mathcal{E}\} \right\rangle [\varnothing]$$
$$= \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \,,$$

which establishes the claim. ∎

10.3.1 Normation

Normed tensors (see Def. 0.18) are directed and directed tensors invariant under normation wrt their incoming and outgoing variable, as we show next.

Theorem 10.11 *For any tensor network $\mathcal{T}^{\mathcal{G}}$ on variables $\mathcal{V}$ that can be normed with respect to $\mathcal{V}^{\text{in}}$ and $\mathcal{V}^{\text{out}}$, the normation is directed with $\mathcal{V}^{\text{in}}$ incoming and $\mathcal{V}^{\text{out}}$ outgoing.*

Proof. We have for any incoming state $x_{\mathcal{V}^{\text{in}}} \in \bigtimes_{v \in \mathcal{V}^{\text{in}}} m_v$ that

$$\left\langle \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\mathcal{V}^{\text{in}} | \mathcal{V}^{\text{out}}\right], e_{x_{\mathcal{V}^{\text{in}}}} \right\rangle [\varnothing] = \frac{\left\langle \mathcal{T}^{\mathcal{G}} \cup \{e_{x_{\mathcal{V}^{\text{in}}}}\} \right\rangle [\varnothing]}{\left\langle \mathcal{T}^{\mathcal{G}} \cup \{e_{x_{\mathcal{V}^{\text{in}}}}\} \right\rangle [\varnothing]} \,.$$

By Def. 0.17, $\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\mathcal{V}^{\text{out}} | \mathcal{V}^{\text{in}}\right]$ is thus directed. ∎

The normation operation coincides in cases of non-negative tensors with the conditioning of a Markov Network representing a probability distribution.

10.3.2 Normation Equations

Theorem 10.12 (Normation as a Contraction Equation) *For any on $\mathcal{V}^{\text{in}}$ normable tensor $T[X_{\mathcal{V}}]$, where $\mathcal{V}^{\text{in}} \dot{\cup} \mathcal{V}^{\text{out}} = \mathcal{V}$, we have*

$$\langle T \rangle [X_{\mathcal{V}}] = \langle \langle T \rangle [X_{\mathcal{V}^{\text{out}}} | X_{\mathcal{V}^{\text{in}}}], \langle T \rangle [X_{\mathcal{V}^{\text{in}}}] \rangle [X_{\mathcal{V}}] \,.$$

Proof. Let us choose indices $x_{\mathcal{V}^{\text{in}}}$ and $x_{\mathcal{V}^{\text{out}}}$. We have that

$$\langle T \rangle [X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}} | X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}] = \frac{\langle T \rangle [X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}, X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}]}{\langle T \rangle [X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}]}$$

and therefor

$$\langle T \rangle [X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}, X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}] = \langle T \rangle [X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}} | X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}] \cdot \langle T \rangle [X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}]$$

Since the equation holds for arbitrary indices, the claim is established. ∎

Theorem 10.13 (Generic Chain Rule) *For any Tensor $T[X_{\mathcal{V}}]$ and any total order $\prec$ on the nodes $\mathcal{V}$ we have*

$$T[X_{\mathcal{V}}] = \left\langle \{\langle T \rangle \left[X_v | X_{\{\tilde{v} : \tilde{v} \prec v, \tilde{v} \neq v\}}\right] : v \in \mathcal{V}\} \right\rangle [X_{\mathcal{V}}]$$

Proof. We apply The. 10.12 on the tensor

$$\langle T \rangle \left[X_v, X_{\{\tilde{v} : v \prec \tilde{v}, \tilde{v} \neq v\}} | X_{\{\tilde{v} : \tilde{v} \prec v, \tilde{v} \neq v\}} = x_{\{\tilde{v} : \tilde{v} \prec v, \tilde{v} \neq v\}}\right] \,,$$

where $v \in \mathcal{V}$ and $x_{\mathcal{V}}$ are chosen arbitrarly. For any $v \in \mathcal{V}$ we get

$$\langle T \rangle \left[X_v, X_{\{\tilde{v} : v \prec \tilde{v}, \tilde{v} \neq v\}} | X_{\{\tilde{v} : \tilde{v} \prec v, \tilde{v} \neq v\}} \right] = \left\langle \langle T \rangle \left[X_{\{\tilde{v} : v \prec \tilde{v}, \tilde{v} \neq v\}} | X_v, X_{\{\tilde{v} : \tilde{v} \prec v, \tilde{v} \neq v\}} \right], \langle T \rangle \left[X_v | X_{\{\tilde{v} : \tilde{v} \prec v, \tilde{v} \neq v\}} \right] \right\rangle [X_{\mathcal{V}}] .$$

Applying this equation iteratively and making use of the commutation of contractions we get for any $v \in \mathcal{V}$

$$\langle T \rangle \left[X_v, X_{\{\tilde{v} : v \prec \tilde{v}, \tilde{v} \neq v\}} | X_{\{\tilde{v} : \tilde{v} \prec v, \tilde{v} \neq v\}} \right] = \left\langle \langle T \rangle \left[X_{\tilde{v}} | X_{\{\bar{v} : \bar{v} \prec \tilde{v}, \bar{v} \neq \tilde{v}\}} \right] : v \prec \tilde{v} \right\rangle [X_{\mathcal{V}}] .$$

With the maximal node v, that is the v, such that no $\tilde{v} \in \mathcal{V}$ with $v \prec \tilde{v}$ and $v \neq \tilde{v}$ exists, this is the claim. ∎

10.3.3 Contraction of Directed Tensors

Let us now investigate, which contractions inherit the directionality of the tensors.

Theorem 10.14 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic hypergraph, such that each node $v \in e$ appears at most in one hyperedge as an outgoing variable and denote $\mathcal{V}^{\text{in}}$ as those nodes, which do not appear as outgoing variables. For any tensor network $\mathcal{T}^{\mathcal{G}}$ respecting the direction of $\mathcal{G}$ we have that*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}^{\text{in}}}] = \mathbb{I} [X_{\mathcal{V}^{\text{in}}}] ,$$

that is $\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}]$ is a directed tensor with $\mathcal{V}^{\text{in}}$ incoming and $\mathcal{V} / \mathcal{V}^{\text{in}}$ outgoing.

Proof. We show the theorem only for the case of hypergraphs, where variables are appearing at most in two hyperedges. If a hypergraph fails to satisfy this assumption, we apply Lem. 10.10 and add delta tensors copying the variables, which are appearing in multiple tensors, and arrive at a tensor network with nodes appearing in at most two hyperedges.

We show the theorem over induction on the number n of cores.

$n = 1$: The claim holds trivially, when $\mathcal{T}^{\mathcal{G}}$ consists of a single core.

$n \to n - 1$: Let us assume, that the claim holds for graphs with $n - 1$ hyperedges and let $\mathcal{T}^{\mathcal{G}}$ be a tensor network with n hyperedges. Since the hypergraph is acyclic, we find an edge $e \in \mathcal{E}$ such that all outgoing nodes of e are not appearing as an incoming node in any edge. We then apply Theorem 13.1 and get

$$\begin{aligned} \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}^{\text{in}}}] &= \left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E} / \{e\})} \cup \{ T^e [X_{e^{\text{in}}}, X_{e^{\text{out}}}] \} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &= \left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E} / \{e\})} \cup \{ \langle T^e \rangle [X_{e^{\text{in}}}] \} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &= \left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E} / \{e\})} \cup \{ \mathbb{I} [X_{e^{\text{in}}}] \} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &\left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E} / \{e\})} \} \right\rangle [X_{\mathcal{V}^{\text{in}}}] . \end{aligned}$$

We then notice that the hypergraph $(\mathcal{V}, \mathcal{E} / \{e\})$ has $n - 1$ hyperedges and each node appears at most once as an incoming and at most once as an outgoing node. Thus, we apply the assumption of the induction and get

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}^{\text{in}}}] = \left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E} / \{e\})} \} \right\rangle [X_{\mathcal{V}^{\text{in}}}] = \mathbb{I} [X_{\mathcal{V}^{\text{in}}}] .$$

∎

10.4 Proof of Hammersley-Clifford Theorem

Let us now proof the Hammersley-Clifford theorem formulated in Chapter 1 as The. 1.22. Different to the original statement (see [**hammersley_markov_1971**]), we here proof the analogous statement for hypergraphs, where we have to demand the property of clique-capturing defined in Def. 1.21. We start with showing the following Lemmata to be exploited in the proof.

Lemma 10.15 *Let $T[X_{\mathcal{V}}]$ be a positive tensor and $y_{\mathcal{V}}$ an arbitrary index. Then we have*

$$T[X_{\mathcal{V}}] = \left\langle \left(\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} : \bar{\mathcal{V}} \subset \tilde{\mathcal{V}} \subset \mathcal{V} \right\rangle [X_{\mathcal{V}}] ,$$

where the exponentiation is performed coordinatewise and positivity of T ensures the well-definedness.

Proof. It suffices to show, that for an arbitrary index $x_{\mathcal{V}}$ be an arbitrary index we have

$$T[X_{\mathcal{V}} = x_{\mathcal{V}}] = \prod_{\tilde{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} .$$

We do this by applying a logarithm on the right hand side and grouping the terms by $\bar{\mathcal{V}}$ as

$$\begin{aligned}
&\ln \left[\prod_{\tilde{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} \Bigg] \\
&= \sum_{\bar{\mathcal{V}} \subset \mathcal{V}} \ln \left[\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right] \left(\sum_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}} : \bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} (-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|} \right) \\
&= \sum_{\bar{\mathcal{V}} \subset \mathcal{V}} \ln \left[\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right] \left(\sum_{i \in [|\mathcal{V}|-|\bar{\mathcal{V}}|]} (-1)^i \binom{|\mathcal{V}| - |\bar{\mathcal{V}}|}{i} \right)
\end{aligned}$$

Now, by the generic binomial theorem we have that for $n \in \mathbb{N}, n \neq 0$

$$0 = (1-1)^n = \sum_{i \in [n]} (-1)^i \binom{n}{i} .$$

Therefore, the summands for $\bar{\mathcal{V}} \neq \mathcal{V}$ vanish and we have

$$\begin{aligned}
&\ln \left[\prod_{\tilde{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} \right] \\
&= \ln \left[T[X_{\mathcal{V}} = x_{\mathcal{V}}] \right] \left(\sum_{i \in [0]} (-1)^i \binom{0}{i} \right) \\
&= \ln \left[T[X_{\mathcal{V}} = x_{\mathcal{V}}] \right] .
\end{aligned}$$

Applying the exponential function on both sides establishes the claim. ∎

Lemma 10.16 *Let T be a positive tensor, $\tilde{\mathcal{V}} \subset \mathcal{V}$ and arbitrary subset and $x_{\tilde{\mathcal{V}}}$ an arbitrary index.*

When there are $A, B \in \tilde{\mathcal{V}}$, such that

$$\langle T \rangle \left[X_{A,B} | X_{\mathcal{V}/\{A,B\}} \right] = \left\langle \langle T \rangle \left[X_A | X_{\mathcal{V}/\{A,B\}} \right], \langle T \rangle \left[X_B | X_{\mathcal{V}/\{A,B\}} \right] \right\rangle [X_{\bar{\mathcal{V}}}]$$

then

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} = 1 \,.$$

Proof. We abbreviate

$$Z_{\bar{\mathcal{V}}} = \langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] .$$

By reorganizing the sum over $\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}$ into $\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}/A \cup B$ we have

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} (Z_{\bar{\mathcal{V}}})^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} = \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}/\{A,B\}} \left(\frac{Z_{\bar{\mathcal{V}}} \cdot Z_{\bar{\mathcal{V}} \cup \{A,B\}}}{Z_{\bar{\mathcal{V}} \cup \{A\}} \cdot Z_{\bar{\mathcal{V}} \cup \{B\}}} \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} . \tag{10.1}$$

From the independence assumption it follows that for any index x

$$\begin{aligned} &\langle T \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B \right] \\ &= \langle T \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \\ &= \langle T \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B \right] \end{aligned}$$

Applying this in each squares bracket term of (10.1) we get

$$\begin{aligned} \frac{Z_{\bar{\mathcal{V}}}}{Z_{\bar{\mathcal{V}} \cup \{A\}}} &= \frac{\langle T \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B \right]}{\langle T \rangle \left[X_A = y_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B \right]} \\ &= \frac{\langle T \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B \right]}{\langle T \rangle \left[X_A = y_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B \right]} \\ &= \frac{Z_{\bar{\mathcal{V}} \cup \{B\}}}{Z_{\bar{\mathcal{V}} \cup \{A,B\}}} \,. \end{aligned}$$

Thus, each factor in (10.1) is trivial, which establishes the claim. ∎

We are finally ready to proof the Hammersley-Clifford The. 1.22 based on the Lemmata above.

Proof of The. 1.22. By Lem. 10.15 we have for any index $x_{\mathcal{V}}$

$$\mathbb{P}\left[X_{\mathcal{V}} = x_{\mathcal{V}} \right] = \prod_{\tilde{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\mathbb{P}\left[X_{\bar{\mathcal{V}}} = x_{\bar{\mathcal{V}}}, X_{\mathcal{V}/\bar{\mathcal{V}}} = y_{\mathcal{V}/\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}}$$

For any subset $\tilde{\mathcal{V}} \subset \mathcal{V}$, which is not contained in a hyperedge, we find $A, B \in \tilde{\mathcal{V}}$ such that X_A is independendent on X_B conditioned on $X_{\tilde{\mathcal{V}}/\{A,B\}}$. If no such nodes $A, B \in \tilde{\mathcal{V}}$ exists, $\tilde{\mathcal{V}}$ would be contained in a hyperedge, since the hypergraph is assumed to be clique-capturing. By Lem. 10.16 we then have

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\mathbb{P}\left[X_{\bar{\mathcal{V}}} = x_{\bar{\mathcal{V}}}, X_{\mathcal{V}/\bar{\mathcal{V}}} = y_{\mathcal{V}/\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} = 1 \,.$$

We label by a function

$$\alpha : \{\tilde{\mathcal{V}} : \exists e \in \mathcal{E} : \tilde{\mathcal{V}} \subset e\} \to \mathcal{E}$$

the remaining node subsets by a hyperedge containing the subset. We build the tensor

$$T^e\left[X_e\right] = \prod_{\tilde{\mathcal{V}} : \alpha(\tilde{\mathcal{V}}) = e} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\mathbb{P}\left[X_{\bar{\mathcal{V}}} = x_{\bar{\mathcal{V}}}, X_{\mathcal{V}/\bar{\mathcal{V}}} = y_{\mathcal{V}/\bar{\mathcal{V}}}\right]\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} .$$

and get, that

$$\begin{aligned}\mathbb{P}\left[X_{\mathcal{V}}\right] &= \left\langle \{T^e \,:\, e \in \mathcal{E}\}\right\rangle\left[X_{\mathcal{V}}\right] \\ &= \left\langle \{T^e \,:\, e \in \mathcal{E}\}\right\rangle\left[X_v|\varnothing\right] .\end{aligned}$$

We have thus constructed a Markov Network with trivial partition function, which contraction coincides with the probability distribution. ∎

10.5 Differentiation of Contraction

The structured mean field approaches discussed in Chapter 2 used differentiations of the parametrized tensor networks. Let us now develop in more detail, how the contraction of tensor networks with variable cores is differentiated. We capture in additional variables Y selecting the coordinates of a tensor, which are varied in a differentiation.

Lemma 10.17 *For any tensor network $\mathcal{T}^{\mathcal{G}}$ with positive T^e we have*

$$\begin{aligned}\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] = \Bigg\langle \delta\left[Y_e, X_e\right], \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_e\right]}{T^e\left[X_e\right]}, \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}/e}|X_e\right]\Bigg\rangle\left[Y_e, X_{\mathcal{V}}\right] \\ - \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] \otimes \left\langle \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[Y_e\right]}{T^e\left[Y_e\right]}\right\rangle\left[Y_e\right] .\end{aligned}$$

Proof. By multilinearity of tensor network contractions we have

$$\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right] = \left\langle \{\delta\left[Y_e, X_e\right]\} \cup \{T^{\tilde{e}}\left[X_{\tilde{e}}\right] \,:\, \tilde{e} \neq e\}\right\rangle\left[Y_e, X_{\mathcal{V}}\right]$$

and thus

$$\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right] = \left\langle \{\delta\left[Y_e, X_e\right]\} \cup \{T^{\tilde{e}}\left[X_{\tilde{e}}\right] \,:\, \tilde{e} \neq e\}\right\rangle\left[Y_e\right] .$$

Using both we get

$$\begin{aligned}\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] &= \frac{\partial}{\partial T^e\left[Y_e\right]}\frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right]}{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]} \\ &= \frac{\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right]}{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]} - \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right]\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}{\left(\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]\right)^2} \\ &= \frac{\left\langle \{\delta\left[Y_e, X_e\right]\} \cup \{T^{\tilde{e}}\left[X_{\tilde{e}}\right] \,:\, \tilde{e} \neq e\}\right\rangle\left[Y_e, X_{\mathcal{V}}\right]}{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]} \\ &\quad - \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] \cdot \frac{\left\langle \{\delta\left[Y_e, X_e\right]\} \cup \{T^{\tilde{e}}\left[X_{\tilde{e}}\right] \,:\, \tilde{e} \neq e\}\right\rangle\left[Y_e\right]}{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]} \qquad (10.2)\end{aligned}$$

For the first term we get with a normation equation (see Theorem 10.12) that

$$\begin{aligned}\frac{\left\langle \{\delta\left[Y_e, X_e\right]\} \cup \{T^{\tilde{e}}\left[X_{\tilde{e}}\right] \,:\, \tilde{e} \neq e\}\right\rangle\left[Y_e, X_{\mathcal{V}}\right]}{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle[\varnothing]} &= \frac{\left\langle \{\delta\left[Y_e, X_e\right]\} \cup \{T^{\tilde{e}}\left[X_{\tilde{e}}\right] \,:\, \tilde{e} \in \mathcal{E}\}\right\rangle\left[Y_e, X_{\mathcal{V}}\right]}{T^e\left[X_e\right] \cdot \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle[\varnothing]} \\ &= \frac{\left\langle \delta\left[Y_e, X_e\right], \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]\right\rangle\left[Y_e, X_{\mathcal{V}}\right]}{T^e\left[X_e\right]} \\ &= \frac{\left\langle \delta\left[Y_e, X_e\right], \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_e|\varnothing\right], \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}/e}|X_e\right]\right\rangle\left[Y_e, X_{\mathcal{V}}\right]}{T^e\left[X_e\right]}.\end{aligned}$$

Analogously, we have

$$\frac{\left\langle \{\delta\left[Y_e, X_e\right]\} \cup \{T^{\tilde{e}}\left[X_{\tilde{e}}\right] \,:\, \tilde{e} \neq e\}\right\rangle\left[Y_e\right]}{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle[\varnothing]} = \frac{\left\langle \delta\left[Y_e, X_e\right], \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_e|\varnothing\right]\right\rangle\left[Y_e\right]}{T^e\left[X_e\right]}.$$

With (10.2), we arrive at the claim

$$\begin{aligned}\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] = &\left\langle \delta\left[Y_e, X_e\right], \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_e\right]}{T^e\left[X_e\right]}, \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}/e}|X_e\right]\right\rangle\left[Y_e, X_{\mathcal{V}}\right] \\ &- \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] \otimes \left\langle \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[Y_e\right]}{T^e\left[Y_e\right]}\right\rangle\left[Y_e\right].\end{aligned}$$

■

Lemma 10.18 *For any function $f(T^e)[X_{\mathcal{V}}]$ we have*

$$\begin{aligned}&\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], f(T^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] \\ &\quad= \frac{\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_e = x_e|\varnothing\right]}{T^e\left[X_e = x_e\right]}\Big(\left\langle \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}/e}|X_e = x_e\right], f(T^e)[X_{\mathcal{V}}, Y_e]\right\rangle[\varnothing] \\ &\qquad\qquad - \left\langle \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], f(T^e)[X_{\mathcal{V}}]\right\rangle[\varnothing]\Big) \\ &\qquad + \left\langle \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] \frac{\partial f(T^e)[X_{\mathcal{V}}]}{\partial T^e\left[Y_e\right]}\right\rangle[\varnothing]\end{aligned}$$

Proof. By product rule of differentiation we have

$$\begin{aligned}\frac{\partial}{\partial T^e\left[X_e = x_e\right]}\left\langle \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], f(T^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] = &\left\langle \frac{\partial}{\partial T^e\left[X_e = x_e\right]}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], f(T^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] \\ &+ \left\langle \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], \frac{\partial}{\partial T^e\left[X_e = x_e\right]} f(T^e)[X_{\mathcal{V}}]\right\rangle[\varnothing].\end{aligned}$$

The claim now follows with the application of Lem. 10.17 on the first term. ■

10.6 Discussion

Representations of linear maps is the typical application of tensors, reason for refering to tensor networks as multilinear algebra.

11

Basis Calculus

Basis Calculus stores informations in the selection of basis elements, while coordinate calculus uses the coordinates to each index for storage. While coordinate calculus is more expressive, basis calculus can be exploited in sparse representations of composed functions.

We frequently worked in Part I and Part II with tensors, which have non-negative coordinates and occasionally are boolean (see Def. 0.16) or directed (see Def. 0.17). While boolean tensors have appeared as semantical representation of formulas, directed tensors have appeared mostly as conditional distributions. In this chapter we provide further insights into the situation, where tensors satisfy both. For a schematic depiction of this see Figure 11.1.

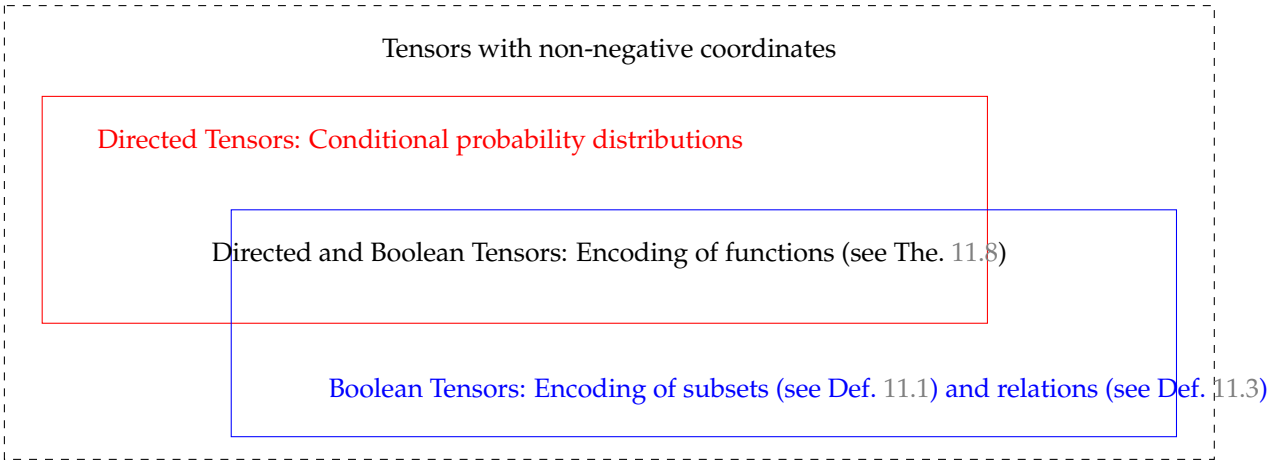


Figure 11.1: Sketch of the tensors with non-negative coordinates. We investigate in this chapter tensors, which are directed and boolean.

11.1 Encoding of Subsets and Relations

Based on the concept of one-hot encodings of states we in this chapter develop the construction of encodings to sets, relations and functions. We start with the definition of subset encodings, which represent set memberships in their boolean coordinates.

Definition 11.1 (Subset Encoding) We say that an arbitrary set $\mathcal{U}$ is enumerated by an enumeration variable $O_{\mathcal{U}}$ taking values in $[r_{\mathcal{U}}]$, when $r_{\mathcal{U}} = |\mathcal{U}|$ and there is a bijective index interpretation function

$$I : [r_{\mathcal{U}}] \to \mathcal{U} \,.$$

Given an set $\mathcal{U}$ enumerated by the variable $O_{\mathcal{U}}$, any subset $\mathcal{V} \subset \mathcal{U}$ is encoded by the tensor

$e_{\mathcal{V}}[O]$ defined for $o \in [|\mathcal{U}|]$ as

$$e_{\mathcal{V}}\left[O = o\right] = \begin{cases} 1 & \text{if} I(o) \in \mathcal{V} \\ 0 & \text{else} \end{cases} .$$

In a one-hot basis decomposition we have

$$e_{\mathcal{V}}\left[O\right] \coloneqq \sum_{o \in [|\mathcal{U}|] \,:\, I(o) \in \mathcal{V}} e_o\left[O\right] .$$

Since relations are subsets of cartesian products between two sets, their encoding is a straightforward generalization of Def. 11.1.

Definition 11.2 (Relation Encoding) A relation between two finite sets $\mathcal{U}^{\text{in}}$ and $\mathcal{U}^{\text{out}}$ is a subset of their cartesian product

$$\mathcal{R} \subset \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{out}} .$$

Given an enumeration of $\mathcal{U}^{\text{in}}$ and $\mathcal{U}^{\text{out}}$ by the categorical variables O_{in} and O_{out} and interpretation maps I_{in}, I_{out}, we define the encoding of this subset as the tensor $e_{\mathcal{R}}[O_{\text{in}}, O_{\text{out}}]$ with the coordinates

$$e_{\mathcal{R}}\left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}} = o_{\text{out}}\right] = \begin{cases} 1 & \text{if } (I_{\text{in}}(o_{\text{in}}), I_{\text{out}}(o_{\text{out}})) \in \mathcal{R} \\ 0 & \text{else} \end{cases} .$$

The relation encoding has a decomposition into one-hot encodings as

$$e_{\mathcal{R}}\left[O_{\text{in}}, O_{\text{out}}\right] = \sum_{o_{\text{in}}, o_{\text{out}} \,:\, (I_{\text{in}}(o_{\text{in}}), I_{\text{out}}(o_{\text{out}})) \in \mathcal{R}} e_{o_{\text{in}}}\left[O_{\text{in}}\right] \otimes e_{o_{\text{out}}}\left[O_{\text{out}}\right] .$$

Relational encodings have a matrix structure by the cartesian product, which can be further folded to tensors, when the sets itself are cartesian products. The relational encoding is a bijection between the relations of two sets and the boolean tensors with their enumeration variables.

11.1.1 Higher order relations

We can extend this contraction to relations of higher order, and arrive at encoding schemes usable for relational databases.

Definition 11.3 Given sets $\mathcal{U}^k$ for $k \in [d]$, a d-ary relation is a subset of a their cartesian product, that is

$$\mathcal{R} \subset \bigtimes_{k \in [d]} \mathcal{U}^k .$$

Given an enumeration of each set $\mathcal{U}^k$ by a variable O_k and an interpretation map I_k, we define the encoding of the relation as the tensor $e_{\mathcal{R}}[O_{[d]}]$ with coordinates

$$e_{\mathcal{R}}\left[O_{[d]} = o_{[d]}\right] = \begin{cases} 1 & \text{if} \quad (I_0(o_0), \ldots, I_{d-1}(o_{d-1})) \in \mathcal{R} \\ 0 & \text{else} \end{cases} .$$

Example 11.4 (Propositional Formulas) Let there be for $k \in [d]$ sets $\mathcal{U}^k$ of truth assignments to the k-th atom, which are all enumerated by $[2]$. A propositional formula then corresponds with a

d-ary relation and we directly defined them in Def. 3.2 by their relational encoding. ◊

Example 11.5 (Relational Databases) Relational Databases can be encoded as tensors using the relation encoding scheme. Each column is thereby understood as an eunumeration variable, which values form the sets $\mathcal{U}^k$. ◊

Let us notice, that the dimensionality of the tensor space used for representing a relation is

$$\prod_{k\in[d]} \left|\mathcal{U}^k\right|$$

and therefore growing exponentially with the number of variables. Relations are however often sparse, in the sense that

$$|\mathcal{R}| << \prod_{k\in[d]} \left|\mathcal{U}^k\right| .$$

It is therefore often benefitially to choose sparse encoding schemes, for example by restricted CP formats (see Chapter 12) to represent $e_{\mathcal{R}}$.

11.2 Encoding of Functions

Let us now restrict to relations, which have an expression by functions. We in this section then show, how contractions of their encodings can be exploited in function evaluation.

11.2.1 Relational Encoding of Functions

Definition 11.6 (Relational Encoding of Maps) Any map

$$f : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$$

can be represented by a relation

$$\mathcal{R}^f := \left\{(x, f(x) \,:\, x \in \mathcal{U}^{\text{in}})\right\} \subset \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{out}} .$$

Given a enumeration of the sets by O_{in} and O_{out} we define the relational encoding of f as the tensor

$$\rho^f\left[O_{\text{in}}, O_{\text{out}}\right] = e_{\mathcal{R}^f}\left[O_{\text{in}}, O_{\text{out}}\right] .$$

Remark 11.7 (Reduction to images) When f maps into a set of infinite cardinality, we restrict $\mathcal{U}^{\text{out}}$ to the image of f and enumerate the image by a variable O_f. This scheme is applied, when f is itself a tensor, i.e. $\mathcal{U}^{\text{out}} = \mathbb{R}$. While the variable O_f can in general be of the same cardinality as the domain set $\mathcal{U}^{\text{in}}$, it will be valued in $[2]$ when considering boolean tensors. ◊

We notice, that any relational representation of a function is also a directed tensor with incoming variables to the domain and outgoing variables to the image. It furthermore holds, that the set of directed and boolean tensors is characterized by the relational encoding of functions. This is shown in the next theorem, by the claim that any boolean tensor which is directed is the relational representation of a function.

Theorem 11.8 *Let $\mathcal{U}^{\text{in}}, \mathcal{U}^{\text{out}}$ be sets and $\mathcal{R} \subset \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{out}}$ a relation. If and only if there exists a map $f : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ such that $\mathcal{R} = \mathcal{R}^f$, the relational encoding ρ^f is a directed tensor with O_{in}*

incoming and O_{out} outgoing.

Proof. "⇒": When f is a function, we have for any $o_{\text{in}} \in [r_{\text{in}}]$

$$\sum_{o_{\text{out}} \in [r_{\text{out}}]} \rho^f \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}} = o_{\text{out}}\right] = \rho^f \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}} = I_{\text{out}}^{-1}(f(I_{\text{in}}(o_{\text{in}})))\right] = 1\,.$$

Thus, $\rho^f \left[O_{\text{out}}, O_{\text{in}}\right]$ is a directed tensor with variables O_{in} incoming and O_{out} outgoing.

"⇐": Conversely let there be a relation $\mathcal{R}$, such that $\rho^{\mathcal{R}}$ is directed. To this end, we observe that for any $o_{\text{in}} \in [r_{\text{in}}]$ the tensor

$$e_{\mathcal{R}} \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}\right]$$

is a boolean tensor with coordinate sum one and therefore a basis vector. It follows that the function $f : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ defined for $x \in \mathcal{U}^{\text{in}}$ as

$$f(x) = I_{\text{out}}(e^{-1}(e_{\mathcal{R}} \left[O_{\text{in}} = I_{\text{in}}(x), O_{\text{out}}\right]))$$

is well-defined. We then have by construction

$$\begin{aligned}
\rho^f \left[O_{\text{out}}, O_{\text{in}}\right] &= \sum_{o_{\text{in}} \in [r_{\text{in}}]} e_{f(o_{\text{in}})} \left[O_{\text{out}}\right] \otimes e_{o_{\text{in}}} \left[O_{\text{in}}\right] \\
&= \sum_{o_{\text{in}} \in [r_{\text{in}}]} e_{\mathcal{R}} \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}\right] \otimes e_{o_{\text{in}}} \left[O_{\text{in}}\right] \\
&= e_{\mathcal{R}} \left[O_{\text{out}}, O_{\text{in}}\right]
\end{aligned}$$

and therefore by Def. 11.6 $\mathcal{R} = \mathcal{R}^f$. ∎

We are specially interested in sets of states of a factored system, which amounts to the case in Def. 0.19. Those state sets have a decomposition into a cartesian product of d sets

$$\mathcal{U} = \bigtimes_{k \in [d]} [m_k]\,.$$

The most obvious enumeration of the set $\mathcal{U}$ is therefore by the collection of state variables $\{X_k \ : \ k \in [d]\}$. Functions between states of factored systems with d_{in} and d_{out} state variables can be represented by $d_{\text{in}} + d_{\text{out}}$-ary relations and Def. 11.6 has an obvious generalization to this case with multiple enumeration variables.

11.2.2 Function Evaluation

We now justify the nomenclature of basis calculus, by showing that contraction with basis elements produce the one-hot encoded function evaluation.

Theorem 11.9 (Function evaluation in Basis Calculus) *Retrieving the value of the function f at a specific state is then the contraction of the tensor representation with the one-hot encoded state. For any $u \in \mathcal{U}^{\text{in}}$ we have*

$$e_{I_{\text{out}}^{-1}(f(u))} \left[O_{\text{out}}\right] = \left\langle \rho^f \left[O_{\text{out}}, O_{\text{in}}\right], e_{I_{\text{in}}(u)} \left[O_{\text{in}}\right] \right\rangle \left[O_{\text{out}}\right]\,.$$

Thus, we can retrieve the function evaluation by the inverse one-hot mapping as

$$f(u) = e^{-1}(\left\langle \rho^f \left[O_{\text{out}}, O_{\text{in}}\right], e_{I_{\text{in}}(u)} \left[O_{\text{in}}\right] \right\rangle \left[O_{\text{out}}\right])\,.$$

Proof. From the representation

$$\rho^f\left[O_{\text{out}}, O_{\text{in}}\right] = \sum_{o_{\text{in}} \in [r_{\text{in}}]} e_{(I_{\text{out}}^{-1} \circ f \circ I_{\text{in}}) o_{\text{in}}}\left[O_{\text{in}}\right] \otimes e_{o_{\text{in}}}\left[O_{\text{in}}\right]$$

and the orthonormality of the one-hot encodings of the input enumeration we get

$$\left\langle \rho^f\left[O_{\text{out}}, O_{\text{in}}\right], e_{I_{\text{in}}(u)}\left[O_{\text{in}}\right] \right\rangle \left[O_{\text{out}}\right] = e_{I_{\text{out}}^{-1}(f(u))}\left[O_{\text{out}}\right] .$$

■

In comparison with the Coordinate Calculus scheme (see The. 10.3), the Basis Calculus produces basis vectors of a functions evaluation instead of scalars. While this seems to produce unnecessary redundancy in representing a function, we will see in the following section, that this scheme is efficient in representing compositions of functions.

11.3 Calculus of relational encodings

We now show the utility of relational encodings for functions, by developing tensor network representation to composed functions. We in this section use the notation of factored system representation, as developed in Part I and enumerate states of factored systems by variables X with states in $[m]$, instead of combinations of variables O with index interpretation functions I enumerating arbitrary sets.

11.3.1 Composition of function

We have already used (see The. 3.7), that combination of propositional formulas by connectives can be represented by contractions. We now show in a more general perspective, that in basis calculus, any composition of functions in its relational encoding the contraction of the encoded functions.

Theorem 11.10 (Composition of Functions) *Let there be two maps between factored systems*

$$f : \bigtimes_{v \in \mathcal{V}_1} [m_v] \to \bigtimes_{v \in \mathcal{V}_2} [m_v]$$

and

$$g : \bigtimes_{v \in \mathcal{V}_2} [m_v] \to \bigtimes_{v \in \mathcal{V}_3} [m_v]$$

with the image system of f is the domain system of g. Then the relational encoding of the composition

$$g \circ f : \bigtimes_{v \in \mathcal{V}_1} [m_v] \to \bigtimes_{v \in \mathcal{V}_3} [m_v]$$

is the contraction

$$\rho^{g \circ f}\left[X_{\mathcal{V}_3}, X_{\mathcal{V}_1}\right] = \left\langle \rho^g\left[X_{\mathcal{V}_3}, X_{\mathcal{V}_2}\right], \rho^f\left[X_{\mathcal{V}_2}, X_{\mathcal{V}_1}\right], \right\rangle \left[X_{\mathcal{V}_3}, X_{\mathcal{V}_1}\right] .$$

Proof. By definition we have the relational encoding of the composition as

$$\rho^{g \circ f}\left[X_{\mathcal{V}_3}, X_{\mathcal{V}_1}\right] = \sum_{x_{\mathcal{V}_1} \in \times_{v \in \mathcal{V}_1} [m_v]} e_{(g \circ f)(x_{\mathcal{V}_1})}\left[X_{\mathcal{V}_3}\right] \otimes e_{x_{\mathcal{V}_1}}\left[X_{\mathcal{V}_1}\right] .$$

By using a similar representation for ρ^g and ρ^f we now show, that this coincides with the contraction of these relational encodings with closed variables $X_{\mathcal{V}_2}$. By the linearity of the contraction operation we get

$$\begin{aligned}\left\langle \rho^f, \rho^g \right\rangle \left[X_{\mathcal{V}_3}, X_{\mathcal{V}_1}\right] &= \sum_{x_{\mathcal{V}_1} \in \bigtimes_{v\in\mathcal{V}_1}[m_v]} \sum_{x_{\mathcal{V}_2} \in \bigtimes_{v\in\mathcal{V}_2}[m_v]} \left\langle \left(e_{g(x_{\mathcal{V}_2})}\left[X_{\mathcal{V}_3}\right] \otimes e_{x_{\mathcal{V}_2}}\left[X_{\mathcal{V}_2}\right]\right), \right. \\ &\qquad \left. \left(e_{f(x_{\mathcal{V}_1})}\left[X_{\mathcal{V}_2}\right] \otimes e_{x_{\mathcal{V}_1}}\left[X_{\mathcal{V}_1}\right]\right) \right\rangle \left[X_{\mathcal{V}_3}, X_{\mathcal{V}_1}\right] \\ &= \sum_{x_{\mathcal{V}_1} \in \bigtimes_{v\in\mathcal{V}_1}[m_v]} \delta_{x_{\mathcal{V}_2}, x_{\mathcal{V}_1}} \cdot e_{g(x_{\mathcal{V}_2})}\left[X_{\mathcal{V}_3}\right] \otimes e_{x_{\mathcal{V}_1}}\left[X_{\mathcal{V}_1}\right] \\ &= \sum_{x_{\mathcal{V}_1} \in \bigtimes_{v\in\mathcal{V}_1}[m_v]} e_{(g\circ f)(x_{\mathcal{V}_1})}\left[X_{\mathcal{V}_3}\right] \otimes e_{x_{\mathcal{V}_1}}\left[X_{\mathcal{V}_1}\right] \\ &= \rho^{g\circ f}\left[X_{\mathcal{V}_3}, X_{\mathcal{V}_1}\right],\end{aligned}$$

where we exploited the orthonormality of the one-hot encodings to the states of $X_{\mathcal{V}_2}$, which contraction thus results in the delta symbol δ applied on the respective states. ∎

We can use The. 11.10 iteratively to further decompose the function g. In this way, the relational encoding of a function consistent of multiple compositions can be represented as the contractions of all the functions. This has been applied in The. 3.7 to efficiently represent propositional formulas, for which syntactical expressions are given.

11.3.2 Compositions with real functions

We here investigate how the composition of a tensor

$$T : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R}$$

with arbitrary functions

$$h : \mathbb{R} \to \mathbb{R}$$

can be represented. This is for example relevant, when representing coordinatewise tensor transforms (see Sect. 10.2) based on tensor network contractions. To this end we understand the tensor $T\left[X_{[d]}\right]$ as a map of the states $\bigtimes_{k\in[d]}[m_k]$ onto its by a variable O_T and index interpretation I enumerated image $\mathrm{im}\,(T)$. We then define the restriction of h onto $\mathrm{im}\,(T)$ as the tensor $h|_{\mathrm{im}(T)}\left[O_T\right]$ with coordinates o_T

$$h|_{\mathrm{im}(T)}\left[O_T = o_T\right] = (h \circ I)(o_T)\,.$$

Let us now show, how contractions with these vectors represents compositions with tensors.

Theorem 11.11 *The coordinatewise transform of any tensor T (see Def. 10.5) by a real function h is the contraction (see Figure 11.2)*

$$h(T)[X_{[d]}] = \left\langle \rho^T\left[O_T, X_{[d]}\right], h|_{\mathrm{im}(T)}\left[O_T\right]\right\rangle \left[X_{[d]}\right].$$

Proof. By the basis calculus The. 11.9 we have for any state $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$, that

$$\left\langle \rho^T\left[O_T, X_{[d]}\right], h|_{\mathrm{im}(T)}\left[O_T\right]\right\rangle \left[X_{[d]} = x_{[d]}\right] = \left\langle \rho^T\left[O_T, X_{[d]} = x_{[d]}\right], h|_{\mathrm{im}(T)}\left[O_T\right]\right\rangle [\varnothing]$$

$$= \left\langle e_{I_T\left[X_{[d]}=x_{[d]}\right]}\left[O_T\right], h|_{\mathrm{im}(T)}\left[O_T\right]\right\rangle[\varnothing]$$
$$= h(T)[X_{[d]} = x_{[d]}]\,.$$

Since both tensors coincide on all coordinates, they are equal. ■

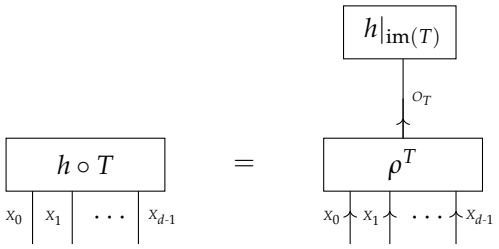


Figure 11.2: Representation of the composition of a tensor T with a real function h.

Corollary 11.12 *For any tensor* $T\left[X_{[d]}\right]$ *we have*

$$T\left[X_{[d]}\right] = \left\langle \rho^T\left[O_T, X_{[d]}\right], \mathrm{Id}|_{\mathrm{im}(T)}\left[O_T\right]\right\rangle\left[X_{[d]}\right]\,.$$

Proof. This follows from The. 11.11 using $h = \mathrm{Id}$ and by noticing that

$$T\left[X_{[d]}\right] = \mathrm{Id}(T)[X_{[d]}]\,.$$

■

Corollary 11.13 *For any tensor* T*, which is directed with* $X_{[d]}$ *incoming, we have*

$$\mathbb{I}\left[X_{[d]}\right] = \left\langle \rho^T\right\rangle\left[X_{[d]}\right]\,.$$

Proof. This follows from The. 11.11 using $h = \mathbb{I}$ and by noticing that

$$\mathbb{I}\left[X_{[d]}\right] = \mathbb{I}(T)[X_{[d]}]\,.$$

■

11.3.3 Decomposition in case of structured images

When a set is structured as the cartesian product of other sets, that is

$$\mathcal{U}^{\mathrm{out}} = \bigtimes_{k\in[d]} \mathcal{U}^k\,,$$

we can enumerate it by a collection $\{O_k : k \in [d]\}$ of enumeration variables, each with respective index interpretation maps. When the image of a function admits such a cartesian representation, we now show that the relational encoding can be represented by a contraction of relational encodings to each image coordinate.

Theorem 11.14 *Let f be a function between factored systems*

$$f : [m] \to \bigtimes_{k\in[d]} [m_k]$$

and denote by

$$f^k : [m] \to [m_k]$$

the image coordinate restrictions of f, that is we have $f = (f^0, \dots, f^{d-1})$. Let us assign the variable X to the factored system in the domain system of f and the variables X_k for $k \in [d]$ to the image system of f. We can then decompose the relational encoding of f into the relational encodings of its image coordinate restrictions, that is

$$\rho^f \left[X_{[d]}, X\right] = \left\langle \{\rho^{f^k} [X_k, X] : k \in [d]\} \right\rangle \left[X_{[d]}, X\right] .$$

Proof. For any $x \in [m]$ we have

$$\begin{aligned}
\rho^f \left[X_{[d]}, X = x\right] &= e_{f(x)} \left[X_{[d]}\right] \\
&= \bigotimes_{k\in[d]} \rho^{f^k} [X_k, X = x] \\
&= \left\langle \{\rho^{f^k} [X_k, X = x] : k \in [d]\} \right\rangle \left[X_{[d]}\right] \\
&= \left\langle \{\rho^{f^k} [X_k, X] : k \in [d]\} \right\rangle \left[X_{[d]}, X = x\right]
\end{aligned}$$

and therefore equality of both tensors. ∎

In Chapter **??** we will apply The. 11.14 in The. 12.20 to show sparse basis CP decompositions to ρ^f. These decompositions are then applied for efficient the representation of empirical distribution, which involve the relational encoding of data maps (see Example 12.21), and for exponential families, which statistics have images, which are included in cartesian products of the images to each coordinate (see Example 12.22).

11.4 Selection Encodings

Selection encodings as introduced in Def. 0.20 are best understood in terms of linear mapping interpretations of tensors. We will first provide by basis encodings a generic relation between the coordinatewise tensor definitions in this work and linear maps.

We then show the utility of this perspective in the representation of composed linear functions. The results are applicable in the exponential family theory, in the tensor representation of energies and means.

11.4.1 Basis encodings of linear maps

Basis encodings are standard linear algebra tools, where matrices are understood as linear maps between vector spaces.

The state sets $\bigtimes_{k\in[d]} [m_k]$ can be interpreted as an enumeration of basis elements e_x of the tensor space $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$.

Along this interpretation, tensors have an interpretation as maps between tensor spaces.

Any tensor and any partition of its variables into two sets can be interpreted as the basis elements of a linear map between the tensor spaces of the respective variables.

Tensor valued functions on state sets $\bigtimes_{k\in[d]}[m_k]$ are an intermediate representation.

Definition 11.15 Let there be two tensor spaces V_1 and V_2 with basis by sets $\mathcal{U}^1 \subset V_1$ and $\mathcal{U}^2 \subset V_2$ of cardinality r_1 and r_2, which are enumerated by variables O_1, O_2 and index interpretation functions I_1, I_2. The basis encoding of any linear map $F \in \mathbb{L}(V_1, V_2)$ is then the tensor

$$\beta^f\left[O_1, O_2\right] \in \mathbb{R}^{r_1} \otimes \mathbb{R}^{r_2}$$

defined for $o_1 \in [r_1]$ and $o_2 \in [r_2]$ by

$$\beta^F\left[O_1 = o_1, O_2 = o_2\right] = \left\langle F^{I_1(o_1)}, I_2(o_2)\right\rangle[\varnothing]\ .$$

Basis encodings for compositions of linear functions can be computed via contractions of the respective basis encodings, as we show next.

Theorem 11.16 *If F^1 is a linear function between V_1 and V_2 and F^2 between V_2 and V_3, and let O_1, O_2 and O_3 be enumerations of orthonormal bases in the spaces with index interpretation functions I_1, I_2 and I_3. We have*

$$\beta^{F^2\circ F^1}\left[O_1, O_3\right] = \left\langle \beta^{F^2}\left[O_2, O_3\right], \beta^{F^1}\left[O_1, O_2\right]\right\rangle\left[O_1, O_3\right]\ .$$

Proof. For arbitrary $o_1 \in [r_1]$ and $o_3 \in [r_3]$ we have to show that

$$\beta^{F^2\circ F^1}\left[O_1 = o_1, O_3 = o_3\right] = \left\langle \beta^{F^2}\left[O_2, O_3 = o_3\right], \beta^{F^1}\left[O_1 = o_1, O_2\right]\right\rangle[\varnothing]\ .$$

By definition we have

$$\beta^{F^2\circ F^1}\left[O_1 = o_1, O_3 = o_3\right] = \left\langle F^2 \circ F^1(I_1(o_1)), I_3(o_3)\right\rangle[\,]\,.$$

Decomposing the linear maps using their basis encoding we get

$$\left\langle F^2 \circ F^1(I_1(o_1)), I_3(o_3)\right\rangle[=]\left\langle F^2\left(\sum_{o_2\in[r_2]}\beta^{F^1}\left[O_1 = o_1, O_2 = o_2\right]\cdot I_2(o_2)\right), I_3(o_3)\right\rangle[=]\sum_{o_2\in[r_2]}\left\langle F^2\left(\beta^{F^1}\left[O_1 = o_1, O_2 = o\right.\right.\right.$$

Therefore, both tensors are equivalent. ∎

For basis encodings we thus have a similar composition theorem as for relational encodings of arbitrary functions (see The. 11.10). What is more, one can understand each relational encodings as a basis encoding of a linear function. Along this line, the composition theorem The. 11.16 as the principle of linear algebra, which underlies The. 11.10. A typical interpretation of The. 11.16 is matrix multiplication, where matrices understood since matrices are basis encodings of linear maps.

11.4.2 Selection encodings as basis encodings

Selection encodings (see Def. 0.20) are related to basis encodings of linear maps as we show in the next theorem.

Theorem 11.17 *Let there be tensor spaces $\bigtimes_{k\in[d]}[m_k]$ and $\bigotimes_{s\in[n]}\mathbb{R}^{p_s}$ with basis by the one-hot*

encodings, enumerated by the categorical variables $X_{[d]}$ and $L_{[n]}$ with index interpretation functions by the one-hot map e. Given a function

$$f : \bigtimes_{k\in[d]} [m_k] \to \bigotimes_{s\in[n]} \mathbb{R}^{p_s}$$

we define a linear map $F^f \in \mathbb{L}(\bigotimes_{k\in[d]} \mathbb{R}^{m_k}, \bigotimes_{s\in[n]} \mathbb{R}^{p_s})$ by the action on the basis elements to $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ as the tensors

$$F^f(e_{x_{[d]}}) := f(x_{[d]})$$

carrying the variables $L_{[n]}$. We then have

$$\gamma^f \left[X_{[d]}, L_{[n]}\right] = \beta^{F^f} \left[X_{[d]}, L_{[n]}\right] .$$

Proof. We show equality on each slice with respect to the variables $X_{[d]}$ and therefore choose arbitrary $x_{[d]}$. It holds by definition of selection encodings and the map F^f that

$$\gamma^f \left[X_{[d]} = x_{[d]}, L_0, \ldots, L_{n-1}\right] = f(x_{[d]})[L_{[n]}] = F^f(e_{x_{[d]}})[L_{[n]}] .$$

We further have

$$F^f(e_{x_{[d]}})[L_{[n]}] = \sum_{l_{[n]}} \left\langle F^f(e_{x_{[d]}})[L_{[n]} = l_{[n]}], e_{l_{[n]}} \left[L_{[n]}\right] \right\rangle [\varnothing] \cdot e_{l_{[n]}} \left[L_{[n]}\right] = \sum_{l_{[n]}} \beta^{Ff} \left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]}\right] \cdot e_{l_{[n]}} \left[L_{[n]}\right] = \beta^{F}$$

For arbitrary $x_{[d]}$ the slices of γ^f and β^{F^f} thus coincide, which proofs the equivalence of both tensors. ∎

While relational encoding works for maps from $\bigtimes_{k\in[d]}[m_k]$ to arbitrary sets (which are enumerated), selection encodings as introduced in Def. 0.20 require and exploit that their image is embedded in a tensor space.

Given a selection encoding of a function, the function is retrieved by slicing with respect to the

$$f(x) = \gamma^f [X = x, L] .$$

More generally, we show in the next Lemma how to construct to any tensor and any partition of its variables functions by slicing operations, such that the tensor is the selection encoding of the function.

Theorem 11.18 *Let $T[X_{\mathcal{V}}]$ be a tensor in $\bigotimes_{v\in\mathcal{V}} \mathbb{R}^{m_v}$ and let A, B be a disjoint partition of $\mathcal{V}$, that is $A \dot\cup B = \mathcal{V}$. Then the function*

$$f : \bigtimes_{v\in A} [m_v] \to \bigotimes_{v\in B} \mathbb{R}^{m_v}$$

defined for $x_A \in \bigtimes_{v\in A}[m_v]$ as

$$f(x_A) := T[X_A = x_A, X_B]$$

obeys

$$\gamma^f [X_A, X_B] = T[X_{\mathcal{V}}] ,$$

where we understand the variables X_B as selection variables.

Proof. We have for any x_B that

$$\gamma^f\left[X_A, X_B = x_B\right] = \sum_{x_A \in \bigtimes_{v\in A}[m_v]} e_{x_A}\left[X_A\right] \otimes f(x_A)[X_B = x_B] = \sum_{x_A \in \bigtimes_{v\in A}[m_v]} e_{x_A}\left[X_A\right] \otimes T\left[X_A = x_A, X_B = x_B\right] = T\left[X_A, \right.$$

and the equivalence follows. ∎

Example 11.19 (Markov Logic Networks and Proposal Distributions) While the statistic of MLN (namely $\mathcal{H}$) and the proposal distribution (namely $\mathcal{H}^T$) have a common selection encoding, both result from the inverse selection encoding described in The. 11.18. We can construct $\mathcal{H}^T$ by first building the selection encoding to $\mathcal{H}$ and then applying the construction of The. 11.18 with $A = L$ and $B = X_{[d]}$. ◊

We use selection encodings to represent weighted sums of functions, based on the next theorem.

Theorem 11.20 (Weighted formula sums by selection encodings) *Let ϕ be a tensor valued function from $\bigtimes_{k\in[d]}[m_k]$ to $\mathbb{R}^p$ with image coordinates ϕ_l and let $\theta\left[L\right]$ be a tensor. Then*

$$\left(\sum_{l\in[p]} \theta\left[L = l\right] \cdot \phi_l\right)\left[X_{[d]}\right] : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}$$

is represented as

$$\left(\sum_{l\in[p]} \theta\left[L = l\right] \cdot \phi_l\right)\left[X_{[d]}\right] = \left\langle \gamma^{\phi}\left[X_{[d]}, L\right], \theta\left[L\right]\right\rangle \left[X_{[d]}\right] .$$

Proof. The representation holds, since for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have

$$\left\langle \gamma^{\phi}\left[X_{[d]}, L\right], \theta\left[L\right]\right\rangle \left[X_{[d]} = x_{[d]}\right] = \sum_{l\in[p]} f(L = l) \cdot \phi_l\left[X_{[d]} = x_{[d]}\right] .$$

∎

This theorem shows, that while relation encodings can represent any composition with another function by a contractions, selection encodings can be used to represent linear transforms. To see this, we interpret ϕ and f in Theorem 11.20 as basis decompositions of linear maps.

11.5 Effective Coordinate Calculus

In some situations, we can perform basis calculus more effectively by avoiding image enumeration variables, and instead apply coordinatewise transforms on tensors (see Def. 10.5). As we show here, these include conjunctions, which correspond with coordinatewise multiplication, and negation, which correspond with coordinatewise substraction from 1. Such schemes are applied for example in [TAP24] in batchwise logical inference.

Theorem 11.21 *For any formulas f, h we have*

$$\left\langle \rho^{\wedge}\left[Y_{f\wedge h}, X_f, X_h\right], e_1\left[Y_{f\wedge h}\right]\right\rangle \left[X_f, X_h\right] = e_1\left[X_f\right] \otimes e_1\left[X_h\right] .$$

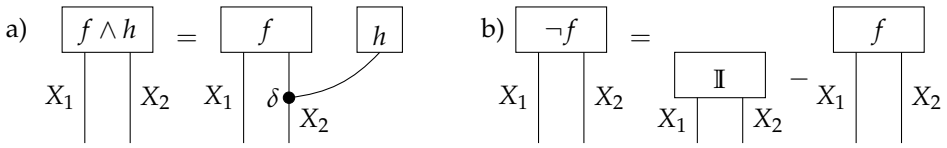


Figure 11.3: Decomposition schemes by effective calculus, using coordinatewise transforms of tensors (see Def. 10.5). a) Conjunction performed by coordinatewise multiplications, b) Negations performed by coordinatewise substraction from one.

In particular, it holds that (see Figure 11.3a)

$$(f \wedge h)[X_{[d]}] = \langle f, h \rangle \left[X_{[d]} \right] .$$

Proof. We decompose

$$\rho^{\wedge} \left[Y_{f\wedge h}, X_f, X_h \right] = e_1 \left[Y_{f\wedge h} \right] \otimes e_1 \left[X_f \right] \otimes e_1 \left[X_h \right] + e_0 \left[Y_{f\wedge h} \right] \left(\mathbb{I} \left[X_f, X_h \right] - e_1 \left[X_f \right] \otimes e_1 \left[X_h \right] \right)$$

and get the first claim as

$$\begin{aligned} \left\langle \rho^{\wedge} \left[Y_{f\wedge h}, X_f, X_h \right], e_1 \left[Y_{f\wedge h} \right] \right\rangle \left[X_f, X_h \right] &= \left\langle e_1 \left[Y_{f\wedge h} \right] \otimes e_1 \left[X_f \right] \otimes e_1 \left[X_h \right], e_1 \left[Y_{f\wedge h} \right] \right\rangle \left[X_f, X_h \right] \\ &= e_1 \left[X_f \right] \otimes e_1 \left[X_h \right] . \end{aligned}$$

To show the second claim we use

$$\begin{aligned} (f \wedge h)[X_{[d]}] &= \left\langle \rho^{f} \left[X_f, X_{[d]} \right], \rho^{h} \left[X_h, X_{[d]} \right], \rho^{\wedge} \left[Y_{f\wedge h}, X_f, X_h \right], e_1 \left[Y_{f\wedge h} \right] \right\rangle \left[X_{[d]} \right] \\ &= \left\langle \rho^{f} \left[X_f, X_{[d]} \right], \rho^{h} \left[X_h, X_{[d]} \right], (e_1 \left[X_f \right] \otimes e_1 \left[X_h \right]) \right\rangle \left[X_{[d]} \right] \\ &= \langle f, h \rangle \left[X_{[d]} \right] . \end{aligned}$$

■

A similar decomposition holds for negations, as we show next.

Theorem 11.22 *For any formula f we have*

$$\left\langle \rho^{\neg} \left[Y_{\neg f}, X_f \right], e_1 \left[Y_{\neg f} \right] \right\rangle \left[X_f \right] = e_0 \left[X_f \right] = \mathbb{I} \left[X_f \right] - e_1 \left[X_f \right] .$$

and

$$\left\langle \rho^{\neg} \left[X_f, Y_{\neg f} \right], e_0 \left[Y_{\neg f} \right] \right\rangle \left[X_f \right] = e_1 \left[X_f \right] .$$

In particular, it holds that (see Figure 11.3b)

$$(\neg f)[X_{[d]}] = \mathbb{I} \left[X_{[d]} \right] - f \left[X_{[d]} \right] .$$

Proof. Using that for two dimensional variables we have $\mathbb{I}[X] = e_0[X] + e_1[X]$. ■

These theorems provide a mean to represent logical formulas by sums of one-hot encodings. Since any propositional formula can be represented by compositions of negations and conjunctions, they are universal. We further notice, that the resulting decomposition is a basis+ CP format, as further discussed in Chapter 12. In Figure 11.4 we provide an example of this decomposition.

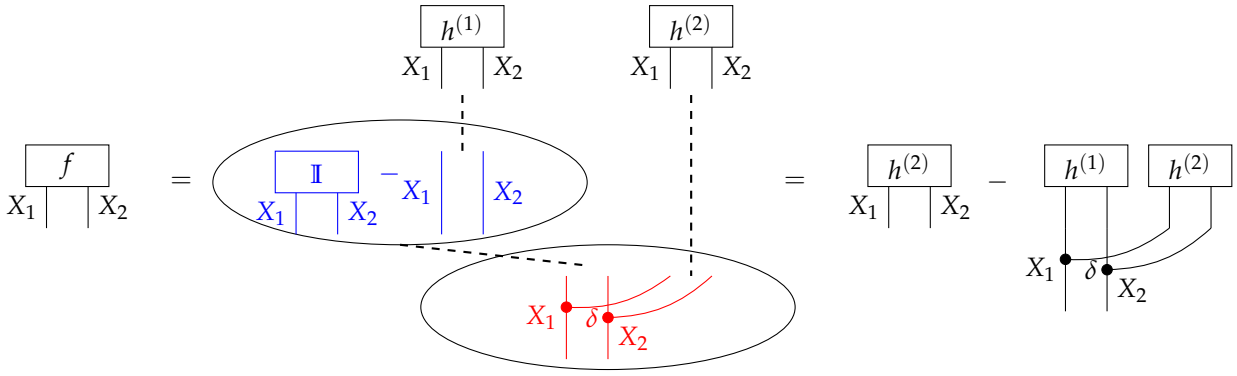


Figure 11.4: Example of a decomposition by effective calculus of a formula $f(X_1, X_2) = \neg h^{(1)}(X_1, X_2) \wedge h^{(2)}(X_1, X_2)$ into a sum of contractions.

In an alternative perspective, effective calculus amounts to an contraction against the directionality of the relational encodings. For specific functions, slices of the relational encodings with respect to head variables are basis vectors. In that case, we can perform basis calculus in the inverse direction than suggested by the directions of the tensors. We examplify this situation in the following theorem for relational encodings of logical conjunctions and negations.

11.6 Applications in Machine Learning

The neural paradigm of Machine Learning describes the relevance of sparse function to be effective models in the sense of learning and approximation.

Our model of the neural paradigm are tensor network decompositions, seen as decomposition of functions into smaller functions, which take each other as input. Summations along input axis are avoided, when having directed and boolean tensor networks with basis calculus interpretation.

We have already observed in Theorem 11.9, that the value of discrete maps can be calculated by contractions of the directed boolean relation encodings. This has been framed as Basis Calculus. What is more, tensor network decompositions into directed boolean tensors correspond with representation of functions as compositions of smaller functions. We can understand each composition as marking a neuron in an architecture and thus have established a neural perspective on boolean directed tensor networks.

12

Sparse Calculus

We in this chapter investigate, which sparsity notations enable tensors to be representable as contractions of tensor networks.

12.1 CP Formats

The CP Decomposition is one way to generalize the ranks of matrices to tensors. It is oriented on the Singular Value Decomposition of matrices, providing a representation of the matrix as a weighed sum of the tensor product of singular vectors. Given a tensor of higher order, each such tensor product is over multiple vectors,

Definition 12.1 A CP Decomposition of rank n of a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ is a collections of tensors $\sigma[I]$ and $V^k[I, X_k]$ for $k \in [d]$, where I takes values in $[n]$, such that

$$T\left[X_{[d]}\right] = \left\langle \{\sigma[I]\} \cup \{V^k[I, X_k] \,:\, k \in [d]\} \right\rangle \left[X_{[d]}\right] .$$

We say that the CP Decomposition is

- directed, when for each k the core V^k is directed with I incoming and X_k outgoing.
- binary, when for each k the core V^k is binary.
- basis, where we demand both properties, that is for each $k \in [d]$ and $i \in [n]$

$$V^k[J = i, X_k] \in \{e_{[x_k]}\,[X_k]\, x_k \in [m_k]\} .$$

- basis+, when for each $k \in [d]$ and $i \in [n]$

$$V^k[J = i, X_k] \in \{e_{[x_k]}\,[X_k]\, x_k \in [m_k]\} \cup \{\mathbb{I}\,[X_k]\} .$$

We denote by $\operatorname{rank}(T)$, respectively $\operatorname{rank}^{\mathrm{bin}}(T)$, $\operatorname{rank}^{\mathrm{bas}}(T)$ and $\operatorname{rank}^{\mathrm{bas}+}(T)$ the minimal cardinality such that T has a CP Decomposition with directed cores, respectively binary cores, basis cores and basis+ cores.

We have by definition

$$T\left[X_{[d]}\right] = \sum_{i\in[n]} \sigma[J = i] \left(\bigotimes_{k\in[d]} V^k[J = i, X_k] \right) .$$

The right side can be seen as an alternative definition of CP Decompositions by summations of elementary tensors.

We introduce different notions of sparsities based on CP Decomposition with different properties of their leg cores.

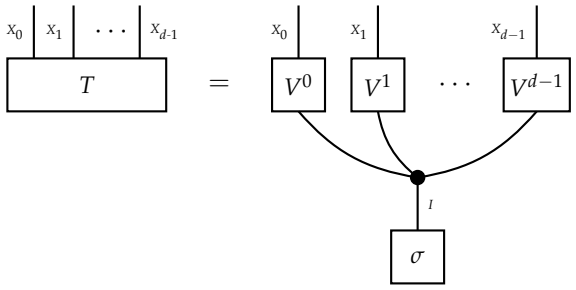


Figure 12.1: Tensor Network diagram of a generic CP decomposition (see Def. 12.1)

12.1.1 Directed Leg Cores

This is the canonical CP Decomposition, where the vectors $V^{k,i}$ are interpreted as generalized singular vectors. Any CP decomposition can be transformed into a directed CP decomposition without enlarging the index set J, simply by diving the vectors by their norms and multiplying it to $\sigma[J = i]$.

We then have a partially directed Tensor Network representing the decomposed tensor. The only undirected core is σ, since we do not demand it to be normed. In many applications applications, however, also the σ is directed with a single outgoing leg (see for example the empirical distributions as discussed in Sect. 1.8). In that case, also the decomposed tensor is directed with outgoing legs.

12.1.2 Basis Leg Cores

Directed and binary leg cores have incoming slices being basis vectors, we thus call them basis CP Decomposition. This allows the interpretation of the directed and binary CP decomposition in terms of mapping to nonzero coordinates. We start by defining the number of nonzero coordinates of tensors by the ℓ_0-norm.

Definition 12.2 The ℓ_0-norm counts the nonzero coordinates of a tensor by

$$\ell_0(T) = \#\left\{x_0, \ldots, x_{d-1} \,:\, T_{x_0,\ldots,x_{d-1}} \neq 0\right\}.$$

The ℓ_0-norm is not a proper norm itself, but the limit of ℓ_p-norms (where $p \to 0$) of the flattened tensor (which are norms for $p \geq 1$).

The ℓ_0 norm is the number of nonzero coordinates. We understand the leg cores as the relational encoding of functions mapping to the slices of these coordinates given an enumeration. This is consistent with the previous analysis of Chapter 11, where we characterized binary and directed cores by the encoding of associated functions. Based on this idea, we can proof, that any tensor has a directed and binary CP decomposition with rand $\ell_0(T)$.

Theorem 12.3 *For any tensor T we have*

$$\operatorname{rank}^{\text{bas}}(T) = \ell_0(T).$$

Proof. We find a map

$$f : [\ell_0(T)] \to \bigtimes_{k\in[d]} [m_k],$$

which image is the set of nonzero coordinates of T. Denoting its image coordinate maps by f^k we

have

$$T = \sum_{j \in [m]} \sigma[f(j)] \left(\bigotimes_{k \in [d]} e_{f^k(j)} \right) .$$

This is a basis CP Decomposition with rank $\ell_0(T)$. Conversely, any basis CP Decomposition of T with dimension r would have at most r coordinates different from zero and thus $\ell_0(T) \leq r$. Thus, there cannot be a CP Decomposition with a dimension $r \leq \ell_0(T)$. ∎

The next theorem relates the basis CP decomposition with encodings of d-ary relations (see Def. 11.3).

Theorem 12.4 *If any only if $T \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ has a basis decomposition with slices $\{x^i_{[d]} : i \in [n]\}$ and trivial cores, it coincides with the encoding of the d-ary relation*

$$\mathcal{R} = \{x^i_{[d]} : i \in [n]\} .$$

If in addition $m_k = 2$, we can interpret basis CP decompositions as propositional formulas.

Theorem 12.5 *If $T \in \bigotimes_{k \in [d]} \mathbb{R}^2$ has a basis decomposition with slices $\{x^i_{[d]} : i \in [n]\}$ and trivial cores, it coincides with the propositional formula*

$$f\left[X_{[d]}\right] = \bigvee_{i \in [n]} Z^{\wedge}_{x^i_{[d]}} .$$

Proof. This is a generalization of The. 6.5, which follows from The. 3.13. ∎

The storage demand of any CP decomposition is at most linear in the dimension and the sum of its leg dimension. When we have a basis CP decomposition, this demand can be further improved. The basis vectors can be stored by its preimage of the one hot encoding *e.*, that is the number of the basis vector in $[m]$. This reduces the storage demand of each basis vector to the logarithms of the space dimension without the need of storing the full vector.

More precisely, we can store the CP Decomposition by the matrix

$$M[I, L] \in \mathbb{R}^{m \times (d+1)}$$

defined for $k \in [d]$

$$M[J = i, L = k] = e^{-1}(V^k[J = i, X_k])$$

and

$$M[J = i, L = d] = \sigma[J = i] .$$

This is a typical tabular format to store relational databases.

12.1.3 Basis+ Leg Cores

The minimal rank of CP Decompositon is closely related to polynomial sparsity of the map T, which we will define next.

Definition 12.6 A monomial decomposition of a tensor $T \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ is a set $\mathcal{M}$ of tuples

(λ, A, x_A) where $\lambda \in \mathbb{R}, A \subset [d]$ and $x_A \in \bigtimes_{k\in A}[m_k]$ such that

$$T\left[X_{[d]}\right] = \sum_{(\lambda,A,x_A)\in\mathcal{M}} \lambda \cdot \langle e_{x_A}\rangle \left[X_{[d]}\right] . \tag{12.1}$$

For any tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we define its polynomial sparsity of order r as

$$\operatorname{rank}^r(T) = \min\left\{ |\mathcal{M}| \,:\, T\left[X_{[d]}\right] = \sum_{(\lambda,A,x_A)\in\mathcal{M}} \lambda \cdot \langle e_{x_A}\rangle \left[X_{[d]}\right] , \forall_{(\lambda,A,x_A)\in\mathcal{M}} |A| \leq r . \right\}$$

We refer to the terms in a decomposition (12.1) in Def. 12.6 as monomials of binary variables, which are enumerated by pairs (k, x_k) and indicate whether the variable X_k is in state $x_k \in [m_k]$. Such indicators are represented by the one-hot encodings

$$e_{x_k}[X_k] .$$

The monomial of multiple such binary variables indicated, whether all variables labelled by a set A are in the state X_A, which is represented by

$$e_{x_A}[X_A] = \bigotimes_{k\in A} e_{x_k}[X_k] .$$

The states of the variables labeled by $k \in [d]/A$ are not specified in the monomial and the monomial is trivially extended to

$$\langle e_{x_A}\rangle \left[X_{[d]}\right] = e_{x_A}[X_A] \otimes \mathbb{I}\left[X_{[d]/A}\right] .$$

For some $r < d$ there are tensors $T\left[X_{[d]}\right]$, which do not have a monomial decomposition of order r. In that case we the minimum is over an empty set and we define $\operatorname{rank}^r(T) = \infty$. We characterize in the next theorem the set of tensors with monomial decompositions.

Theorem 12.7 *For any d, r, the set of tensors of d variables with leg dimension m, which have a monomial decomposition of order r, is a linear subspace $V^{d,r}$ with dimension*

$$\dim(V^{d,r}) \leq \sum_{s\in[r]} m^s \binom{d}{s} .$$

Proof. Any sum of tensors with a monomial decomposition of order r admits again a monomial decomposition, which is the concatenation of both. The same holds for a scalar multiplication, and thus, the sets of such tensors form a linear subspace.

The number of different tensors $\langle e_{x_A}\rangle \left[X_{[d]}\right]$ is

$$\sum_{s\in[r]} m^s \binom{d}{s}$$

Since any tensor with a monomial decomposition is a weighted sum of those,

We notice, that the set of slices is in general not linear independent, and therefore forms a frame and not a linear basis [CKP13]. The number of elements in the frame is therefore an upper bound on the dimension. ∎

The. 12.7 implies, that the tensors admiitting a monomial decomposition of a small order build

a low-dimensional subspace in the m^d dimensional space of tensors, since for $r << d$ we have

$$\dim(V^{d,r}) << m^d \,.$$

If $r \geq d$, we always find a monomial decomposition by an enumeration of nonzero coordinates. In the next theorem, we show that in that case the $\text{rank}^r(T)$ furthermore coincides with the basis+ CP rank $\text{rank}^{\text{bas+}}(T)$.

Theorem 12.8 *For any tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we have*

$$\text{rank}^d(T) = \text{rank}^{\text{bas+}}(T) \,.$$

When $x_k = 2$ for all $k \in [d]$, we also have

$$\text{rank}^{\text{bin}}(T) = \text{rank}^d(T) \,.$$

Proof. To proof the first claim, we construct a basis+ CP decomposition given a monomial decomposition and vice versa. Let there be a tensor $T\left[X_{[d]}\right]$ with a monomial decomposition by $\mathcal{M}$ with $|\mathcal{M}| = m$ and let us enumerate the elements in $\mathcal{M}$ by $(\lambda^i, A^i, x^i_{A^i})$ for $i \in [n]$. We define for each $k \in [d]$ the tensors

$$V^k[I, X_k] = \left(\sum_{i\in[n]\,:\,k\in A} e_i[I] \otimes e_{x^i_k}[X_k]\right) + \left(\sum_{i\in[n]\,:\,k\notin A} e_i[I] \otimes \mathbb{I}[X_k]\right)$$

and

$$\sigma[I] = \sum_{i\in[n]} \lambda^i \cdot e_i[I]$$

and notice that

$$\begin{aligned} T\left[X_{[d]}\right] &= \sum_{i\in[n]} \lambda^i \cdot \left\langle e_{x^i_A}\right\rangle \left[X_{[d]}\right] \\ &= \sum_{i\in[n]} \left(\sigma[J=i] \cdot \bigotimes_{k\in[d]} V^k[J=i, X_k]\right) \\ &= \left\langle \{\sigma[I]\} \cup \{V^k[I, X_k] \,:\, k \in [d]\}\right\rangle \left[X_{[d]}\right] \,. \end{aligned}$$

By construction this is a basis+ CP decomposition with rank n. Since any monomial decomposition can be transformed into a basis+ CP decomposition with same rank we have

$$\text{rank}^d(T) \geq \text{rank}^{\text{bas+}}(T) \,.$$

Let there now be a basis+ CP decomposition we define for each $i \in [n]$

$$A^i = \{k \in [d] : V^k[J=i, X_k] \neq \mathbb{I}[X_k]\} \quad \text{and} \quad x^i_A = \{e^{-1}(V^k[J=i, X_k]) : k \in A\}$$

where by $e^{-1}(\cdot)$ we denote the inverse of the one-hot encoding.

We notice that this is a monomial decomposition of $T\left[X_{[d]}\right]$ to the tuple set

$$\mathcal{M} = \{(\sigma[J=i], A^i, x^i_{A^i}) \,:\, i \in [n]\} \,.$$

It follows from this that

$$\text{rank}^d(T) \leq \text{rank}^{\text{bas}+}(T)$$

and the first claim is shown.

The second claim follows from the observation, that whenever $x_k = 2$ for all $k \in [d]$ the binary CP decompositions with non-vanishing slices $V^k[J = i, X_k]$ for $k \in [d]$ and $i \in [n]$ are also basis+ CP decompositions and vice versa. ■

Example 12.9 (Propositional Formulas) When all leg dimensions of a binary tensor T are 2, we can interpret T as a logical formula. We can use the binary CP decomposition of any tensor $\tilde{T}$ with $\mathbb{I}_{\neq 0}(\tilde{T}) = T$ as a CNF of T. Finding the sparsest CNF thus amounts to finding the $\tilde{T}$ with minimal $\text{rank}^d(\tilde{T})$ such that $\mathbb{I}_{\neq 0}(\tilde{T}) = T$. ♢

12.2 Constructive Bounds on CP Ranks

After having defined three CP Decompositions, let us investigate bounds on their ranks which proofs come with constructions of the cores.

12.2.1 Format Transformations

Especially useful, when the leg dimensions are two, where the slice decomposition shows decomposition of the tensor into monomials.

Theorem 12.10 *For any tensor* $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ *we have*

$$\text{rank}(T) \leq \text{rank}^{\text{bin}}(T) \leq \text{rank}^{\text{bas}+}(T) \leq \text{rank}^{\text{bas}}(T) .$$

Proof. Since any CP decomposition into binary leg cores can be normed to a CP decomposition with directed leg cores, the first bound holds. The second bound holds analogously, since any CP decomposition with basis leg cores is also a CP decomposition with binary leg cores. ■

Consider for example the tensor $\mathbb{I}$ having maximal ℓ_0-norm being the dimension of the tensor space, but, since it is elementary, a CP decomposition with rank 1.

12.2.2 Summation of CP Decompositions

Theorem 12.11 *For any collections of tensors* $\{T^l[X_\mathcal{V}] : l \in [n]\}$ *with identical variables and scalars* $\lambda^l \in \mathbb{R}$ *for* $l \in [n]$ *we have*

$$\text{rank}\left(\sum_{l\in[n]} \lambda^l \cdot T^l\right) \leq \sum_{l\in[n]} \text{rank}\left(T^l\right) .$$

The bound still holds, when we replace on both sides $\text{rank}(\cdot)$ *by* $\text{rank}^{\text{bin}}(\cdot)$*, by* $\text{rank}^{\text{bas}}(\cdot)$ *or by* $\text{rank}^{\text{bas}+}(\cdot)$.

Proof. Products with scalars do not change the rank, since they just rescale the core σ. The sum of CP Decomposition is just the combination of all slices, thus the rank is at most additive. ■

12.2.3 Contractions of CP Decompositions

More general, we can bound the sparsity of any contraction by the product of sparsities of affected tensors.

Theorem 12.12 *For any tensor network with variables $\mathcal{V}$ and edges $\mathcal{E}$ we have for any subset $\tilde{\mathcal{V}} \subset \mathcal{V}$*

$$\operatorname{rank}\left(\left\langle \{T^e : e \in \mathcal{E}\}\right\rangle \left[\tilde{\mathcal{V}}\right]\right) \leq \prod_{e\in\mathcal{E}\,:\,\tilde{\mathcal{V}}\cap e\neq\varnothing} \operatorname{rank}\left(T^e\right) .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$*, by* $\operatorname{rank}^{\text{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas}+}(\cdot)$.

Remarkably, in The. 12.12 the upper bound on the CP rank is build only by the ranks of the tensor cores, which have remaining open edges. We prepare for its proof by first showing the following Lemmata.

Lemma 12.13 *For any tensors $T^1\left[X_{\mathcal{V}_1}\right]$ and $T^2\left[X_{\mathcal{V}_2}\right]$ and any set of variables $\tilde{\mathcal{V}} \subset \mathcal{V}_1 \cup \mathcal{V}_2$ we have*

$$\operatorname{rank}\left(\left\langle \{T^1, T^2\}\right\rangle \left[\tilde{\mathcal{V}}\right]\right) \leq \operatorname{rank}\left(T^1\right) \cdot \operatorname{rank}\left(T^2\right) .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$*, by* $\operatorname{rank}^{\text{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas}+}(\cdot)$.

Proof. By connecting the cores and restoring the binary or basis properties. ∎

When one core of the contracted tensor network does not contain variables which are left open, we can drastically sharpen the bound provided by Lem. 12.13 as we show next.

Lemma 12.14 *For any tensor network consistent of two tensors $T^1\left[X_{\mathcal{V}_1}\right]$ and $T^2\left[X_{\mathcal{V}_2}\right]$ and any set $\tilde{\mathcal{V}}$ with $\tilde{\mathcal{V}} \cap \mathcal{V}_2 = \varnothing$ we have*

$$\operatorname{rank}\left(\left\langle \{T^1, T^2\}\right\rangle \left[\tilde{\mathcal{V}}\right]\right) \leq \operatorname{rank}\left(T^1\right) .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas}}(\cdot)$.

Proof. We show the lemma by constructing a CP decomposition of $\operatorname{rank}\left(\left\langle \{T^1, T^2\}\right\rangle \left[\tilde{\mathcal{V}}\right]\right)$ for any CP decomposition of T^1. Let therefore take any CP decomposition of T^1 consistent of the leg cores $\{V^v : v \in \mathcal{V}_1\}$ and a scalar core σ. Then we define a new σ by

$$\tilde{\sigma} = \left\langle \{\sigma\} \cup \{V^v : v \in \mathcal{V}_1, v \notin \tilde{\mathcal{V}}\} \cup \{T^2\}\right\rangle [I] .$$

Then, the leg cores $\{V^v : v \in \tilde{\mathcal{V}}\}$ build with the scalar core $\tilde{\sigma}$ a CP decomposition of $\left\langle \{T^1, T^2\}\right\rangle \left[\tilde{\mathcal{V}}\right]$. When the CP decomposition of T^1 was binary, basis or basis+, this property is also satisfied by the constructed CP decomposition. Thus the bound also holds for the ranks $\operatorname{rank}^{\text{bin}}(\cdot)$ or $\operatorname{rank}^{\text{bas}}(\cdot)$. ∎

Proof of The. 12.12. Use delta tensor representation to represent contractions by graphs. We then iterate through the cores and contract them to the previously contracted tensor, where we apply Lem. 12.13 when the tensor core has variables left open and Lem. 12.14 if not. ∎

Example 12.15 (Composition of formulas with connectives) For any formula f we have $1 - f = \neg f$. The CP rank bound brings an increase by at most factor 2 when taking the contraction with

$\rho^{\neg}$ which has slice sparsity of 2. This is not optimal, since $\neg f$ has at most an absolute slice sparsity increase of 1.

For any formulas f and h we have $f \cdot h = f \wedge h$. Here the CP rank bounds on contractions can also be further tightened. ◊

Example 12.16 (Distributions of independent variables) Independence means factorization, conditional independence means sum over factorizations. Again, the ℓ_0 norm is bounded by the product of the ℓ_0 norm of the factors. ◊

12.2.4 Normations of CP Decompositions

As a theorem: If any of the above CP Decomposition is normable, the normation has the same CP ranks. Especially interesting when learning Bayesian Networks, where each core has a CP bound by the number of datapoints.

12.2.5 Sparse Encoding of Functions

We now state that the basis CP rank of relational encodings is equal to the cardinality of the domain. The basis CP format can therefore not provide a sparse representation when the factored system contains many categorical variables.

Theorem 12.17 *For any function*

$$f : \bigtimes_{k \in [d]} [m_k] \to \bigtimes_{l \in [r]} [m_l]$$

between factored systems we have

$$\operatorname{rank}^{\text{bas}} \left(\rho^f \right) = \prod_{k \in [d]} m_k \,.$$

Proof. With The. 12.3, the basis CP rank coincides with the number of not vanishing coordinates, which is the cardinality of the domain of f. ■

Allowing for trivial leg vectors can decrease the CP rank, as we show next.

Theorem 12.18 *We have*

$$\operatorname{rank}^{\text{bas+}} \left(\rho^f \right) \leq \sum_{y \in \operatorname{im}(f)} \operatorname{rank}^{\text{bas+}} \left(\mathbb{I}^{f=y} \right) ,$$

where by $\mathbb{I}^{f=y}$ we denote the indicator, whether the function f evaluates to y, that is the tensor

$$\mathbb{I}^{f=y} \left[X_{[d]} \right] = \begin{cases} 1 & \text{if} \quad f(x_{[d]}) = y \\ 0 & \text{else} \,. \end{cases}$$

Proof. We have

$$\rho^f = \sum_{y \in \operatorname{im}(f)} \mathbb{I}^{f=y}[X] \otimes e_{I(y)} \left[Y_f \right] .$$

For each $y \in \operatorname{im}(f)$ we represent $e_{I(y)} \left[Y_f \right]$ in an basis+ CP format with $\operatorname{rank}^{\text{bas+}} \left(\mathbb{I}^{f=y} \right)$ summands and arrive at a basis+ CP decomposition of ρ^f with $\sum_{y \in \operatorname{im}(f)} \operatorname{rank}^{\text{bas+}} \left(\mathbb{I}^{f=y} \right)$ summands. ■

The above claim still holds when replacing $\text{rank}^{\text{bas}+}(\cdot)$ with the ranks $\text{rank}^{\text{bas}}(\cdot)$ or $\text{rank}^{\text{bin}}(\cdot)$. For the rank $\text{rank}^{\text{bas}}(\cdot)$ it leads to the bound of The. 12.17, since summing the number of non zero coordinators of the indicators is the cardinality of the domain.

Example 12.19 Conjunction of variables For the propositional formula $f = X_0 \wedge X_1$ we have

$$\rho^f[X_0, X_1] = e_{1,1}[X_0, X_1] \otimes e_1\left[X_f\right] + (\mathbb{I}[X_0, X_1] - e_{1,1}[X_0, X_1]) \otimes e_0\left[X_f\right]$$

and thus

$$\text{rank}^{\text{bas}+}\left(\rho^f\right) \leq 3$$

while $\text{rank}^{\text{bas}}\left(\rho^f\right) = 4$. Especially useful for d-ary conjunctions, see Remark 3.8! ♢

12.2.6 Construction by averaging the incoming legs

We restate The. 11.14 as a basis CP decomposition bound.

Theorem 12.20 *Let f and be a function between factored systems*

$$f : [m] \to \bigtimes_{k \in [d]} [m_k]$$

and f^k as in The. 11.14. Then $\rho^f\left[X_{[d]}, X\right]$ has a basis CP decomposition with decomposition index X, trivial slices $\mathbb{I}[X]$ leg vectors $\rho^{f^k}[X_k, X]$.

Proof. Reinterpretation of the contractions in the decomposition claimed in The. 11.14. ■

Basis CP Decompositions can be constructed by understanding the variable O_{in} of the relational encoding of a function $f : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ as the slice selection variable.

Example 12.21 Empirical distributions, see The. 1.42 Let there be a data map

$$D : [m] \to \bigtimes_{k \in [d]} [m_k].$$

We can use The. 12.20 to find a tensor network representation fo ρ^D as

$$\rho^D\left[X, X_{[d]}\right] = \left\langle \{\rho^{D^k}[X, X_k] : k \in [d]\} \right\rangle \left[X, X_{[d]}\right].$$

This representation is in the CP format, when adding trivial scalar core and and delta tensor to the data index. It is furthermore in a basis CP format, since all ρ^{D^k} are directed and binary tensors. Normation to get the empirical distribution amounts to setting a slice core with coordinates $\frac{1}{m}$. ♢

Example 12.22 Exponential families The statistic has a CP decomposition with rank by the cardinality of states ♢

12.3 Representation by slice selection architectures

The set of slice-sparse tensors coincides with the expressivity of specific selection architecture. We first define a slice selecting tensor and then show its decomposition into a formula selecting neural network.

Definition 12.23 Given a set of atomic variables $X_{[d]}$, a slice selecting tensor of maximal cardinality r is the tensor

$$\mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{0,0}, \ldots, L_{r-1,0}, L_{0,1}, \ldots, L_{r-1,1}\right]$$

with dimensions

$$p_{s,0} = 2, p_{s,1} = d$$

and coordinates

$$\mathcal{H}_{\wedge,d,r}\left[X_{[d]} = x_{[d]}, L_{0,0} = l_{0,0}, \ldots, L_{r-1,0} = l_{r-1,0}, L_{0,1} = l_{0,1}, \ldots, L_{r-1,1} = l_{r-1,1}\right]$$
$$= \begin{cases} 1 & \text{if} \quad \forall_{k,s} : \left(l_{s,1} = k \wedge l_{s,0} \neq 2\right) \Rightarrow \left(l_{s,0} = x_k\right) \\ 0 & \end{cases}.$$

In the next two Lemmata we first show that the defined slice selecting tensors indeed selects slices and then provide a representation as a formula selecting network.

Lemma 12.24 *If all input neurons with same selection index are agreeing on the connective index, the selected formula does not vanish and coincides with a slice to the set*

$$A = \{k : \exists_{s \in [n]} : l_{s,1} = k \wedge l_{s,0} \neq 2\}$$

and for $k \in A$

$$x_k = l_{s,0} \quad \textit{if} \quad l_{s,1} = k\,.$$

Proof. We need to show that

$$\mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{0,0} = l_{0,0}, \ldots, L_{r-1,0} = l_{r-1,0}, L_{0,1} = l_{0,1}, \ldots, L_{r-1,1} = l_{r-1,1}\right] = e_{x_A}\left[X_{[d]}\right] \quad (12.2)$$

If and only if an index $\tilde{x}_{[d]}$ reduced on A does not coincide with x_A, we have $e_{x_A}\left[X_{[d]} = \tilde{x}_{[d]}\right] = 0$ and otherwise $e_{x_A}\left[X_{[d]} = \tilde{x}_{[d]}\right] = 1$. Let us notice, that this condition is equivalent to

$$\forall_{k,s} : \left(l_{s,1} = k \wedge l_{s,0} \neq 2\right) \Rightarrow \left(l_{s,0} = x_k\right)$$

and thus (12.2) holds. ∎

Lemma 12.25 *The slice selection tensor coincides with a formula selecting neural network with neurons (see Figure 12.3):*

- *unary input neuron enumerated by s, selecting one of the $X_{[d]}$ with the variable $L_{s,1}$ and selecting a connective in $\{\neg, \mathrm{Id}, \mathrm{True}\}$ by $L_{s,0}$*
- *r-ary output neuron fixed to the $\wedge$ connective.*

Proof. This can be easily checked on each coordinate. ∎

It follows, that the expressivity of the slice selecting neural network coincides with the set of tensors with a bound on their slice sparsity, when $r \geq d$. For arbitrary r, the following theorem holds.

Theorem 12.26 *Let $\mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L\right]$ be a slice selecting tensor. For any parameter tensor $\theta\left[L\right]$ we have*

$$\operatorname{rank}^{\text{bas+}}\left(\left\langle \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L\right], \theta\left[L\right]\right\rangle \left[X_{[d]}\right]\right) \leq \operatorname{rank}^{\text{bas}}\left(\theta\left[L\right]\right).$$

Proof. Each non-vanishing coordinate of θ represents by Lem. 12.25 a slice and their weighted sum is thus a monomial decomposition. ∎

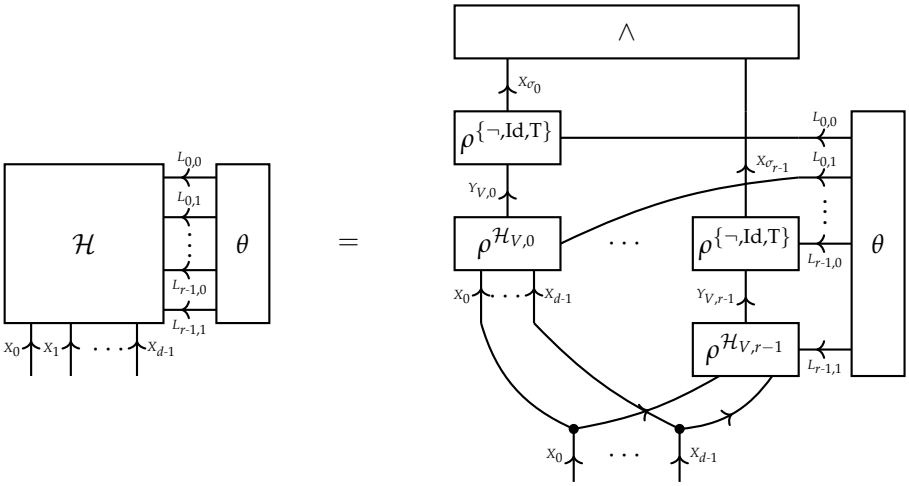


Figure 12.2: Representation of a basis+ Tensor by the contraction of a parameter tensor θ with a slice selecting architecture $\mathcal{H}$, which has a decomposition as a formula selecting neural network (see Lem. 12.25). The nonzero coordinates of θ represent the (see Lem. 12.24).

12.3.1 Applications

One application is as a parametrization scheme in the approximation of a tensor by a slice-sparse tensor, see Chapter 14.

The approximated parameter can then be used as a proxy energy to be maximized. When choosing $r = 2$, the approximating tensor contains only quadratic slices, which then poses a QUBO problem.

Remark 12.27 (Extension to arbitrary CP-formats) Select at each input neuron a specific leg. For finite number of legs, as it is the case in the binary, basis and basis+ formats, we can enumerate all possibilities by the selection variable. For the basis+ format, in case of binary leg dimensions, we here exemplified the approach, by enumerating the three possibilities $e_0, e_1, \mathbb{I}\left[1\right]$. This approach, however, fails as a generic representation of the directed format, since the directed legs are continuous and there therefore are infinite choosable legs. ♢

12.4 Optimization of sparse tensors

Let us now study the problem of searching for the maximal coordinate in a tensor represented by a monomial decomposition.

12.4.1 Mode search in exponential families

Mode search

$$\max_{x_{[d]} \in \times_{k\in[d]}[2]} \left\langle \gamma^{\phi} \left[X_{[d]} = x_{[d]}, L \right], \theta \right\rangle [\varnothing] = \max_{\mu \in \mathcal{M}} \langle \mu [L], \theta [L] \rangle [\varnothing]$$

The search for maximal coordinates appears in various reasoning tasks:

- MAP query as mode search of MLN: T is the contraction of evidence with the distribution, leaving the query variables open.
- Grafting as mode search of proposal distribution: T is the contraction of the gradient of the likelihood with the relational encoding of the hypothesis.

Both tasks have been formulated as mode search problems in exponential families.

12.4.2 Higher-Order Unconstrained Binary Optimization (HUBO)

Here binary refers to the leg dimensions m_k being 2, not to binary coordinates as often refered to in this work.

Definition 12.28 The binary optimization of a tensor $T \left[X_{[d]} \right] \in \bigtimes_{k\in[d]}[2]$ is the problem

$$\text{argmax}_{x_{[d]} \in \times_{k\in[d]}[2]} \quad T \left[X_{[d]} = x_{[d]} \right]$$

We call Problem P_T a Higher Order Unconstrained Binary Optimization (HUBO) problem of order r and sparsity $\text{rank}^r(T)$, when T has a monomial decomposition (see Def. 12.6) with $\left|A^i\right| \leq r$ for all $i \in [n]$, that is when $\text{rank}^r(T) < \infty$.

Remark 12.29 (Leg dimensions larger than 2) We demanded leg dimensions $m_k = 2$ to have binary valued variables X_k, which is required to connect with the formalism of binary optimization. Categorical variables with larger dimensions can be represented by atomization variables, which are created by contractions with categorical constraint tensors (see Sect. 6.2.4). ♢

The sparsity $\text{rank}^r(T)$ is the minimal number of monomials, for which a weighted sum is equal to T. Thus we interpret Problem P_T as searching for the maximum in a polynomial consistent of $\text{rank}^r(T)$ monomial terms. Each monomial is also refered to as potential.

12.4.3 Quadratic Unconstrained Binary Optimization (QUBO)

Quadratic Unconstrained Binary Optimization problems are HUBOs of order $r = 2$.

We refine the monomial decomposition of tensors (see Def. 12.6) by demanding that monomials consist of at most two variables.

Definition 12.30 We call a monomial decomposition $\mathcal{M}$ of a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^2$ a quadratic decomposition, if $|A| \leq 2$ for all $(\lambda, A, x_A) \in \mathcal{M}$. We denote the smallest cardinality $|\mathcal{M}|$ among quadratic decompositions of T by $\text{rank}^{\text{qua}}(T)$.

If a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^2$ has a quadratic decomposition, we call Problem P_T a Quadratic Unconstrained Binary Optimization (QUBO) problem of sparsity $\text{rank}^{\text{qua}}(T)$.

We can transform certain HUBO problems in QUBO problems with the usage of auxiliary variables, as we show in the next lemma.

Lemma 12.31 *For any $x_0, \dots, x_{d-1} \in [2]$ and $A \subset [d]$ we have*

$$\left(\prod_{k\in A} x_k\right)\left(\prod_{k\notin A}(1-x_k)\right) = \max_{z\in[2]} z\cdot 2\cdot\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right).$$

Proof. Only if $x_k = 1$ for $k \in A$ and $x_k = 0$ else we have

$$\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right) \geq 0\,.$$

In this case the maximum is taken for $z = 1$ and we have

$$\max_{z\in[2]} z\cdot 2\cdot\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right) = 1 = \left(\prod_{k\in A} x_k\right)\left(\prod_{k\notin A}(1-x_k)\right).$$

In all other cases, the maximum is taken for $z = 0$ and thus vanishes, that is

$$\max_{z\in[2]} z\cdot 2\cdot\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right) = 0 = \left(\prod_{k\in A} x_k\right)\left(\prod_{k\notin A}(1-x_k)\right).$$

Thus, the claim holds in all cases. ∎

12.4.4 Integer Linear Programming

Let us now show how optimization problems can be represented as linear programming problems.

We define for each index tuple $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ a vector $v_{x_{[d]}}[L] \in \mathbb{R}^d$ with coordinates

$$v_{x_{[d]}}[L = k] = x_k\ .$$

Definition 12.32 The integer linear program (ILP) of $M[J, L] \in \mathbb{R}^{n\times d}$, $b[J] \in \mathbb{R}^n$ and $c \in \mathbb{R}^d$ is the problem

$$\text{argmax}_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \left\langle c[L], v_{x_{[d]}}[L]\right\rangle[\varnothing] \quad \text{subject to} \quad \left\langle M[J,L], v_{x_{[d]}}[L]\right\rangle[\varnothing] \prec b[J]\,,$$

where by $\prec$ we denote partial ordering of tensors (see Def. 4.14).

We now show that any binary optimization problem of a tensor can be transformed into a integer linear program, given a monomial decomposition of the tensor $T\left[X_{[d]}\right]$ by $\mathcal{M} = \{(\lambda^i, A^i, x^i_{A^i}) : i \in [n]\}$. For this we choose state indices by vectors

$$y_{[d+n]} = x_0, \dots, x_{d-1}, z_0, \dots z_{n-1} \in \left(\bigtimes_{k\in[d]}[2]\right) \times \left(\bigtimes_{i\in[n]}[2]\right),$$

that is we added for each monomial an index z_i, which will represent the evaluations of the respective monomial.

We furthermore define a vector $c[L]$, where L takes values in $[d+n]$, as

$$c[L = l] = \begin{cases} \lambda^i & \text{if for a } i \in [n] \text{ we have } l = d + i \\ 0 & \text{else} \end{cases}. \tag{12.3}$$

To construct a matrix $M[J, L]$ and a vector $b[J]$ to the monomial decomposition $\mathcal{M}$, we now introduce a variable J enumerating linear inequalities, which takes values in $[m]$, where

$$m = \sum_{i \in [n]} \left(\left| A^i \right| + 1 \right) .$$

We define for each $i \in [n]$ an auxiliary number

$$m_i = \sum_{\tilde{i}=0}^{i} \left(\left| A^{\tilde{i}} \right| + 1 \right)$$

enumerate by $\tilde{k}$ the set A^i and set for $l \in [d + n]$ the coordinates

$$M^{\mathcal{M}}[J = m_i + j, L = l] = \begin{cases} 1 - 2 \cdot x^i_{\tilde{k}} & \text{if } j = l \text{ and } l = \tilde{k} \\ 1 & \text{if } j < d \text{ and } l = d + i \\ -x^i_{\tilde{k}} & \text{if } j = \left| A^i \right| + 1 \text{ and } l = \tilde{k} \\ -1 & \text{if } j = \left| A^i \right| + 1 \text{ and } l = d + i \end{cases} . \tag{12.4}$$

All further coordinates of $M^{\mathcal{M}}[J, L]$ not reached by this construction are set to 0. Similarly, we define $b^{\mathcal{M}}[\mathcal{M}]$ as the vector which nonvanishing coordinates are for $i \in [n]$ at

$$b^{\mathcal{M}}[\mathcal{M}] = \begin{cases} 1 & \text{if } j = \tilde{k} \text{ and } x^i_{\tilde{k}} = 0 \\ -1 + \left| \{ \tilde{k} \in A^i \, : \, x^i_{\tilde{k}} = 1 \} \right| & \text{if } j = \left| A^i \right| + 1 \text{ and } l = d + i \end{cases} . \tag{12.5}$$

Informally, we pose for each slice $|A| + 1$ linear equations. The first $|A|$ enforce, that the slice representing variable z is zero once a leg is 0. The last enforces that the slice representing variable is 1. We prove this claim more formally in the next theorem.

Theorem 12.33 *Given a monomial decomposition $\mathcal{M} = \{(\lambda^i, A^i, x^i_{A^i}) \, : \, i \in [n]\}$ of a tensor T, let $x^{ILP,\mathcal{M}}$ be a solution of the integer linear program defined by the matrix and vectors in equations (12.3), (12.4) and (12.5). Then we have*

$$x^{ILP,\mathcal{M}}|_{[d]} \in \operatorname{argmax}_{x_{[d]} \in \bigtimes_{k \in [d]} [2]} \quad T \left[X_{[d]} = x_{[d]} \right] ,$$

where by $x^{ILP,\mathcal{M}}|_{[d]}$ we denote the vector of the first d indices of $x^{ILP,\mathcal{M}}$.

Proof. We show that the constraints by

$$\left\langle M^{\mathcal{M}}[J, L] \, , v_{x_{[d]}}[L] \right\rangle [\varnothing] \prec b^{\mathcal{M}}[\mathcal{M}]$$

are satisfied, if and only if for all $i \in [n]$

$$z_i = \left(\prod_{\tilde{k} \in A^i \, , x^i_{\tilde{k}} = 0} (1 - x^i_{\tilde{k}}) \right) \cdot \left(\prod_{\tilde{k} \in A^i \, , x^i_{\tilde{k}} = 1} x^i_{\tilde{k}} \right) .$$

For any $i \in [n]$ the constraints $m_i + \tilde{k}$ for $\tilde{k} \in \left| A^i \right|$ are

$$z_i \leq \begin{cases} x^i_{\tilde{k}} & \text{if } x^i_{\tilde{k}} = 1 \\ (1 - x^i_{\tilde{k}}) & \text{if } x^i_{\tilde{k}} = 0 \end{cases}$$

and thus $z_i = 0$ if we have a mismatch on one leg, which establishes

$$z_i \leq \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 0} (1 - x^i_{\tilde{k}}) \right) \cdot \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 1} x^i_{\tilde{k}} \right) .$$

The constraint $m_i + \left| A^i \right|$ is

$$z_i \geq 1 - \left(\sum_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 0} x^i_{\tilde{k}} \right) + \left(\sum_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 1} (x^i_{\tilde{k}} - 1) \right)$$

which enforces

$$z_i \geq \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 0} (1 - x^i_{\tilde{k}}) \right) \cdot \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 1} x^i_{\tilde{k}} \right) .$$

In summary we arrive at

$$z_i = \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 0} (1 - x^i_{\tilde{k}}) \right) \cdot \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 1} x^i_{\tilde{k}} \right)$$

if and only if the indices $y_{[d+n]}$ are feasible.

Now, for any $x_{[d]}$, there is exactly one feasible index $y_{[d+n]}$ extending $x_{[d]}$, and the objective takes for this index the value

$$\begin{aligned} \left\langle c[L], v_{x_{[d]}}[L] \right\rangle [\varnothing] &= \sum_{i \in [n]} \lambda^i \cdot z_i \\ &= \sum_{i \in [n]} \lambda^i \cdot \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 0} (1 - x^i_{\tilde{k}}) \right) \cdot \left(\prod_{\tilde{k} \in A^i \,,\, x^i_{\tilde{k}} = 1} x^i_{\tilde{k}} \right) \\ &= T \left[X_{[d]} = x_{[d]} \right] . \end{aligned}$$

Therefore, any solution of the ILP reduced to the first d indices corresponding with the axis of T, is a solution of the binary optimization problem to T. ∎

In order to achieve a sparse linear program it is benefitial to use a monomial decomposition with small order and rank. Beside this sparsity, the matrix $M^{\mathcal{M}}[J, L]$ is often ℓ_0-sparse, and has thus an efficient representation in a basis CP format. More precisely we have by the above construction

$$\ell_0 \left(M^{\mathcal{M}}[J, L] \right) \leq \sum_{i \in [n]} 3 \cdot \left| A^i \right| + 1 .$$

12.5 Value subspaces of functions

To Basis Calculus?

We here provide a subspace perspective for the sparse representation of decomposable functions as tensor networks in the $\rho^{\cdot}$ relational encoding scheme of basis calculus.

Definition 12.34 Given any function f on $\bigtimes_{k\in[d]}[m_k]$ we define its value subspace as

$$V^f = \left\{ \left\langle \rho^f \left[Y_f, X_{[d]} \right], W[Y_f] \right\rangle \left[X_{[d]} \right] \,:\, W[Y_f] \in \mathbb{R}^{|\mathrm{im}(f)|} \right\}$$

Composition of functions f, h on distinct variables is then a choice of a subspace in the tensor product of the two function subspaces, that is

$$V^{f\circ h} \subset V^f \otimes V^h \,.$$

12.5.1 Propositional Formula Subspaces

Each formula defines by its relational encoding the subspace of $\bigotimes_{k\in[d]} \mathbb{R}^2$

$$V^f = \mathrm{span}\,(\neg f, f)$$

Let us notice that the spanning vector of the subspace V^f are binary tensors summing up to the tensor of ones.

For each atom X_k we have

$$V^{X_k} = \mathbb{R}^2 \,.$$

The tensor space carrying the factored representation of the worlds is thus

$$\bigotimes_{k\in[d]} V^{X_k} \,.$$

12.5.2 Formula Decomposition as a Subspace Choice

Given a formula $f \circ h$ composed of formulas f and h containing different atoms we have

$$V^{f\circ h} \subset V^f \otimes V^h$$

A connective $\circ$ thus determines the selection of a two-dimensional subspace in the four-dimensional tensor product of subspaces to both subformulas.

Reconstruction of a formula given its formula tensor amounts to finding the HT Decomposition under the constraints of subspace choices according to the allowed logical connectives.

Given a set of positive and negative examples of a formula poses further an approximation problem of the examples by a HT Decomposition.

Advantages of this perspective are

- Given a HT the best approximation always exists (Theorem 11.58 in [Hac12]), but need to further restrict to cores given by logical connectives
- Apply Approximation algorithms: ALS or HOSVD

13

Message Passing

In this chapter we introduce local contraction passed along tensor clusters to approximatively calculate global contractions. These message passing schemes provide tradeoffs between efficiency increases and exactness of the global contraction. We use the CP Decompositions to investigate the asymptotic behavior of the message passing algorithms.

The application of message passing schemata can be justified based on commutation and monotonous propoerties of large contractions. We first show these properties and then provide message passing schemata.

13.1 Commutation of Contractions

We show in the next theorem, that a contractions can be performed by contracting a subnetwork first and then further contracting the result with the rest.

Theorem 13.1 *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let us now split the $\mathcal{G}$ into two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, such that $\mathcal{E}_1 \dot\cup \mathcal{E}_2 = \mathcal{E}$, $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$ and all nodes in $\mathcal{V}_2$ are contained in an hyperedge of $\mathcal{E}_2$. We then have*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \mathcal{T}^{\mathcal{G}_1} \cup \{ \left\langle \mathcal{T}^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}_2 \cap (\mathcal{V}_1 \cup \tilde{\mathcal{V}})} \right] \} \right\rangle [X_{\tilde{\mathcal{V}}}] \ .$$

Proof. For any index $x_{\tilde{\mathcal{V}}}$ we show that

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = \left\langle \mathcal{T}^{\mathcal{G}_1} \cup \{ \left\langle \mathcal{T}^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}_2 \cap (\mathcal{V}_1 \cup \tilde{\mathcal{V}})} \right] \} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \ .$$

By definition we have

$$\begin{aligned}
\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] &= \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}}} \prod_{e \in \mathcal{E}} T^e [X_e = x_e] \\
&= \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}}} \left(\prod_{e \in \mathcal{E}_1} T^e [X_e = x_e] \right) \cdot \left(\prod_{e \in \mathcal{E}_2} T^e [X_e = x_e] \right) \\
&= \sum_{x_{\mathcal{V}_1/\tilde{\mathcal{V}}}} \sum_{x_{\mathcal{V}_2/(\tilde{\mathcal{V}} \cup \mathcal{V}_1)}} \left(\prod_{e \in \mathcal{E}_1} T^e [X_e = x_e] \right) \cdot \left(\prod_{e \in \mathcal{E}_2} T^e [X_e = x_e] \right) \\
&= \sum_{x_{\mathcal{V}_1/\tilde{\mathcal{V}}}} \left(\prod_{e \in \mathcal{E}_1} T^e [X_e = x_e] \right) \cdot \left(\sum_{x_{\mathcal{V}_2/(\tilde{\mathcal{V}} \cup \mathcal{V}_1)}} \prod_{e \in \mathcal{E}_2} T^e [X_e = x_e] \right) \ .
\end{aligned}$$

When contracting the variables $X_{\mathcal{V}_2/(\tilde{\mathcal{V}} \cup \mathcal{V}_1)}$ on $\mathcal{T}^{\mathcal{G}_2}$, the variables $X_{\mathcal{V}_2 \cap (\tilde{\mathcal{V}} \cup \mathcal{V}_1)}$ are left open. We

therefore have for any $x_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}$

$$\left\langle \mathcal{T}^{\mathcal{G}_2}\right\rangle \left[X_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)} = x_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}\right] = \left(\sum_{x_{\mathcal{V}_2/(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}} \prod_{e\in\mathcal{E}_2} T^e\left[X_e = x_e\right]\right).$$

It follows with the above, that

$$\begin{aligned}\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] &= \sum_{x_{\mathcal{V}_1/\tilde{\mathcal{V}}}} \left(\prod_{e\in\mathcal{E}_1} T^e\left[X_e = x_e\right]\right)\cdot \left\langle \mathcal{T}^{\mathcal{G}_2}\right\rangle \left[X_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)} = x_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}\right] \\ &= \left\langle \mathcal{T}^{\mathcal{G}_1}\cup\{\left\langle \mathcal{T}^{\mathcal{G}_2}\right\rangle \left[X_{\mathcal{V}_2\cap(\mathcal{V}_1\cup\tilde{\mathcal{V}})}\right]\}\right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}]\,.\end{aligned}$$

■

13.2 Support of Contractions

To state the next theorem we introduce the nonzero function $\mathbb{I}_{\neq 0} : \mathbb{R} \to [2]$ by

$$\mathbb{I}_{\neq 0}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{else} \end{cases}$$

Applied coordinatewise on tensors it marks the nonzero coordinates by 1.

We show that adding binary tensor cores to an contraction orders the results by the partial ordering introduced in Def. 4.14

Theorem 13.2 (Monotonicity of Tensor Contractions) *Let $\mathcal{T}^{\mathcal{G}}, \mathcal{T}^{\tilde{\mathcal{G}}}$ be tensor network of non-negative tensors and $X_{\tilde{\mathcal{V}}}$ an arbitrary set of random variables. Then we have*

$$\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}}\cup\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\tilde{\mathcal{V}}}]\right) \prec \mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\tilde{\mathcal{V}}}]\right).$$

Proof. It suffices to show that for any $x_{\tilde{\mathcal{V}}}$ with

$$\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}}\cup\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}]\right) = 1$$

we also have

$$\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}]\right) = 1\,.$$

For any $x_{\tilde{\mathcal{V}}}$ satisfying the first equation we find an extension $x_{\mathcal{V}}$ to all variables of the tensor networks such that

$$\left\langle \mathcal{T}^{\mathcal{G}}\cup\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] > 0$$

and it follows that

$$\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] > 0 \quad \text{and} \quad \left\langle \mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] > 0\,.$$

But this already implies, that

$$\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}]\right) = 1\,.$$

■

Let us now state an equivalence of the contraction, when we add the result of the same contraction

Theorem 13.3 (Invariance under adding subcontractions) *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network of non-negative tensors with variables $X_{\mathcal{V}}$ and let $\mathcal{T}^{\tilde{\mathcal{G}}}$ be a subset. Then we have for any subset $X_{\tilde{\mathcal{V}}}$ of $X_{\mathcal{V}}$*

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\tilde{\mathcal{V}}}]\right)\}\right\rangle [X_{\mathcal{V}}] = \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}}]\ .$$

Proof. For any $x_{\mathcal{V}}$ with

$$\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0$$

we also have

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\tilde{\mathcal{V}}}]\right)\}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0\,.$$

For any $x_{\mathcal{V}}$ with

$$\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \neq 0$$

we have for the reduction $x_{\tilde{\mathcal{V}}}$ of the index $x_{\mathcal{V}}$ that

$$\left\langle \mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \neq 0$$

and thus

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\tilde{\mathcal{V}}}]\right)\}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \cdot \mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle [X_{\tilde{\mathcal{V}}}]\right)[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = \left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}]\ .$$

■

Remark 13.4 Similar statements hold, when dropping the non-negativity assumption on the, but demanding that all variables are left open. ◊

13.3 Exact Contractions

We apply Theorem 13.1 to split a contraction into subcontractions, which are consecutively performed.

Contractions can be performed partially, and the result passed to the rest of the network as a message.

13.3.1 Construction of Cluster Graphs

Definition 13.5 (Cluster Graph) Given a tensor network $\mathcal{T}^{\mathcal{G}}$ a cluster partition is a partition of the tensor network into n clusters, by a function

$$\alpha : \mathcal{E} \to [n]\,.$$

The clusters are with tensors decorated edge sets $C_i = \{e : \alpha(e) = i\}$ with variables $\mathcal{V}_i = \bigcup_{e\in C_i} e$. The clusters form a graph where edges between C_i and C_j exist, when the node sets $\mathcal{V}_i$ and $\mathcal{V}_j$ are not disjoint. In this case, we define separation sets $S_{i,j} = C_i \cup C_j$

Theorem 13.6 *Given a tensor network $\mathcal{T}^{\mathcal{G}}$ and a cluster graph. We then define for each cluster the node set*

$$\tilde{\mathcal{V}}_i = \bigcup_{j\neq i} \mathcal{V}_j$$

and have

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \{ \left\langle \mathcal{T}^{C_i} \right\rangle [\mathcal{V}_i \cap (\tilde{\mathcal{V}}_i \cup \tilde{\mathcal{V}})] : i \in [n] \} \right\rangle [X_{\tilde{\mathcal{V}}}] \ .$$

Proof. By Theorem 13.1 applied for each cluster seen as a subgraph. ∎

13.3.2 Message Passing to calculate contractions

Having a hypergraph $\mathcal{G}$, we iteratively apply Theorem 13.1 and call the $\mathcal{G}_2$ a cluster. When iterating until $\mathcal{G}$ is empty, we get a cluster graph, where all tensors are assigned to a cluster.

When the cluster are a polytree, that is a union of disjoint trees, we define messages between neighbored clusters C_i and C_j with $C_j \prec C_i$ by the contractions

$$\delta_{j \to i} = \left\langle \{ \delta_{\tilde{j} \to j} \ : \ C_{\tilde{j}} \prec C_j \} \cup \mathcal{T}^{C_j} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] \ .$$

and

$$\delta_{j \leftarrow i} = \left\langle \{ \delta_{i \leftarrow \tilde{j}} \ : \ C_i \prec C_{\tilde{j}} \} \cup \mathcal{T}^{C_i} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] \ .$$

We note, that the messages are well defined by these recursive equations, exactly when the cluster graph is a polytree.

Lemma 13.7 *When the cluster graph is a tree, we have for neighbored clusters C_i and C_j with $C_j \prec C_i$*

$$\delta_{j \to i} = \left\langle \{ \mathcal{T}^{C_{\tilde{j}}} \ : \ C_{\tilde{j}} \prec C_j \} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right]$$

and

$$\delta_{j \leftarrow i} = \left\langle \{ \mathcal{T}^{C_{\tilde{j}}} \ : \ C_i \prec C_{\tilde{j}} \} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] \ .$$

Proof. By induction over the cardinality of the preceding clusters.

$\boldsymbol{n = 1}$: Only a single cluster before, therefore trivial.

$\boldsymbol{n + 1 \to n}$: Assuming the statement holds for up to n preceding clusters, let there be $n + 1$ preceding clusters. Then Theorem 13.1 splits contractions into terms, which are by inductive assumption the messages. ∎

Theorem 13.8 *When the cluster graph is a tree, then we have for each cluster C_i with neighbors $N(i)$*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\mathcal{V}_i] = \left\langle \{ \delta_{j \to i} \ : \ j \in N(i), \, C_j \prec C_i \} \cup \{ \delta_{i \leftarrow j} \ : \ j \in N(i), \, C_i \prec C_j \} \cup \mathcal{T}^{C_i} \right\rangle [\mathcal{V}_i] \ .$$

Proof. By Theorem 13.1 we split into contractions of the clusters up and down of the respective neighbors and apply the above lemma. ∎

13.3.3 Variable Elimination Cluster Graphs

Remark 13.9 (Construction of Cluster Graphs by Variable Elimination) Following an elimination order of the colors, mark those tensors containing the colors, which have not been marked before, as the cluster. A clique tree can be constructed by these cluster, when iterating through the clusters and either connect them to previous disconnected clusters or leave the current cluster

disconnected. Add the disconnected clusters with the current cluster in case there are overlaps of their open colors. If the disconnected cluster added has more open colors, ◊

13.3.4 Bethe Cluster Graphs

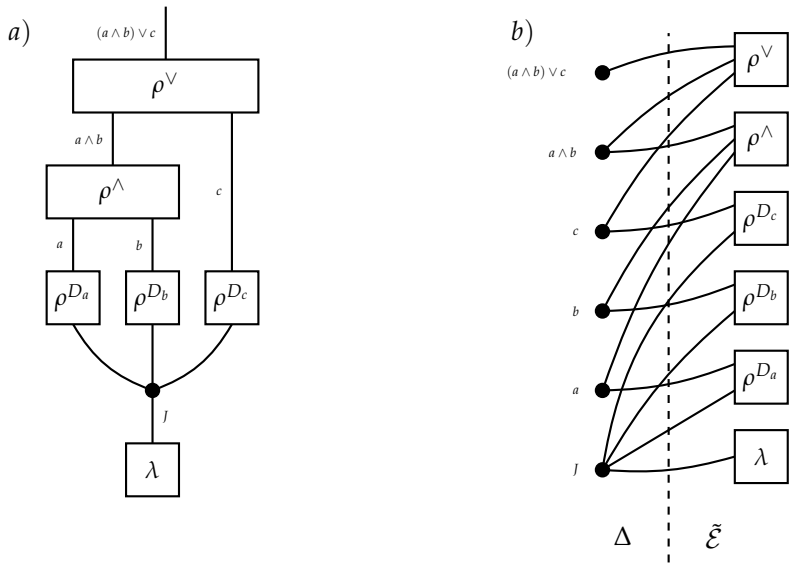


Figure 13.1: Example of a Bethe Cluster Graph. a) Example of a Tensor Network $\mathcal{T}^{\mathcal{G}}$, which represents the by λ averaged evaluation of the formula $(a \wedge b) \vee c$ on data D. b) Corresponding Bethe Cluster Hypergraph, which dual is bipartite by the sets Δ and $\tilde{\mathcal{E}}$.

By adding delta tensors to each node $v \in \mathcal{V}$ and defining its leg variables by v^e for $e \in \mathcal{E}$. We mark each such delta tensor by a cluster in $\Delta^{\mathcal{G}}$, as defined in the following (see also Figure 13.1).

Definition 13.10 Given a tensor network $\mathcal{T}^{\mathcal{G}}$ on a decorated hypergraph $\mathcal{G}$, we define the Bethe Cluster Hypergraph $\tilde{\mathcal{G}}$ as $(\tilde{\mathcal{V}}, \tilde{\mathcal{E}} \cup \Delta^{\mathcal{G}})$ where we have

- Recolored Edges $\tilde{\mathcal{E}} = \{\tilde{e} : e \in \mathcal{E}\}$ where $\tilde{e} = \{v^e : v \in e\}$, which decoration tensor has same coordinates as T^e
- Nodes $\tilde{\mathcal{V}} = \bigcup_{e \in \mathcal{E}} \tilde{e}$
- Delta Edges $\Delta^{\mathcal{G}} = \big\{\{v^e : e \ni v\} : v \in \mathcal{V}\big\}$, each of which decorated by a delta tensor $\delta^{\{v^e : e \ni v\}}$

By Lem. 10.10 this construction does not change contractions.

The dual is bipartite, since any variable appears exactly in one cluster in $\tilde{\mathcal{E}}$ and in one cluster of $\Delta^{\mathcal{G}}$. This further makes the dual of the Bethe Cluster Hypergraph a proper graph (i.e. edges consistent of node pairs).

13.3.5 Computational Complexity

Tree-width here.

Naive execution of $\langle \mathcal{T}^{\mathcal{G}} \rangle [\tilde{\mathcal{V}}]$: $\prod_{v \in \mathcal{V}} m_v$ many products are built and summed up. When splitting contractions into local subcontractions, the product can be turned into sums with tremendous decrease in complexity.

13.4 Approximate Contractions

We ignore that cluster graphs are not trees and perform contraction message passing along neighbored clusters. For the contraction of basis tensor networks, this scheme still provides the exact contraction.

13.4.1 Exact Message Passing for Directed and Binary Contractions

A Tensor Network of directed and binary cores represents the evaluation of composed functions. In a Message Passing Perspective each component (let us call them neurons) can be evaluated, when the evaluation of the ancestor neurons are known.

Lemma 13.11 *Required? Basis vector factorization suffices? For any collection of categorical variables $X_{[d]}$ with identical dimension and any $x_0 \in [m_0]$ we have*

$$\langle \delta[X_0, \dots, X_{d-1}] \rangle \, [X_0 = x_0, X_1, \dots, X_{d-1}] = \bigotimes_{k \in \{1,\dots,d-1\}} e_{x_0}[X_k] \, .$$

Proof. Directly by sum decomposition of delta tensors. ∎

Theorem 13.12 *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network on a directed acyclic hypergraph $\mathcal{G}$, such that each tensor is Boolean and directed, and such that each variable is appearing only once as an outgoing variable of a hyperedge. We build a cluster graph by storing each edge as a cluster and use the topological order on $\mathcal{G}$. Then*

$$\delta_{j \to i} = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] .$$

Proof. Lemma above needed? By using that each message is a basis vector (Using Theorem 10.14 in an induction argument) and can thus be splitted into the product of multiple copies.

Any hypercore, which is not a precessor to T^{e_i} can be omitted from the contraction by a root-stripping argument using its directionality. Therefore we have

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] = \left\langle \{ T^{e_{\tilde{j}}} \, : \, e_{\tilde{j}} \prec e_i \} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] .$$

We then replace each variable which is appearing more than once in outgoing legs by a delta tensor, which does not change the contraction by Lem. 10.10.

We then follow a leaf stripping argument and apply Theorem 13.1 iteratively on the remaining leaves. Along that line, the leave and its successors are contracted. The contraction is a basis vector and can therefore be represented as an outer product of basis vectors.

∎

When replacing e_{x_0} by an arbitrary vector in Lem. 13.11 we have

$$\langle \delta[X_0, \dots, X_{d-1}], V[X_0] \rangle \, [X_1, \dots, X_{d-1}] \neq \bigotimes_{k \in \{1,\dots,d-1\}} V[X_k] \, .$$

Therefore, the messages will in general differ from the exact contractions. To provide intuition of what happens in this case, let us take the following cases into account:

- $\lambda \cdot e_x$ sent multiple times: Result gets a factor of $\lambda^{\#\text{copies}}$ compared with the exact contraction.
- $e_x + e_{\tilde{x}}$: Result is the exact contraction added by the crossterms of sending e_x in one and $e_{\tilde{x}}$ in the other direction.

13.4.2 Case of Matrices

Here a toy example of cycling messages starting with $\mathbb{I}$. When normating the messages, the maximal singular vectors will be dominant.

We investigate the Bethe message passing for a tensor network consisting of a single matrix.

Theorem 13.13 *The stable messages are the linear subspace of the maximal singular values of the fixed core T.*

Proof. Having a Singular Value Decomposition of T and decompose the messages in the orthonormal system of the respective singular vectors. ∎

For the propagation of a and b on binary $f(a,b)$ starting with trivial messages of a and b the above theorem implies:

- Case of single possible world: Exact $f(a,b) \in \{a \wedge b, a \wedge \neg b, \neg a \wedge b, \neg a \wedge \neg b\}$ Messages are after first iteration exact
- Case of two possible worlds and $f(a,b) \in \{a \Leftrightarrow b, a \Leftrightarrow \neg b, \neg a \Leftrightarrow b, \neg a \Leftrightarrow \neg b\}$. In this situation any start message is stable and determines the other.
- Case of two possible worlds and $f(a,b) \in \{a, b, \neg a, \neg b\}$. In this situation one stable message is determined by the specified atom and the other is always stable.
- Case of three possible worlds: Approximative (exact: $1/3, 2/3$, approximative: $1 - goldenratio, goldenratio$
- Case of four possible worlds: Exact ($\mathbb{I}$)

13.4.3 Case of Tensors

Let there now be a single tensor of arbitrary order. When the tensor is not ODECO, we cannot find a CP-Decomposition with leg vectors building an orthonormal system in the respective leg spaces. This prohibits direct application of the same techniques in the case of a matrix, which is always ODECO.

13.5 Basis Calculus

Message Passing of directed and binary message by relational encoding of functions can be interpreted as function evaluation. This is because any relational encoding of a function, the decomposition

$$\rho^f = \sum_{y \in \mathrm{im}(f)} (\sum_{i : f(i) = y} e_i) \otimes e_y$$

is a SVD of the matrification of ρ^f with respect to incoming and outgoing legs.

Passing a message e_i in direction thus gives the message $e_{f(i)}$.

Remark 13.14 (Basis Calculus as Message Passing) Given a tensor network of directed and binary tensor cores T^e, each representing a function f^e. When there are not directed cycles, we define the compositions of f^e to be the function f from the nodes $\mathcal{V}_1$ not appearing as incoming nodes to the nodes $\mathcal{V}_2$ not appearing as outgoing nodes in an edge. Choosing arbitrary $x_v \in [m_v]$ for $v \in \mathcal{V}_1$ we have

$$\langle \{T^e : e \in \mathcal{E}\} \rangle [\mathcal{V}_2] = e_{f(x_v : v \in \mathcal{V}_1)} .$$

♢

13.6 Applications

When queries share same parts, can perform their contraction using dynamic programming. For conditional probability queries, which variables are the clusters of a cluster tree, this results in belief propagation.

14

Tensor Approximation

Often reasoning requires the execution of demanding contractions of tensors networks, or combinatorical search of maximum coordinates. We in this chapter investigate methods, to replace hard to be sampled tensor networks by approximating tensor networks, which then serve as a proxy in inference tasks.

14.1 Approximation of Energy tensors

14.1.1 Direct Approximation

Direct approximation is the problem

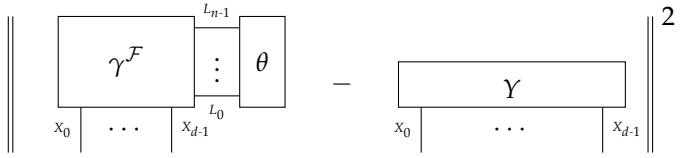
$$\text{argmin}_{\theta\in\Gamma^{\mathcal{G}}} \| E\left[X_{[d]}\right] - \theta\left[X_{[d]}\right] \|^2 .$$

14.1.2 Approximation involving Selection Architectures

Approximation involving a selection architecture $\mathcal{H}$ is the problem

$$\text{argmin}_{\theta\in\Gamma^{\mathcal{G}}} \| E - \left\langle \gamma^{\mathcal{H}}, \theta \right\rangle \left[X_{[d]}\right] \|^2 .$$

In a tensor network diagram we depict this as

$\text{argmin}_{\theta\in\Gamma^{\mathcal{G}}}$ 

Example 14.1 (Approximate based on a slice sparsity selecting architecture) Use a term selecting neural network (conjunction neuron on d unary neurons selecting a variable and Id, $\neg$, True as connective selector. Demand the parameter tensor θ to be in a basis CP format, then each slice of the parameter tensor corresponds with the slice of the energy. The use the approximation for MAP search. Same construction possible for probability tensors, but often more involved to instantiate them as tensor network. ♢

14.2 Transformation of Maximum Search to Risk Minimization

By the squares risk trick, maximum coordinate searches involving contractions with boolean tensors can be turned into squares risk minimization problems. This trick can be applied in MAP inference of MLN and the proposal distribution.

14.2.1 Weighted Squares Loss Trick

Lemma 14.2 *Let T be a Boolean tensor, that is* $\mathrm{im}\,(T) \subset \{0,1\}$. *Then*

$$T\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] - \left(T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2$$

where $\mathbb{I}$ is a tensor with same shape as T and all coordinates being 1.

Proof. Since for each $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have $T[X_{[d]} = x_{[d]}] \in \{0,1\}$, it holds that

$$T\left[X_{[d]} = x_{[d]}\right] = 1 - (T\left[X_{[d]} = x_{[d]}\right] - 1)^2$$

and thus in coordinatewise calculus

$$T\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] - \left(T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2 .$$

■

We apply this property to reformulate optimization problems over boolean tensors into weighted least squares problems.

Theorem 14.3 (Weighted Squares Loss Trick) *Let Γ be a set of boolean tensors in $\bigotimes_{k\in[d]}\mathbb{R}^{m_k}$ and $I \in \bigotimes_{k\in[d]}\mathbb{R}^{m_k}$ arbitrary. Then we have*

$$\mathrm{argmax}_{T\in\Gamma}\,\langle I, T\rangle\,[\varnothing] = \mathrm{argmin}_{T\in\Gamma}\left\langle I, (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2\right\rangle[\varnothing]$$

Proof. Using the Lemma above, T is identical to $\mathbb{I}\left[X_{[d]}\right] - (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2$ and we get

$$\langle I, T\rangle\,[\varnothing] = \left\langle I, \mathbb{I}\left[X_{[d]}\right]\right\rangle[\varnothing] - \left\langle I, (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2\right\rangle[\varnothing]$$

Since the first term does not depend on T, it can be dropped in the maximization problem. The (-1) factor then turns the maximization into a minimization problem. ■

The. 14.3 reformulates maximation of binary tensors with respect to an angle to another tensor into minimization of a squares risk. This squares risk trick is especially useful when combining it with a relaxation of Γ to differentiably parametrizable sets, since then common squares risk solvers can be applied. We will call I in the The. 14.3 importance tensor, since it manipulates the relevance of each coordinate in the squares loss.

As a result, we interpret the objective

$$\left\langle I, (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2\right\rangle[\varnothing]$$

as a weighted squares loss.

Example 14.4 (Proposal distribution maxima) The Problem **??** of finding the maximal coordinate can thus be turned into

$$\mathrm{argmax}_{l_{[n]}}\left\langle(\mathbb{P}^D - \tilde{\mathbb{P}}), \mathcal{H}\right\rangle\left[L_{[n]} = l_{[n]}\right] = \mathrm{argmin}_{l_{[n]}}\left\langle(\mathbb{P}^D - \tilde{\mathbb{P}}), \left(\left\langle\mathcal{H}, e_{l_{[n]}}\left[L_{[n]}\right]\right\rangle\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2\right\rangle[\varnothing] .$$

♢

14.2.2 Problem of the trivial tensor

By the above we motivated least squares problems on the set of one-hot encoded states. One is tempted to extend this set to $\Gamma^{\mathcal{G},\mathbb{I}}$ for efficient solutions by alternating algorithms.

However, for any hypergraph $\mathcal{G}$ we have $\mathbb{I}\left[X_{[d]}\right] \in \Gamma^{\mathcal{G},\mathbb{I}}$. In many situations (e.g. disjoint model sets supported at positive data) the objective is more in favor at the trivial tensor than at the one-hot encoding. As a result, we do not solve the previously posed one-hot encoding problem, when allowing such an hypothesis embedding.

Example 14.5 (Fitting a tensor by a formula tensor) Task: Given a tensor T, find a formula $f \in \mathcal{F}$ such that it coincides with T.

If T is a binary tensor, we understand it as a formula and want to find an f such that its number of worlds is maximal, that is solve the problem

$$\operatorname{argmax}_{f\in\mathcal{F}} \langle f \Leftrightarrow T\rangle [\varnothing] \ .$$

We can use the squares risk trick and get an equivalent problem

$$\operatorname{argmin}_{f\in\mathcal{F}} \| \ \langle f \Leftrightarrow T\rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right] \|^2 .$$

$\Diamond$

14.3 Alternating Solution of Least Squares Problems

When the parameter tensor θ is only restricted to have a decomposition as a tensor network on $\mathcal{G}$, we can iteratively update each core. The resulting algorithm is called Alternating Least Squares (ALS) (see Algorithm 10).

Algorithm 10 Alternating Least Squares (ALS)

for $e \in \mathcal{E}$ **do**
  Set $T^e[X_e]$ to a random element in $\bigotimes_{k\in e} \mathbb{R}^{m_k}$
end for
while Stopping criterion is not met **do**
  for $e \in \mathcal{E}$ **do**
    Set $T^e[X_e]$ to a to a solution of the local problem, that is

$$T^e[X_e] \leftarrow \operatorname{argmin}_{T^e[X_e]} \left\langle I, (\langle \mathcal{H}, \theta\rangle \left[X_{[d]}\right] - Y[X_{[d]}])^2 \right\rangle [\varnothing]$$

  end for
end while

14.3.1 Choice of Representation Format

The choice of the hypergraph $\mathcal{G}$ used for approximation bears a tradeoff between expressivity and complexity in sampling. Hidden variables, that is variables only present in $\mathcal{G}$, but not in the sensing matrix, increase the expressivity, especially when assigning large dimensions to them. When there are no hidden variables, the maximum of θ can be found by maximum calibration through a message passing algorithm, since no hidden variable has to marginalized.

In case of skeleton expressions with many placeholders further decomposition for algorithmic efficiency are required.

- Elementary Format (EL-Format):

- CP-Format: Closest to sum of formula tensors (when all vectors are basis, then have a sum).
- TT-Format: Showed better heuristic performance in optimization

For any tensor network decomposition into cores θ_s have the derivative $\frac{\partial}{\partial \theta_s}\theta$ as the tensor network with out the core θ_s.

14.4 Regularization and Compressed Sensing

When regularizing the least squares problem by enforcing the sparsity of θ, we arrive at the compressed sensing problem

$$\text{argmin}_{\theta[L]} \ell_0(\theta) \quad \text{subject to} \quad \left\| \left\langle \gamma^{\phi}, \theta \right\rangle \left[X_{[d]} \right] - E\left[X_{[d]} \right] \right\|_2 \leq \eta$$

Here, the sensing matrix is the selection tensor.

Example 14.6 (Formula fitting to an example) Choosing the best formula fitting data (see Example 14.5) is the problem

$$\text{argmin}_{\theta[L]:\ell_0(\theta)=1} \left\| \left\langle I, \gamma^{\phi}, \theta \right\rangle \left[X_{[d]} \right] - Y \right\|_2$$

where I has nonzero entries at marked coordinates and Y stores in Boolean coordinates whether the marked coordinates are positive or negative examples. When the number of positive and negative examples are identical, we can linearly transform the objective to that of a grafting instance, where the current model is the empirical distribution of negative examples and the data consists of the positive examples. ♢

The sparse tensor solving the problem then has a small number of nonzero coordinates and the selection tensor can be restricted to those. As a consequence, inference can be performed more efficiently.

The algorithmic solution of these problems can be done by greedy algorithms, thresholding based algorithms or optimization based algorithms [FR13].

Guarantees for the success of the algorithms depend on the properties of the sensing matrices. Here the sensing matrices are deterministic, since constructed as selection tensors, and concentration based approaches towards probabilistic bounds on these properties (see [Goe21]) are not applicable.

Example 14.7 (Propositional Formulas) Let there be a set $\mathcal{F}$ of formulas, then we have

$$\left\langle \gamma^{\mathcal{F}} \left[X_{[d]}, L_{\text{in}} \right], \gamma^{\mathcal{F}} \left[X_{[d]}, L_{\text{out}} \right] \right\rangle [L_{\text{in}} = l_{\text{in}}, L_{\text{out}} = l_{\text{out}}] = \left\langle f_{l_{\text{in}}}, f_{l_{\text{out}}} \right\rangle [\varnothing] \ .$$

If the formulas have disjoint model sets then

$$\left\langle \gamma^{\mathcal{F}} \left[X_{[d]}, L_{\text{in}} \right], \gamma^{\mathcal{F}} \left[X_{[d]}, L_{\text{in}} \right] \right\rangle [L_{\text{in}} = l_{\text{in}}, L_{\text{out}} = l_{\text{out}}] = \begin{cases} \langle f_{\text{in}} \rangle [\varnothing] & \text{if } l_{\text{in}} = l_{\text{out}} \\ 0 & \text{else} \end{cases} .$$

♢

Example 14.8 (Slice selection networks) For the slice selection network

$$\left\langle \mathcal{H}_{\wedge,d,r} \left[X_{[d]}, L_{\text{in}} \right], \mathcal{H}_{\wedge,d,r} \left[X_{[d]}, L_{\text{out}} \right] \right\rangle [L_{\text{in}} = l_{\text{in}}, L_{\text{out}} = l_{\text{out}}] = \begin{cases} 0 & \text{if for a } \tilde{k} \in A^{l_{\text{in}}} \cap A^{l_{\text{out}}} \text{ we have } x_{\tilde{k}}^{l_{\text{i}}} \\ \prod_{\tilde{k} \notin A^{l_{\text{in}}} \cup A^{l_{\text{out}}}} m_{\tilde{k}} & \text{else} \end{cases}$$

Given a fixed l_{in}, the maximum value in the respective slice is thus taken at $l_{\text{in}} = l_{\text{out}}$ ♢

A

Implementation in the tnreason package

We here document the implementation of the discussed concepts in the python package tnreason .

tnreason is an abbreviation of **t**ensor **n**etwork **reason**ing, by which we emphasize the capabilities of this package to represent and answer reasoning tasks by tensor network contractions.

The package can be installed either by cloning https://github.com/EnexaProject/enexa-tensor-reasoning or by

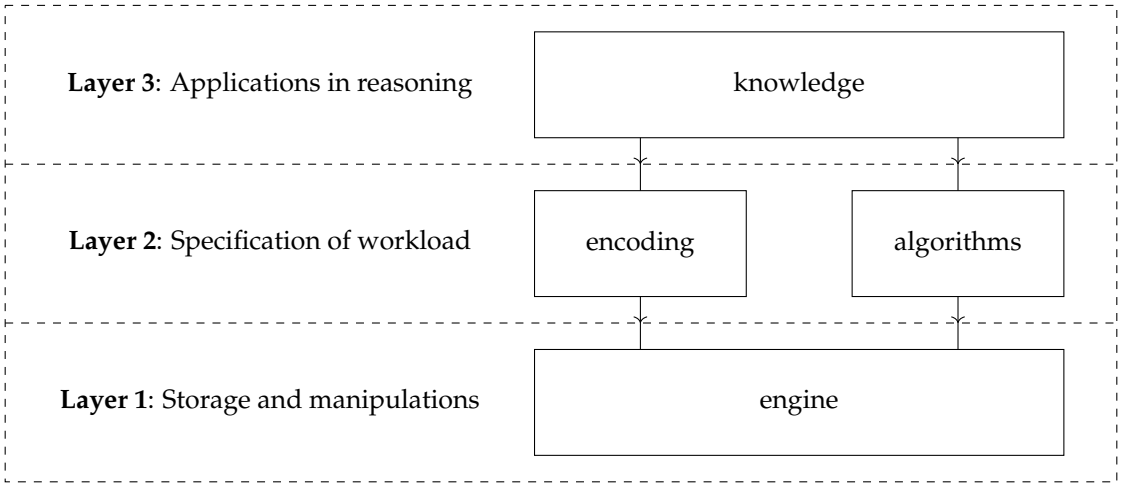
```
!pip install tnreason
```

A.1 Architecture

tnreason is structured in four subpackages and three layers

- Layer 1: Storage and numerical manipulations, by subpackage engine , "Tensor Networks" -> building "tn" of tnreason
- Layer 2: Specification of workload, subpackage encoding specific for storage, subpackage algorithms specific for manipulations
- Layer 3: Applications in reasoning, by subpackage knowledge , "Reasoning" -> building "reason" of tnreason

We sketch this structure by



A.2 Subpackage engine

The engine subpackage is for the storage and numerical manipulation of tensors and tensor networks.

We think of it as the lowest layer, specializing in storage of Tensor Networks and performing the contractions.

A.2.1 Contraction Calculus

We have described two main encoding schemes of functions, by a direct interpretation of functions as tensors or a more relational encoding. Both come with a different calculus scheme, which we have framed coordiante calculus and basis calculus.

A.2.2 Cores and Contractions

Cores
Each Tensor core has attributes

- values (array-like): storing the value of the coordinates
- colors (list of str): specifying the name of the variables represented by its axes
- name (str): to distinguish from other cores

The implemented core types differ in the values argument. Cores are instantiated by

```
engine.getCore(coreType)(coreValues, coreColors, coreName)
```

Polynomial Cores Polynomial Cores are implementations of the monomial decomposition or basis+ (see Def. 12.6). Here the each tuple (λ, A, X_A) is stored as a tuple of the scalar λ and a dictionary with A as keys and X_A as values.
The supported cores are

coreType	Package	Explanation
"NumpyTensorCore"	numpy	Numpy array storing the values
"PolynomialCore"	numpy	Storeing the values in a binary CP Decomposition

Binary CP Decomposition
Based on the monomial decomposition $\text{rank}^d(\cdot)$ as specified in Def. 12.6. To store the values of a tensor we store the slices of tensors by the indices x_A.
Contractions can be performed by partially contracting the cores of the decomposition. In this way, one can avoids coordinatewise storages of high-order tensors, which can be intractable.
Tensor Networks
Tensor networks $\mathcal{T}^{\mathcal{G}}$ are defined by hypergraphs with hyperedges decorated by tensor cores. We store them by dictionaries with values being tensor cores and keys coinciding with the name of each tensor core.
Contractions
Reflected in the notation

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}]$$

a contraction is defined by

- Tensor Network $\mathcal{T}^{\mathcal{G}}$, i.e. a dictionary of tensor cores
- Open Variables $\mathcal{V}$

Contraction calls are done by

```
engine.contract(contractionMethod, coreDict, openColors,
dimensionDict, evidenceColorDict)
```

Where

- contractionMethod: str, chooses one of the contraction providers
- coreDict: Dictionary of TensorCores (of the above formats), representing the Tensor Network $\mathcal{T}^{\mathcal{G}}$

- openColors: List of str, each str identifying a color, that is a variable to be left open in the contraction
- dimensionDict: Dict valued by int and keys by str, storing dimensions to each variable. This is of optional usage, when a color in openColors does not appear in the coreDict.
- evidenceColorDict: Dict valued by int and keys by str, indicating sliced variables

The supported contraction methods are

contractionMethod (str)	**Package**	Explanation
"NumpyEinsum"	numpy	Einstein summation of numpy arrays
"TensorFlowEinsum"	tensorflow	Einstein summation of tensorflow tensors
"TorchEinsum"	torch	Einstein summation of torch tensors
"TentrisEinsum"	tentris	Einstein summation of tentris hypertries
"PgmpyVariableEliminator"	pgmpy	Variable Elimination of DiscreteFactors in pgmpy
"PolynomialContractor"	numpy	Contraction of CP Decompositions stored in numpy arrays

Contractions represented as Einstein summation, as implemented in:

- numpy
- tensorflow
- pytorch
- tentris

Contractions can be executed by variable elimination as implemented in:

- pgmpy

Manipulation of Binary CP Decomposition Contraction of tensors in Binary CP Decomposition as in Sect. **??**.

Coordinate Calculus
Main function

```
engine.coordinate_transform(coresList, transformFunction)
```

Basis Calculus
Main function

```
engine.relational_encoding()
```

basis calculus then based on contractions

A.3 Subpackage encoding

In the encoding subpackage we encode maps Here the relational encodings ρ^f of various maps f are created. The maps are either specified by the script language (logical formulas or neuro-symbolic architectures), categorical constraints or data. Given a specification of a formula f in script language $S(\cdot)$, the task amounts to building a semantic representation based on the syntactic specification.

We arrange the encoding subpackage into the second layer of the tnreason architecture, since it specifies tensor cores which formats are specified in engine .

A.3.1 Script Language

To specify propositional sentences, neuro-symbolic architectures and Markov Logic Networks, we developed a script language.

Propositional Sentences by Nested Lists

Are those of Propositional Logics, but instead of brackets we nest the symbols into lists.

Connectives are represented by strings, where the following are supported (see Def. **??**):

Unary connective $\circ$	$S(\circ)$
$\neg$	"not"
$()$	"id"

Binary connective $\circ$	$S(\circ)$
$\wedge$	"and"
$\vee$	"or"
$\Rightarrow$	"imp"
$\oplus$	"xor"
$\Leftrightarrow$	"eq"

Besides these specific connectives we exploit a generic representation scheme of propositional formulas by the so-called Wolfram code orginially designed for the classification of cellular automaton rules [Wol83] and popularized in the book [Wol02]. Along this, the coordinate encodings of connectives $\circ$ with differing arity are flattened and interpreted as a binary number, which is transformed into a decimal number and represented as a string $S(\circ)$. We then choose a prefix to encode the arity by

- "u" for unary
- "b" for binary
- "t" for ternary
- "q" for quarternary

connectives. Together, the connective is represented by the string concatenation

$$S(\circ) = S(d) + S(\circ) \ .$$

Atomic Formulas are represented by arbitrary strings, which are not used for the representation of connectives. We further avoid the symbols {"(", ")", "_"} in the names of atoms, to not confuse them with colors of categorical variables.

Composed Formulas $f_1\circ, f_2$ are represented by

$$S(f_1\circ, f_2) = [S(\circ), S(f_1), S(f_2)]$$

where we apply the conventions

- Connectives are at the 0th position in each list
- Further entries are either atoms as strings or encoded formulas itself

The applied grammar in Backus-Naur form is

Unary Connective	"not" | "id"
Binary Connective	"and" | "or" | "imp" | "xor" | "eq"
Atomic Formula	Set of strings not in Connectives
Complex Formula	Atomic Formula | [Unary Connective, Complex Formula] | [Binary Connective, Complex Formula, Complex Formula]

Example A.1 (Encoding of the Wet Street example) For example we have

- Atomic variable *Rained* by
  $S(\textit{Rained})$ = "Rained"

- Negative literal $\neg Rained$ by
  $S(\neg Rained)$ = ["not","Rained"]
- Horn clause $(Rained \Rightarrow Wet)$ by
  $S(Rained \Rightarrow Wet)$ = ["imp","Rained","Wet"]
- Knowledge Base $(\neg Rained) \wedge (Rained \Rightarrow Wet)$ by
  $S(\neg Rained) \wedge (Rained \Rightarrow Wet)$ = ["and", ["not", "Rained"], ["imp", "Rained", "Wet"]]

$\Diamond$

Knowledge Bases

We distinguish here formulas, with propositional logic interpretation and formulas which have a soft logic interpretation. The formulas with hard interpretation are called facts in a knowledge base $\mathcal{KB}$ and encoded by dictionaries

$$\{\text{key}(f) : S(f) \text{ for } f \in \mathcal{KB}\}$$

Markov Logic Networks

The formulas with soft interpretation are called weighted formulas and encoded by $\exp\left[\theta_f \cdot f\right]$. We thus require a specification of the weights, which we do by adding θ_f as a float or an int to the list $S(f)$. We then store Markov Logic Networks by dictionaries

$$\{\text{key}(f) : S(f) + [\theta_f] \text{ for } f \in \mathcal{F}\}$$

Neuro-Symbolic Architecture by Nested Lists

To specify neuro-symbolic architectures in terms of formula selecting maps, as has been the subject of Chapter 5 we further exploit the nested list structure of encoding propositional logics. We replace, in each hierarchy of the nested structure each entry by a list of possible choices. In this way, we reinterpret the list index as the choice indices l introduced for connective and formula selections (see Def. 5.2 and **??**).

A connective selector (see Def. 5.2) is encoded by the list

$$S(\circ) = [S(\circ_0), \ldots, S\left(\circ_{p\text{-}1}\right)]$$

and a formula selector (see Def. **??**) by

$$S(\mathcal{H}) = [S(\circ_0), \ldots, S\left(\circ_{p\text{-}1}\right)]$$

A logical neuron of order n (see Def. 5.8), defined by a connective selector $\circ$, and a formula selector $\mathcal{H}_k$ on each argument $k \in [n]$, is encoded by

$$S(\sigma) = [S(\circ), S(\mathcal{H}_0), \ldots, S(\mathcal{H}_{n-1})]$$

Only the unary $n = 1$ and the $n = 2$ cases are supported.

The resulting nested lists indices have an alternating interpretation at each level compared with the elements of each list. That is, when $S(\sigma)$ is the encoding of a neuron, then any element $x \in S(\sigma)$ represents a list of choices. When x is not the first element, then each choice is either the encoding $S(X)$ of an atomic formula, or another neuron.

A neural architecture $\mathcal{A}$ is then represented in the dictionary

$$S(\mathcal{A}) = \{\text{key}(\sigma) : S(\sigma) \text{ for } f \in \mathcal{A}\}$$

where key(σ) is a string, which can be used in the formula selections of other neurons.

It is important that the directed graph of neurons induced by the choice possibilities is acyclic, to ensure well-definedness of the architecture.

In order to represent neuro-symbolic architectures, the grammar of $S(\cdot)$ in Backus-Naur Form is extended by the production rules

Unary Connectives	[Unary Connective] ∣ [Unary Connective] + Unary Connectives
Binary Connectives	[Unary Connective] ∣ [Binary Connective] + Binary Connectives
Dependency Choice	Atomic Formula ∣ Neuron
Dependency Choices	[Dependency Choice] ∣ [Dependency Choice] + Dependency Choices
Neuron	[Unary Connectives, Dependency Choices] ∣ [Binary Connectives, Dependency Choices, Dependency Choices]

Example A.2 (Neuro-Symbolic Architecture for the Wet Street) Following the wet street example, we can define a neuron by

$$S(\sigma) = [["imp","eq"],["Wet","Sprinkler"],["Street"]]$$

from which the formulas

["imp", "Wet", "Street"]
["eq", "Wet", "Street"]
["imp", "Sprinkler", "Street"]
["eq", "Sprinkler", "Street"]

can be chosen. Combining this neuron with further neurons, e.g. by the architecture

$$S(\mathcal{A}) = \{ \text{"neur1": } [["imp","eq"],["neur2"],["Street"]] \text{ ,}$$
$$\text{"neur2": } [["lnot","id"],["Wet","Sprinkler"],["Street"]] \}$$

the expressitivity increases. In this case, the further neuron provides the flexibility of the first atoms to be replaced by its negation. ◊

A.3.2 Core Nomenclature

In encoding.suffixes we defined suffixes for the names of cores and colors, which highlight their origin and purpose.

Cores are named with suffixes based on their funcitonality

- "_conCore": logical connectives (relational encoding of the connective map)
- "_headCore": two-dimensional vectors representing of the activation core to a formula
- "_dataCore": Representing (relational encoding of the data map)
- "_catCore": Categorical constraint cores
- Add: Formula selecting cores

A.3.3 Color Nomenclature

Represent the three appearing types of variables:

- Categorical variables $X.$: cVar / aVar
- Selection variables $L.$: sVar
- Term variables $O.$: tVar

A.3.4 Relational encoding of formulas

Propositional formulas f are represented in three schemes:

- Script language $S(f)$ by nested lists (see Sect. A.3.1). Most practical to choose a formula from a neuro-symbolic architecture.
- Strings specifying the categorical variables X_f.
- Representation of formulas by tensor networks being contracted to ρ^f

Conversions of the formats:

- $S(f)$ to color by
  encoding.get_formula_color($S(f)$)
  Here the nested lists are turned in a string by concatenating all elements of a list with "_" and adding "[" and "]" at the beginning and end of each list.
- $S(f)$ to tensor network
  encoding.create_raw_cores($S(f)$)
  This creates the connective cores for the semantic representation of ρ^f. We encode them by

When encoding formulas with hard interpretation, we furthermore add a head core of type "truthEvaluation" since we have

$$f = \left\langle \rho^f, e_1 \right\rangle \left[X_f \right] .$$

A.3.5 Representation of MLNs

Structure Cores are binary cores relating the variables in a predefined way, which is not changing during reasoning.

- Logical interpretation: Cores ρ° Structure Cores are those of the Bayesian Propositional Network
- Categorical constraints: Cores ρ^Z

Activation Cores encode the weights of the formulas in a Markov Logic Network. For proper MLN only have unary cores, which we call headCores. Head cores with suffix "headCore" in name.

They are modified during reasoning: Selection of activation cores in structure learning, assigning a weight in parameter estimation.

A.3.6 Formula Selecting Maps

Encoding of Neurons according to Def. 5.8:

- Activation selection core with suffix "actCore" in name. Selection by variable with suffix "actVar"
- Selection of neurons as arguments with suffix "selCore" in name. Each argument of each neuron comes with a control variable with suffix "selVar".

Encoding of Formula Selecting Neural Networks (Def. 5.8) by creating all formula selecting neurons.

Skeleton expression (Def. **??**) are stored with placeholderkeys and the candidatelists by dictionaries with the placeholderkeys and values being the possible symbols.

A.4 Subpackage algorithms

The algorithms subpackage implements basic tensor network algorithms with calls of specific execution in engine . As the encoding subpackage it is arranged in the second layer of the tnreason architecture, since it specifies the manipulation of tensor networks in the engine subpackage.

A.4.1 Alternating Least Squares

- Tensor Network of Structure Cores
- Tensor Network of Parameter Cores
- List of importance cores

algorithms.ALS

A.4.2 Gibbs Sampling

- Tensor Network of Structure Cores
- Parameter cores: Variable tensor network cores representing basis vectors.
- List of importance cores

algorithms.Gibbs

A.4.3 Knowledge Propagation

algorithms.ConstraintPropagator

A.4.4 Energy-based Algorithms

algorithms.NaiveMeanField

algorithms.GenericMeanField

algorithms.EnergyBasedGibbs

A.5 Subpackage knowledge

With the knowledge subpackage we provide an interface for reasoning workload. It builds a third layer, since it used encoding to represent knowledge by tensor networks and algorithms in the execution of reasoning tasks.

A.5.1 Distributions

We encode Markov Networks by specifying a set of tensor cores. Each distribution needs to have a routine

.create_cores()

creating the factor cores and

.get_partition_function()

calculating the partition function. Although the partition function can be calculated by the contraction of all cores, we separate the method since there are situations where a faster calculation can be performed.

Empirical Distributions are distributions of sample data. We represent the values as a CP Format of data cores as specified in Sect. 1.8

knowledge.EmpiricalDistribution

Here the partition function is the number of samples used in the creation of the empirical distribution.

HybridKnowledgeBases are probability distributions, which are specified by propositional formulas in the script language.

knowledge.HybridKnowledgeBase

They are initialized with arguments

- facts: Dictionary of propositional formulas stored as $S(f)$ representing hard logical constraints
- weightedFormulas: Dictionary of propositional formulas stored as $S(f)$+$[\theta_f]$ representing soft logical constraints
- evidence: Dictionary of atomic formulas, where key are the formulas in string representation and values the certainty in $[0, 1]$ (float or int) of the atom being true
- categoricalConstraints: Dictionary of categorical constrained, which values are lists of atomic formulas stored as strings $S(X)$

A.5.2 Inference

By

knowledge.InferenceProvider

taking a distribution from the above as argument.

Probabilistic queries as specified Def. 2.1) by

.query(variableList, evidenceDict)

MAP queries by

.exact_map_query()

or by

.annealed_sample()

using Simulated Annealing (see Remark **??**) to find an approximate maximum. The second method circumvents the creation of the coordinatewise representation of the distribution and circumvents therefore, at the expense of potentially approximative solutions, a bottleneck in case of many query variables.

Entailment from the distribution (Def. **??**) is be decided by

.ask(queryFormula, evidenceDict)

where queryFormula is the formula f to be tested for entailment in the representation $S(f)$.

Samples can be drawn by

.draw_samples(sampleNum, variableList, annealingPattern)

based on Gibbs sampling, where

- sampleNum (int) gives the number of samples to be drawn
- variableList (list of str) defines the variables to be represented by the samples (default: all atoms in the distribution)
- annealingPattern specifies an annealing pattern

A.5.3 Parameter Estimation

EntropyMaximizer implements Algorithm 8, which is motivated by the maximum entropy principle (see Sect. 2.5.3) to optimize Markov Logic Networks. The class

knowledge.EntropyMaximizer

is initialized with the arguments

- expressionsDict: Dictionary of formulas in the format $S(f)$
- satisfactionDict: Dictionary of the satisfaction rates (mean parameters) to be matched by the optimal distribution

The optimization is then performed by

.alternating_optimization(sweepNum, updateKeys)

method, where the iteration in Algorithm 8 through the updateKeys is performed sweepNum times.

A.5.4 Structure Learning

Formula Booster chooses a formula given a formula selecting map.

knowledge.FormulaBooster

is initialized with the arguments

- knowledgeBase: Distribution representing a current model to be improved
- specDict: A neuro-symbolic architecture encoded in a dictionary of neurons

Bibliography

[Bad+22] Samy Badreddine et al. "Logic Tensor Networks". *Artificial Intelligence* 303 (Feb. 2022), 103649. ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103649. URL: https://www.sciencedirect.com/science/article/pii/S0004370221002009 (visited on 11/20/2023).

[Bar16] Albert-László Barabási. *Network Science*. English. Illustrated edition. Cambridge: Cambridge University Press, July 2016. ISBN: 978-1-107-07626-6.

[Big+20] Alexander Bigerl et al. "Tentris -A Tensor-Based Triple Store". Sept. 2020.

[CKP13] Peter G. Casazza, Gitta Kutyniok, and Friedrich Philipp. "Introduction to Finite Frame Theory". en. *Finite Frames: Theory and Applications*. Ed. by Peter G. Casazza and Gitta Kutyniok. Boston: Birkhäuser, 2013, 1–53. ISBN: 978-0-8176-8373-3. DOI: 10.1007/978-0-8176-8373-3_1. URL: https://doi.org/10.1007/978-0-8176-8373-3_1 (visited on 02/04/2025).

[Cic+15] Andrzej Cichocki et al. "Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis". *IEEE Signal Processing Magazine* 32.2 (Mar. 2015). Conference Name: IEEE Signal Processing Magazine, 145–163. ISSN: 1558-0792. DOI: 10.1109/MSP.2013.2297439.

[CYM20] William Cohen, Fan Yang, and Kathryn Rivard Mazaitis. "TensorLog: A Probabilistic Database Implemented Using Deep-Learning Infrastructure". en. *Journal of Artificial Intelligence Research* 67 (Feb. 2020), 285–325. ISSN: 1076-9757. DOI: 10.1613/jair.1.11944. URL: https://jair.org/index.php/jair/article/view/11944 (visited on 02/20/2024).

[DN21] Caglar Demir and Axel-Cyrille Ngonga Ngomo. "DRILL- Deep Reinforcement Learning for Refinement Operators in ALC". eng. *CoRR* abs/2106.15373 (2021). URL: https://ris.uni-paderborn.de/record/25217 (visited on 11/06/2023).

[FR13] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. en. Applied and Numerical Harmonic Analysis. Birkhäuser Basel, 2013. ISBN: 978-0-8176-4947-0. DOI: 10.1007/978-0-8176-4948-7. URL: https://www.springer.com/de/book/9780817649470 (visited on 12/11/2019).

[Fou25] Python Software Foundation. *Python Language Reference, version 3.13.2*. Feb. 2025. URL: https://docs.python.org/3/ (visited on 03/06/2025).

[Gal+13] Luis Antonio Galárraga et al. "AMIE: association rule mining under incomplete evidence in ontological knowledge bases". en. *Proceedings of the 22nd international conference on World Wide Web*. Rio de Janeiro Brazil: ACM, May 2013, 413–422. ISBN: 978-1-4503-2035-1. DOI: 10.1145/2488388.2488425. URL: https://dl.acm.org/doi/10.1145/2488388.2488425 (visited on 11/06/2023).

[Gel+19] Patrick Gelß et al. "Multidimensional Approximation of Nonlinear Dynamical Systems". *Journal of Computational and Nonlinear Dynamics* 14.6 (Apr. 2019), 061006–061006–12. ISSN: 1555-1415. DOI: 10.1115/1.4043148. (Visited on 05/01/2019).

[GT19] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. Englisch. MIT Press, Sept. 2019. ISBN: 978-0-262-53868-8.

[Gil07] Rafael Gillmann. "0/1-Polytopes: Typical and Extremal Properties". English. PhD thesis. Feb. 2007. URL: https://depositonce.tu-berlin.de/items/urn:nbn:de:kobv:83-opus-14695 (visited on 03/04/2025).

[Gio17] Vincenzo Nicosia Giovanni Russo Vito Latora. *Complex Networks: Principles, Methods and Applications. With 58 exercises*. English. Cambridge, United Kingdom ; New York, NY: Cambridge University Press, Sept. 2017. ISBN: 978-1-107-10318-4.

[Gla+19] Ivan Glasser et al. "Expressive power of tensor-network factorizations for probabilistic modeling". en. *Advances in Neural Information Processing Systems* 32 (2019). (Visited on 07/06/2021).

[Goe21] Alex Christoph Goeßmann. "Uniform Concentration of Tensor and Neural Networks: An Approach towards Recovery Guarantees". en. Accepted: 2021-12-30T15:00:58Z. PhD Thesis. Berlin: Technische Universität Berlin, 2021. URL: https://depositonce.tu-berlin.de/handle/11303/15990 (visited on 01/13/2022).

[Goe+20] Alex Goeßmann et al. "Tensor network approaches for data-driven identification of non-linear dynamical laws". en. *Advances in Neural Information Processing Systems - First Workshop on Quantum Tensor Networks in Machine Learning*. 2020, 21.

[HK09] W. Hackbusch and S. Kühn. "A New Scheme for the Tensor Representation". en. *Journal of Fourier Analysis and Applications* 15.5 (Oct. 2009), 706–722. ISSN: 1531-5851. (Visited on 06/18/2021).

[Hac12] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. en. Springer Series in Computational Mathematics. Berlin Heidelberg: Springer-Verlag, 2012. ISBN: 978-3-642-28026-9. DOI: 10.1007/978-3-642-28027-6. (Visited on 01/30/2020).

[HL93] Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. English. 1993rd edition. Berlin, Heidelberg: Springer, Oct. 1993. ISBN: 978-3-540-56852-0.

[Hoc22] Sepp Hochreiter. "Toward a broad AI". en. *Communications of the ACM* 65.4 (Apr. 2022), 56–57. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3512715. URL: https://dl.acm.org/doi/10.1145/3512715 (visited on 02/22/2024).

[Hog+21] Aidan Hogan et al. *Knowledge Graphs*. English. 1st edition. Cham: Springer, Nov. 2021. ISBN: 978-3-031-00790-3.

[KB09] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". en. *SIAM Review* 51.3 (Aug. 2009), 455–500. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/07070111X. (Visited on 01/28/2021).

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. English. 1. edition. Cambridge, Mass.: The MIT Press, July 2009. ISBN: 978-0-262-01319-2.

[Kou+22] N'Dah Jean Kouagou et al. *Neural Class Expression Synthesis*. en. arXiv:2111.08486 [cs]. Dec. 2022. URL: http://arxiv.org/abs/2111.08486 (visited on 11/06/2023).

[Kou+23] N'Dah Jean Kouagou et al. "Neural Class Expression Synthesis". en. *The Semantic Web*. Ed. by Catia Pesquita et al. Vol. 13870. Series Title: Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023, 209–226. ISBN: 978-3-031-33454-2 978-3-031-33455-9. DOI: 10.1007/978-3-031-33455-9_13. URL: https://link.springer.com/10.1007/978-3-031-33455-9_13 (visited on 11/06/2023).

[Leh+11] Jens Lehmann et al. "Class expression learning for ontology engineering". *Journal of Web Semantics* 9.1 (Mar. 2011), 71–81. ISSN: 1570-8268. DOI: 10.1016/j.websem.2011.01.001. URL: https://www.sciencedirect.com/science/article/pii/S1570826811000023 (visited on 11/06/2023).

[Mac03] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Englisch. Illustrated Edition. Cambridge: Cambridge University Press, Sept. 2003. ISBN: 978-0-521-64298-9.

[Mar+24] Giuseppe Marra et al. “From statistical relational to neurosymbolic artificial intelligence: A survey”. *Artificial Intelligence* 328 (Mar. 2024), 104062. ISSN: 0004-3702. DOI: 10.1016/j.artint.2023.104062. URL: https://www.sciencedirect.com/science/article/pii/S0004370223002084 (visited on 02/20/2024).

[MD94] Stephen Muggleton and Luc De Raedt. “Inductive logic programming: Theory and methods”. *The Journal of Logic Programming* 19 (1994). Publisher: Elsevier, 629–679. URL: https://www.sciencedirect.com/science/article/pii/0743106694900353 (visited on 11/06/2023).

[Nic+16] Maximilian Nickel et al. “A Review of Relational Machine Learning for Knowledge Graphs”. en. *Proceedings of the IEEE* 104.1 (Jan. 2016), 11–33. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2015.2483592. URL: https://ieeexplore.ieee.org/document/7358050/ (visited on 11/06/2023).

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Englisch. s.l.: Morgan Kaufmann, Sept. 1988. ISBN: 978-1-55860-479-7.

[Pea09] Judea Pearl. *Causality: Models, Reasoning and Inference. Ausgezeichnet: ACM Turing Award for Transforming Artificial Intelligence 2011*. Englisch. 2nd ed. Cambridge New York, NY Port Melbourne New Delhi Singapore: Cambridge University Press, Nov. 2009. ISBN: 978-0-521-89560-6.

[RD06] Matthew Richardson and Pedro Domingos. “Markov logic networks”. en. *Machine Learning* 62.1-2 (Feb. 2006), 107–136. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/s10994-006-5833-1. URL: http://link.springer.com/10.1007/s10994-006-5833-1 (visited on 01/14/2023).

[RS19] Elina Robeva and Anna Seigal. “Duality of graphical models and tensor networks”. *Information and Inference: A Journal of the IMA* 8.2 (June 2019), 273–288. ISSN: 2049-8772. DOI: 10.1093/imaiai/iay009. (Visited on 07/06/2021).

[Roc97] Ralph Tyrell Rockafellar. *Convex Analysis*. English. reprint edition. Princeton: Princeton University Press, Jan. 1997. ISBN: 978-0-691-01586-6.

[RN21] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach, Global Edition: A Modern Approach, Global Edition*. Englisch. 4th ed. Boston: Pearson, May 2021. ISBN: 978-1-292-40113-3.

[SIS17] Chiaki Sakama, Katsumi Inoue, and Taisuke Sato. “Linear Algebraic Characterization of Logic Programs”. en. *Knowledge Science, Engineering and Management*. Ed. by Gang Li et al. Vol. 10412. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, 520–533. ISBN: 978-3-319-63557-6 978-3-319-63558-3. DOI: 10.1007/978-3-319-63558-3_44. URL: http://link.springer.com/10.1007/978-3-319-63558-3_44 (visited on 11/06/2023).

[Sat17] Taisuke Sato. “A linear algebraic approach to datalog evaluation”. en. *Theory and Practice of Logic Programming* 17.3 (May 2017). Publisher: Cambridge University Press, 244–265. ISSN: 1471-0684, 1475-3081. DOI: 10.1017/S1471068417000023. URL: https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/linear-algebraic-approach-to-datalog-evaluation/CED3EEB903D9D8A16843CFCA5AC4D577 (visited on 11/06/2023).

[SB14] Shalev-Schwartz, Shai and Ben-David, Shai. *Understanding Machine Learning: From Theory to Algorithms*. Englisch. New York, NY, USA: Cambridge University Press, July 2014. ISBN: 978-1-107-05713-5.

[Tal14] Michel Talagrand. *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*. en. Berlin, Heidelberg: Springer, 2014. ISBN: 978-3-642-54074-5. DOI: 10.1007/978-3-642-54075-2. (Visited on 05/05/2020).

[Tro17] Theo Trouillon. “Knowledge Graph Completion via Complex Tensor Factorization”. en (2017).

[TN17] Théo Trouillon and Maximilian Nickel. *Complex and Holographic Embeddings of Knowledge Graphs: A Comparison*. en. arXiv:1707.01475 [cs, stat]. July 2017. URL: http://arxiv.org/abs/1707.01475 (visited on 11/06/2023).

[TAP24] Efthimis Tsilionis, Alexander Artikis, and Georgios Paliouras. “A Tensor-Based Formalization of the Event Calculus”. en. *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*. Jeju, South Korea: International Joint Conferences on Artificial Intelligence Organization, Aug. 2024, 3584–3592. ISBN: 978-1-956792-04-1. DOI: 10.24963/ijcai.2024/397. URL: https://www.ijcai.org/proceedings/2024/397 (visited on 09/24/2024).

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. English. 1st edition. New York, NY: Cambridge University Press, Sept. 2018. ISBN: 978-1-108-41519-4.

[Wai19] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press, 2019. ISBN: 978-1-108-49802-9. DOI: 10.1017/9781108627771. (Visited on 11/23/2020).

[WJ08] Martin J. Wainwright and Michael Irwin Jordan. *Graphical Models, Exponential Families, and Variational Inference*. en. Now Publishers Inc, 2008. ISBN: 978-1-60198-184-4.

[Wol83] Stephen Wolfram. “Statistical mechanics of cellular automata”. *Reviews of Modern Physics* 55.3 (July 1983). Publisher: American Physical Society, 601–644. DOI: 10.1103/RevModPhys.55.601. URL: https://link.aps.org/doi/10.1103/RevModPhys.55.601 (visited on 02/11/2025).

[Wol02] Stephen Wolfram. *A New Kind of Science*. Englisch. Illustrated Edition. Champaign (Ill.): Wolfram Media, May 2002. ISBN: 978-1-57955-008-0.

[Yan+15] Bishan Yang et al. “Embedding Entities and Relations for Learning and Inference in Knowledge Bases”. en. arXiv:1412.6575 [cs]. arXiv, Aug. 2015. URL: http://arxiv.org/abs/1412.6575 (visited on 11/06/2023).

[Zie00] Günter M. Ziegler. “Lectures on 0/1-Polytopes”. en. *Polytopes — Combinatorics and Computation*. Ed. by Gil Kalai and Günter M. Ziegler. Basel: Birkhäuser, 2000, 1–41. ISBN: 978-3-0348-8438-9. DOI: 10.1007/978-3-0348-8438-9_1. URL: https://doi.org/10.1007/978-3-0348-8438-9_1 (visited on 03/04/2025).

[Zie13] Günter M. Ziegler. *Lectures on Polytopes*. English. 1995th edition. New York: Springer, Oct. 2013. ISBN: 978-0-387-94365-7.