The Tensor Network Approach to Efficient and Explainable AI

Research Notes in the ENEXA Project

Alex Goessmann, DATEV eG

April 4, 2025

Abstract

Recent models in artificial intelligence, despite performance breakthroughs in large language models, suffer from limited efficiency and explainability, which prevents them from unlocking their full application potential for economic and trustworthy use. We in this work leverage the mathematical structure of tensor networks, which has been eminent in artificial intelligence from the beginning, to achieve the goals of efficiency and explainability.

While tensors appear naturally in artificial intelligence as factored representations of systems, their decompositions into tensor networks is necessary to avoid the curse of dimensionality. Since the curse of dimensionality prevents feasible generic representations, logical and probabilistic reasoning approaches trade off efficiency and generality. This work presents these tradeoffs in the tensor network formalism and formulates 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

1 Introduction

Artificial intelligence is a long-standing dream, which has in recent years received enormous attention, especially driven by breakthroughs in large language models. Among the key priorities towards an economic and trustworthy usage is the improvement of efficiency and explainability of models.

Instead of post-hoc explainability of a models given inference on specific data, this work aims at the intrinsic human understandability of a model. We are motivated by the theory of logic, whose formalization of human thoughts serves as an interface between mechanized reasoning on a machine and human understandability. 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 need for efficiency stems more from economic concerns on the resource demand of training and infering a model. Tensors naturally represent states of systems with multiple variables, both in logical and probabilistic approaches towards artificial intelligence. However, even for a moderate numbers of variables, the curse of dimensionality prevents a typical machine's memory to store a generic representation. Carefully designing representation formats is therefore crucial to prevent exponential storage growth while balancing expressivity and efficiency.

In this work, we utilize the formalism of tensor networks in the creation of efficient representation schemes. The chosen tensor network formats are motivated as explainable model architectures and provide a synergy between the aims of efficiency and explainability. More precisely, tensor networks appear as the numerical structures behind probabilistic graphical models and logical knowledge bases. Understanding these foundations of tensor networks reveals their vast application potential in neuro-symbolic artificial intelligence.

1.1 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.

1.1.1 Logic and Explainability in AI

The logical tradition of artificial intelligence is motivated by the resembling of human thought in logics McCarthy (1959). Historic approaches towards artificial intelligence have focused on models by vast knowledge bases and inference by logical reasoning. The main problems hindering the success of this approach is the inability of classical first-order logic to handle uncertainty of information, as present in realistic scenarios.

Integrating observed data into a learning process has been framed Inductive Logic Programming Muggleton and De Raedt (1994). Along that line the Amie method Galárraga et al. (2013) has been developed to learn Horn clauses using a refinement operator. Class Expression Learning Lehmann et al. (2011) is a more recent approach to assist in the design of reasoning capabilities in Knowledge Graphs. However, this approach is limited by the expressivity of description logics and the exponentially large hypothesis sets for the choice of formulas. Efficient search methods in these exponentially large hypothesis sets have been provided based on reinforcement learning Demir and Ngonga Ngomo (2021) and neural networks Kouagou et al. (2022, 2023).

Logical approaches are still dominant in the description of data. Here the semantic web initiative developed data storage formats based on Knowledge Graphs Antoniou et al. (2012); Hogan et al. (2021), which describe structured data based on description logic.

Towards extenting the practical usage of logics, the field of Statistical Relational AI Nickel et al. (2016); Getoor and Taskar (2019) studies statistical models of logical relations. This directly treats uncertainty and therefore unifies logics with statistical approaches. This aims have more recently reframed as neuro-symbolic AI Hochreiter (2022); Sarker et al. (2022), with close relations to statistical relational AI Marra et al. (2024). Neuro-symbolic AI focuses on the unification of the neural and the symbolic paradigm Garcez et al. (2019), where early approaches are Towell and Shavlik (1994); Avila Garcez and Zaverucha (1999). While the symbolic paradigm is roughly understood as human understandable reasoning in formal logics, the neural paradigm is the computational benefit of decomposing a model into layerwise computation. These decompositions provide both expressive and efficient to train and infer model architectures. While modern black-box AI focuses on large neural networks, which size prevents human understanding of the inference process, neuro-symbolic AI aims at a re-implementation of the symbolic paradigm into such architectures.

1.1.2 Tensor Networks in AI

Decomposition schemes of tensors have been developed in numerics to efficiently operate in high-dimensional tensor spaces Hackbusch (2012) and to avoide the curse of dimensionality Bellman (1961). Each decomposition schemes has a graphical depiction, as we will introduce in Chapter 2, and decompositions are therefore refered to as networks. The first decomposition schemes by Tucker, originally introduced in Hitchcock (1927), suffered from exponential increases of the degrees of freedom with the tensor order. The CP format (see Chapter 15) can in principle establish storage in linear with the order. Sets of tensors with fixed rank with respect to this format are however not closed Beylkin and Mohlenkamp (2005) and approximation problems are often ill posed de Silva and Lim (2008). The Tensor Train decomposition Oseledets and Tyrtyshnikov (2009), which appears in quantum mechanics as matrix product states Perez-Garcia et al. (2007), overcomes these numerical problems Holtz et al. (2012). Hierarchical Tucker decompositions Hackbusch and Kühn (2009) are generalizations of tensor train decompositions, which have useful properties for tensor approximations Grasedyck (2010); Falco and Hackbusch (2012).

Tensors are used in the processing of big data Cichocki (2014) and in many-body physics Orús (2019). Besides that, there have been pioneering approaches to exploit them in the data-driven identification of governing equations Gelß et al. (2019); Goeßmann et al. (2020), more general supervised learning Stoudenmire and Schwab (2016) and the simulation of noisy quantum mechanics Sander et al. (2025). The duality between tensor networks and graphical models has been first discussed in Robeva and Seigal (2019) and motivated further expressivity studies such as Glasser et al. (2019). Tensor Networks have further been applied for batch logical inference Sakama et al. (2017); Sato (2017); Tsilionis et al. (2024). Whereas these are conceptual interesting approaches, they have so far been limited to matrix multiplication, whereas obvious expressivity benefits would come from more general contraction schemes. Similar ideas have been led to TensorLog Cohen et al. (2020), Real Logic Serafini and d'Avila Garcez (2016) and based on that Logical Tensor Networks Badreddine et al. (2022).

Further, sparse representation of knowledge graphs by tensor networks has motivated several embedding schemes for objects in the knowledge graph. The sparse decomposition of the adjacency tensor capturing the ternary relations between objects provides embeddings schemes encoding relations between the objects in a latent space. The specific approaches distinguish between the format used, where Nickel et al. (2011) and Balazevic et al. (2019) used tucker decompositions, Yang et al. (2015) the CP decomposition and Trouillon and Nickel (2017) complex extensions. Beyond embeddings, tensor based storage of knowledge graphs has recently shown tremendous improvements in querying knowledge graphs Bigerl et al. (2020). Here, queries on the knowledge graph are performed as contractions of the tensors efficiently representing knowledge graphs.

Tensors further serve as a central object in large-scale machine learning libraries such as TensorFlow Abadi et al. (2016) and PyTorch Paszke et al. (2019). Layerwise execution of neural network inference amounts then to tensor network contractions of tensors storing the activation of previous layers and weights. Beyond providing a central framework for the software design, also the design of AI-dedicated hardware orients on tensor contractions, with a current focus on Tensor Processing Units (TPU) Nikolić et al. (2022); Jouppi et al. (2023). Both the dedicated software and hardware design exploits the parallelization potential rooted in the contraction formalism of tensor networks.

1.1.3 Representation Schemes of Systems

We start with ontological commitments in the description of a system and follow the book Russell and Norvig (2021) 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 Richardson and Domingos (2006), which are instantiated on systems with differing number of objects. We will discuss those in Chapter 12.

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.

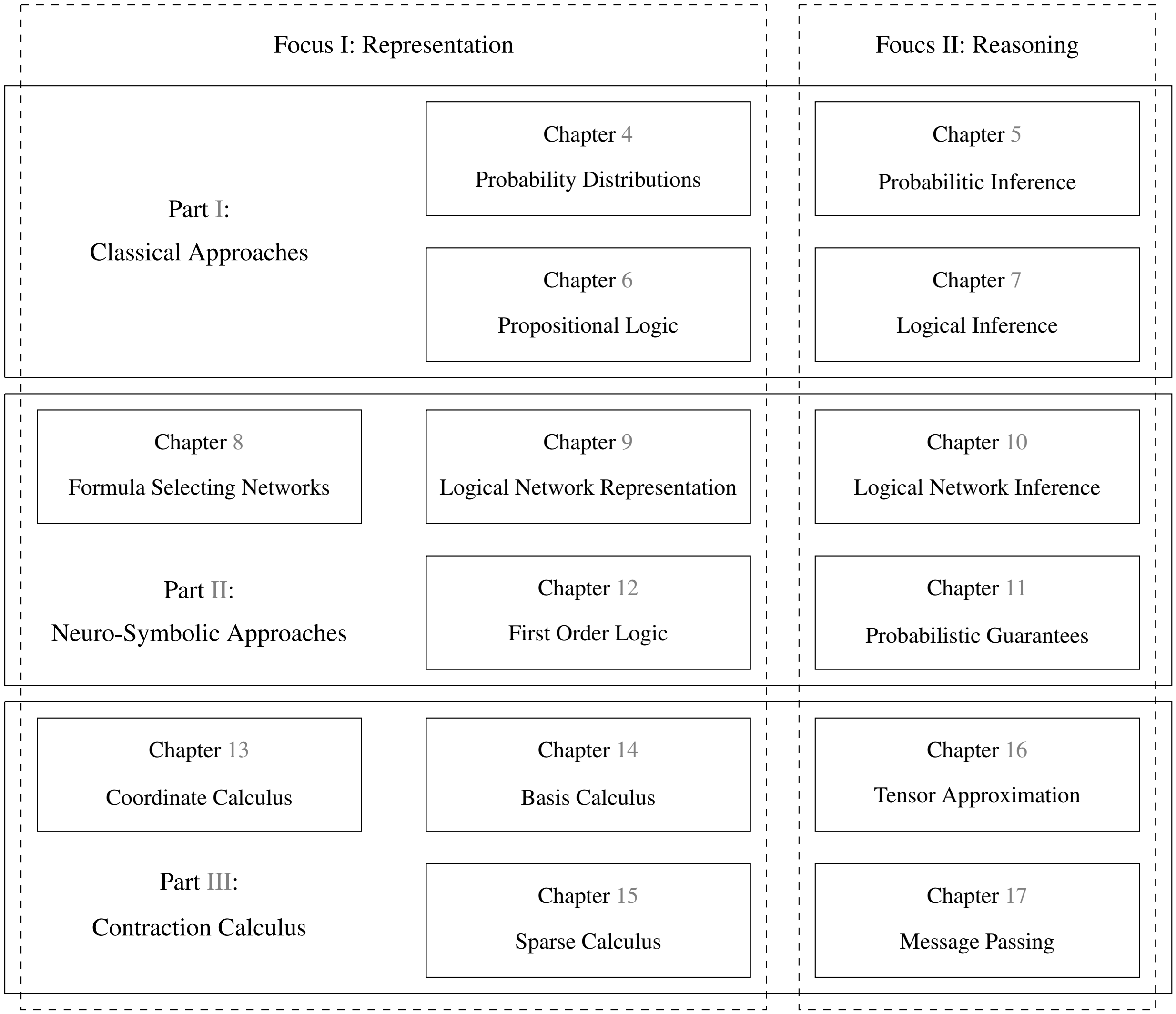


Figure 1: Sketch of the structure of this work. We assign the chapters to three parts and two focuses. The parts distinguish the coarse topics of this work into classical, neuro-symbolic approaches and the applied contraction calculus. The assigned focuses indicate whether the chapter orients more onto a representation format of the respective concepts or onto its exploitation in reasoning.

1.2 Structure of the work

The chapters are structured into three parts, and two focuses, see Figure 1.

1.2.1 Part I: Classical Approaches

The probabilistic and logical approaches towards artificial intelligence are reviewed in the tensor network formalism. We in this part restrict the discussion to atomic and factored system representation. In probability theory (see Chapter 4 and Chapter 5), tensors appear as generalized truth tables, storing the joint probability of each possible state of a system in factored representation. Tensors describing such distributions are of non-negative coordinates and are normed, which we will formalize by directed edges of hypergraphs. Applying the formalism, we introduce marginalization and conditioning operations based on contractions, and show how assumptions such as conditional independence lead to network decompositions. We then study the formalism of exponential families of probability distributions, which generalizes probabilistic graphical models. For generic exponential families we provide in Chapter 4 a tensor network representation, which structure is exploited for inference in Chapter 5. In logics (see Chapter 6), we motivate boolean tensors as a natural representation of propositional semantics. Logical entailment is then in Chapter 7 decided based contractions of these tensors, which we will further relate with marginal distributions in probabilistic inference.

The syntax of propositional logics thereby hints at efficient decompositions schemes of these semantic representing tensors. We exploit the syntax to find efficient tensor network decompositions of the tensors in Chapter 6 and use them for efficient logical inference algorithms in and Chapter 7.

1.2.2 Part II: Neuro-Symbolic Approaches

Motivated by the classical approaches we apply the tensor network formalism towards learning and infering neuro-symbolic models. We understand the decomposition of tensors into networks as an implementation of the neural paradigm of AI. Further, the symbolic paradigm is eminent in the interpretation of tensor networks using logical syntax, and enables the human-interbretable verbalization of learned models. Motivated by this central thoughts, we present vast classes of interpretable models in Chapter 9, which are unifying the logical and probabilistic approaches studied in Part I. The central idea here is to leverage the formalism of exponential families by choosing base measures and statistics based on logical formulas. We then turn in Chapter 10 towards inductive learning scenarios in this formalism, where new features are to be learned from data and parameters are calibrated. Here we apply the parametrization schemes developed in Chapter 8 to represent hypothesis classed for new features. While these approaches rely on propositional logics, in Chapter 12 we extend towards more expressive first-order logics. With knowledge graphs serving as examples we therein provide a tensor-network formalism to capture queries and motivate our learning schemes in propositional logics based on queries on random first-order worlds. In Chapter 11 we further derive statistical guarantees on these learning methods given random data, based on probabilistic bounds on uniform concentration events.

1.2.3 Part III: Contraction Calculus

In Part III the applied schemes of calculus using tensor network contractions are investigated in more detail. In particular, we distinguish between the schemes of coordinate, basis and sparse calculus.Coordinate calculus will be discussed in Chapter 13 using one-hot encodings as orthogonal basis elements. We will further properties related to directed tensors and a generic version of the Hammersley-Clifford decomposition theorem, which have been applied in the probabilistic approached in Part I. Basis calculus in Chapter 14 introduces generic encodings of subsets, relations and functions by boolean tensors used in previous parts. We show, that these encoding schemes translate function compositions into tensor network contractions and are therefore a central technique to execute batchwise function evaluation by efficient tensor network contractions. In Chapter 15 we provide sparse schemes oriented on the CP format for the storage of tensors. We further investigate the origins of sparsity based on encodings of functions, and provide rank bounds for summations and contractions of these tensors. Then we formalize optimization problems as maximal coordinate searched among tensors and relate the investigated CP formats with standard optimization frameworks. We continue with studies of tensor approximation in Chapter 16, where we adapt formula selecting networks of Chapter 8 to select sparse CP tensors. In Chapter 17 we then investigate schemes of efficient contraction calculus based on local contractions, which are passed through the network as messages. These schemes can be regarded as generic numerical tools underlying message passing schemes such as belief propagation in probability theory and constraint propagation in logics.

1.2.4 Focus I: Representation

In this focus, we motivate and investigate the efficient representation of tensors based on tensor network decompositions, where formats are captured by hypergraph as we introduce in Chapter 2. Besides being a necessity to overcome the curse of dimensionality, we show in Part I multiple motivations of tensor network decompositions originating from principles of artificial intelligence. As such, decompositions originate from conditional independence assumptions on probability distributions (see Chapter 4) and from logical syntax (see Chapter 6). Towards neuro-symbolic AI, we provide in Chapter 8 a generic representation scheme for batches of logical formulas. This scheme introduces additional axes to a tensor, which are assigned with selection variables and which slices select specific tensors. We exploit this scheme in Chapter 9 for efficient representation of exponential families, which statistics are sets of logical formulas. In Part III we investigate the applied representation scheme from a more theoretical viewpoint. More precisely, we distinguish between the schemes of coordinate calculus (Chapter 13) and basis calculus (Chapter 14). These schemes differ in the exploitation of the real coordinates of a tensor or of sums over chosen basis elements, in the encoding of information. In Chapter 15 we define restricted CP decompositions of tensors for sparse representations of d-ary relations, which appear in sparse representation of relational databases.

1.2.5 Foucs II: Reasoning

We develop schemes to efficiently perform inductive and deductive reasoning based on information stored in decomposed tensor. Contractions of tensor networks representing models in artificial intelligence are the central scheme to retrieve information. While in probability theory contractions compute marginal distribution (see Chapter 5), contraction of logical formulas are model counts central to the formalism of logical entailment (see Chapter 7). We will further exploit them to calculate queries in first-order logic such as on knowledge graphs (see Chapter 12). The statistical foundation on the success of contraction-based learning, which lies in the phenomenon of uniform concentration of contractions with empirical random tensors, will be investigated in Chapter 11. In Part III we further study generic tools for efficient execution of contraction-based reasoning. The tensor network approximation schemes in Chapter 16 bear the potential to approximate reasoning tasks by more efficient ones. The efficient execution of contractions using message-passing algorithms in Chapter 17 have been exploited in a variety of exact and approximated reasoning schemes.

2 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.

2.1 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 1. *An atomic representation of a system is described by a categorical variables X taking values x in a finite set*

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

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

2.2 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 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*

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

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

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

Tensors $T[X_0, \dots, X_{d-1}]$ 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 $X_0, \dots, X_{d-1}$ of categorical variables by $X_{[d]}$, that is denote $T[X_0, \dots, X_{d-1}]$ 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 1 (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* $x_0, \dots, x_{d-1} \in \bigtimes_{k\in[d]} [m_k]$

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

2.3 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 4 (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*

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

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

$$e_x[X] = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T\,, \tag{2}$$

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.

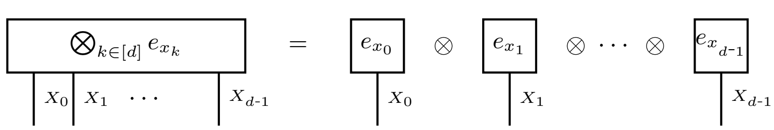
Definition 5 (One-hot encodings to Factored Representations)**.** *Let there be a factored system defined by a tuple* $(X_0, \dots, X_{d-1})$ *of variables taking values in* $\bigtimes_{k\in[d]} [m_k]$*. The one-hot encoding of its states is the tensor product of the one-hot encoding to each categorical 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*

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

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



In Chapter 13 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 1 (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.*

2.4 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.

2.4.1 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 6. *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 $e^{\mathrm{in}}, e^{\mathrm{out}} \subset \mathcal{V}$, that is*

$$e = \left(e^{\mathrm{in}}, e^{\mathrm{out}}\right).$$

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 2a). 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 Pearl (1988), 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 Pearl (2009).

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. This depiction scheme has been established in the literature as wiring diagrams (see Landsberg (2011) and dates back at least to the work Penrose (1987).

Both depiction schemes are simplifications of factor graphs, by highlighting the categorical variables in the depiction in Figure 2b) and the tensors in the depiction in Figure 2c). We in this work will prefer the simplification of the tensor network community, depicted in Figure 2b).

In another interpretation (see Robeva and Seigal (2019)), both simplification schemes are different hypergraphs, which are dual to each other.

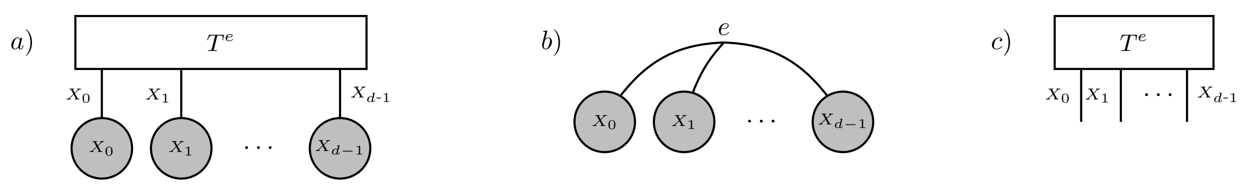
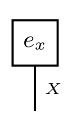


Figure 2: Depiction of Tensors a) As a factor in a factor graph, depicted by a block, and connected to categorical variables assigned to nodes. b) Highlighting only the variable dependencies by a hyperedge connecting the variables X_k to each axis $k \in [d]$. c) Highlighting the tensor by a blockwise 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.

Figure 3: Example of a tensor network on a a) hypergraph with edges $e_0 = \{X_0, X_1, X_2\}$, $e_1 = \{X_1, X_2\}$ and $e_2 = \{X_2, X_3\}$, which is decorated by the tensor cores b), representing a contraction with leaving all variables open.

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

Definition 7. *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

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

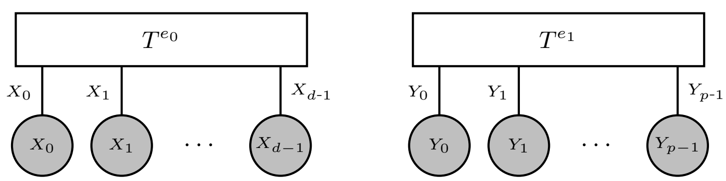
where we denote by X_e the set of categorical variables X_v with $v \in e$. Then we call the set

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

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

2.4.2 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 8 (Tensor Product)**.** *Let there be two tensor*

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

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

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

Other popular standard notations of tensor products (see Kolda and Bader (2009); Hackbusch (2012); Cichocki et al. (2015))

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

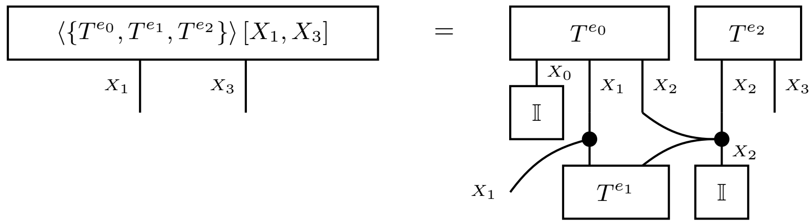


Figure 4: Example of a tensor network contraction of all but the variables X_1, X_3. Contraction of variables can always be depicted by closing the open legs with trivial tensors $\mathbb{I}$ performing index sums.

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 1), we further make a notation convention to omit that tensor, that is

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

2.4.3 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 9. *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*

$$\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\tilde{\mathcal{V}}}] \in \bigotimes_{v\in\tilde{\mathcal{V}}} \mathbb{R}^{m_v} \tag{3}$$

defined coordinatewise by the sum

$$\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}}\in \times_{v\in\mathcal{V}/\tilde{\mathcal{V}}}[m_v]} \left(\prod_{e\in\mathcal{E}} T^e\left[X_e = x_e\right]\right) . \tag{4}$$

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 2).

Remark 2 (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. 9, 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.*

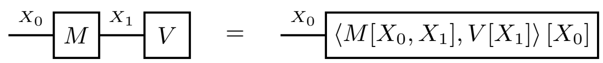
Further notations without usage of axis variables are mode products (see Kolda and Bader (2009); Hackbusch (2012); Cichocki et al. (2015)), often denoted by the operation $\times_n$. With our more generic variable-based notations, we can capture these more specific contractions by coloring the tensor 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 2. *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:



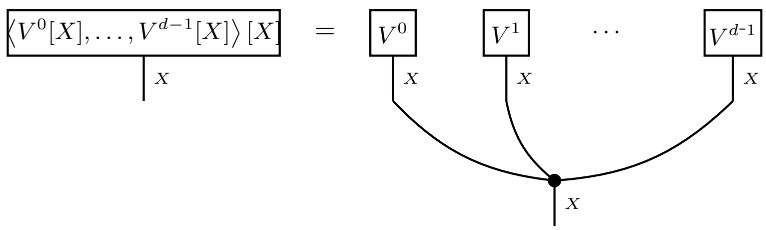
Example 3. *Hadamard products of vectors A node appearing in arbitrary many hyperedges denotes a Hadamard product of the axis of the respective decorating tensors. To give an example, let $V^k[X] \in \mathbb{R}^m$ be vectors for $k \in [d]$. Their hadamard product is the vector*

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

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:



2.4.4 Decompositions

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

Definition 10. *A Tensor Network Decomposition of a tensor $T\left[X_{\mathcal{V}}\right]$ is a Tensor Network $\mathcal{T}^{\mathcal{G}}$ such that*

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

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

2.5 Properties of Tensors

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

Definition 11. *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 12. *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 $\mathcal{V}^{\mathrm{in}}$ and outgoing variables $\mathcal{V}^{\mathrm{out}}$, where $\mathcal{V} = \mathcal{V}^{\mathrm{in}} \dot{\cup} \mathcal{V}^{\mathrm{out}}$, when

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

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

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

Definition 13. *A tensor $T\left[X_{\mathcal{V}}\right]$ is said to be normable on $\mathcal{V}^{\mathrm{in}} \subset \mathcal{V}$, if for any $x_{\mathcal{V}^{\mathrm{in}}} \in \bigtimes_{v \in \mathcal{V}^{\mathrm{in}}} [m_v]$ we have*

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

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

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

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

We will investigate the contractions of directed tensors in Part III, where we show in Theorem 91 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:

2.6 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.

2.6.1 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 14, see Def. 79.

Definition 14 (Relation encoding of maps between Factored Systems)**.** *Let f be a function*

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

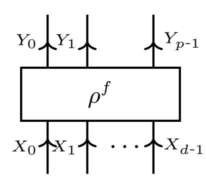
which maps the states of a factored system to variables $X_0, \dots, X_{d-1}$ to the states of another factored system with variables $Y_0, \dots, Y_{p\text{-}1}$. Then the tensor representation of f is a tensor

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

defined by

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

We depict relational encodings by directed tensors:



2.6.2 Tensor-valued functions

Definition 15 (Selection encoding of Maps between Factored Systems)**.** *Given a tensor space $\bigotimes_{s\in[n]} \mathbb{R}^{p_s}$ described by categorical variables $L_0, \dots, L_{n-1}$ 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

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

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.

3 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 4 we introduce:

- Marginal probabilities (Def. 19, The. 1)
$$\mathbb{P}[X_0] = \langle \mathbb{P} \rangle [X_0]$$
- Conditional probabilities (Def. 20, The. 2)
$$\mathbb{P}[X_0|X_1] = \langle \mathbb{P} \rangle [X_0|X_1]$$
- The probability distribution of a Markov Network is (Def. 23)
$$\mathbb{P}^{\mathcal{T}^{\mathcal{G}}} = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\mathcal{V}|\varnothing]$$
The partition function of a Markov Networks
$$\mathcal{Z}\left(\mathcal{T}^{\mathcal{G}}\right) = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing]$$
Bayesian Networks (Def. 27), 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. 21, The. 6)
$$\langle \mathbb{P} \rangle [X_0, X_1] = \langle \mathbb{P} \rangle [X_0] \otimes \langle \mathbb{P} \rangle [X_1]$$
- X_0 and X_1 are called independent conditioned on X_2 when (Def. 22, The. 7)
$$\langle \mathbb{P} \rangle [X_0, X_1|X_2] = \langle \mathbb{P} \rangle [X_0|X_2] \otimes \langle \mathbb{P} \rangle [X_1|X_2]$$

In Chapter 6 we introduce:

- Propositional formulas by boolean tensors (Def. 40)
$$f\left[X_{[d]}\right] : \bigtimes_{k\in[d]} [2] \to [2] \subset \mathbb{R}\,.$$
- Syntactical representation of formulas corresponding with tensor networks of boolean tensors (The. 35)

Part I
Classical Approaches

4 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.

4.1 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.

Definition 16 (Probability Tensor). *Let there be for each $k \in [d]$ a categorical variable X_k taking values in $[m_k]$. A joint probability distribution of these categorical variables is a tensor*

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

such that

$$\langle \mathbb{P}[X_0, \ldots, X_{d-1}] \rangle [\varnothing] = 1 \, .$$

The probability tensor to the distribution is a tensor

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

where we again use the abbreviation $X_{[d]}$ for the list of variables $X_0, \ldots, X_{d-1}$. The tensor can be decomposed as the sum (see Lem. 18 in Part III for more details)

$$\mathbb{P}\left[X_{[d]}\right] = \sum_{x_{[d]}\in \bigtimes_{k\in[d]}[m_k]} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] \cdot e_{x_{[d]}}\left[X_{[d]}\right] \, ,$$

where we understand $\mathbb{P}\left[X_{[d]} = x_{[d]}\right]$ as the probability of the categorical variables to take the state $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$. The normation condition $1 = \left\langle \mathbb{P}\left[X_{[d]}\right] \right\rangle [\varnothing]$ has a more convenient equivalence by the coordinate sum

$$1 = \left\langle \mathbb{P}\left[X_{[d]}\right] \right\rangle [\varnothing] = \sum_{x_{[d]}\in \bigtimes_{k\in[d]}[m_k]} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] \, ,$$

and thus ensures that all probabilities sum to 1, which is necessary for the probabilistic interpretation. While the assumptions of non-negative coordinates in Def. 16 reflects the first probability axiom of Kolmogorov, the assumption of contraction 1 implements the second axiom (see for example DeGroot (2016)). Since probability distributions contract to 1, they are directed (see Def. **??**) with all distributed variables outgoing and empty incoming variables (see Figure 5).

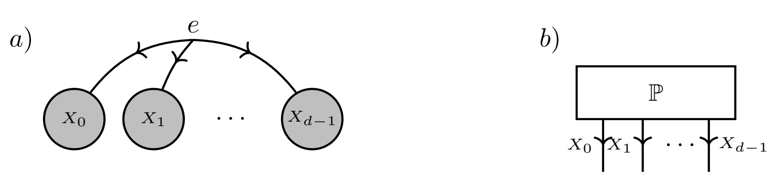


Figure 5: Probability distributions of variables $X_0, \ldots, X_{d-1}$, sketched a) by a directed edge e with all variables outgoing, which is decorated b) by a directed tensor $\mathbb{P}\left[X_{[d]}\right]$.

4.1.1 Base measures

From a measure theoretic perspective, probabilities are measurable functions called probability densities, which integrals are 1 (see for example DeGroot (2016)). 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 $\mathbb{P}\left[X_{[d]}\right]$ as a probability density with respect to $\nu\left[X_{[d]}\right]$, any probabilistic interpretation will be through the contraction $\left\langle \mathbb{P}\left[X_{[d]}\right], \nu\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right]$ and the normation condition reads as

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

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 17. *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 $\mathbb{P}\left[X_{[d]}=x_{[d]}\right]=0$. We denote the set of by ν representable distributions by $\Gamma^{\delta,\nu}$.*

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

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

and can therefore safely ignore the base measures. This enables the characterization of by ν representable distributions by

$$\Gamma^{\delta,\nu}=\left\{\mathbb{P}\left[X_{[d]}\right]\,:\,\forall x_{[d]}\in\bigtimes_{k\in[d]}[m_k]:\mathbb{P}\left[X_{[d]}=x_{[d]}\right]\geq 0,\,\left\langle \mathbb{P}\left[X_{[d]}\right],\nu\left[X_{[d]}\right]\right\rangle\left[X_{[d]}\right]=\mathbb{P}\left[X_{[d]}\right]\right\}.$$

Starting with Chapter 6 we will further investigate boolean tensors and relate them with propositional formulas. In Chapter 7 we will connect the representation and positivity with respect to boolean base measures with the formalism of entailment. The notation $\Gamma^{\delta,\nu}$ of by ν representable distributions will later in Chapter 9 relate to minterm exponential families introduced therein.

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 find 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 7 orders based on entailment and show the minimality in The. 44. Here, we just introduce the minimality criterion as positivity of a distribution with respect to a base measure.

Definition 18. *We say that a probability distribution $\mathbb{P}\left[X_{[d]}\right]$ is positive with respect to a boolean base measure $\nu\left[X_{[d]}\right]$, if the distribution is representable by ν (i.e. $\langle\mathbb{P},\nu\rangle[\varnothing]=1$) and for all $x_{[d]}$ with $\nu\left[X_0=x_0,\ldots,X_{d-1}=x_{d-1}\right]=1$ we have $\mathbb{P}\left[X_0=x_0,\ldots,X_{d-1}=x_{d-1}\right]>0$.*

4.2 Marginal Distribution

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

Definition 19 (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 by the contraction

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

To connect with a more standard defining equation of marginal distributions, let us notice that for any $x_0\in[m_0]$

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

Thus, each coordinate of the marginal distribution is the sum of the joint probability of compatible states. We say that the variable X_1 is marginalized out, when building the marginal distribution $\mathbb{P}\left[X_0\right]$ of X_0. Let us now justify this terminology and show, that any marginal distribution is a probability distribution as introduced in Def. 16.

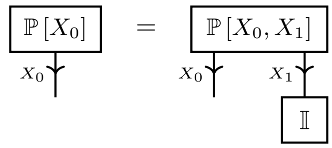
Theorem 1. *Any marginal distribution is a probability distribution.*

Proof. We further have that any marginal distribution is normed, since by the commutativity of contractions (see for more details The. 121 in Part III)

$$\left\langle\mathbb{P}\left[X_0\right]\right\rangle[\varnothing]=\left\langle\left\langle\mathbb{P}\left[X_0,X_1\right]\right\rangle\left[X_0\right]\right\rangle[\varnothing]=\left\langle\mathbb{P}\left[X_0,X_1\right]\right\rangle[\varnothing]=1.$$

Further any coordinate is non-negative, since it is a sum of non-negative coordinates. It follows from Def. 16, that any marginal distribution is a probability distribution. □

In a tensor network diagram we often represent variables X_1 not appearing as open variables of a contraction as contracted with the trivial tensor $\mathbb{I}\left[X_1\right]$. Following this notation, we depict the marginal distribution in Def. **??** by



Since we have shown, that marginal distributions are themself probability distributions, they inherit the outgoing directionality in tensor network diagrams.

We notice, that Def. 19 generalizes to marginalizations of arbitrary sets of variables, when having a distribution $\mathbb{P}\left[X_{[d]}\right]$ of an arbitrary number of categorical variables. It suffices for this to interpret X_0 and X_1 as collections of variables, which indices take the states of the respective factored systems.

4.3 Conditional Probabilities

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

Definition 20 (Conditional Probability)**.** *Let $\mathbb{P}[X_0, X_1]$ be a distribution of the categorical variables X_0 and X_1, such that $\mathbb{P}[X_0, X_1]$ is normable on $\{X_1\}$. Then the distribution of X_0 conditioned on X_1 is defined by*

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

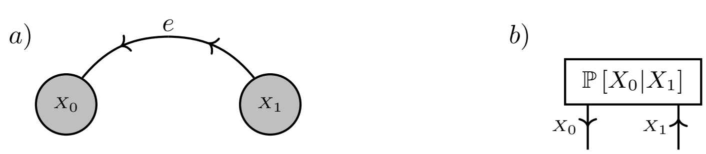
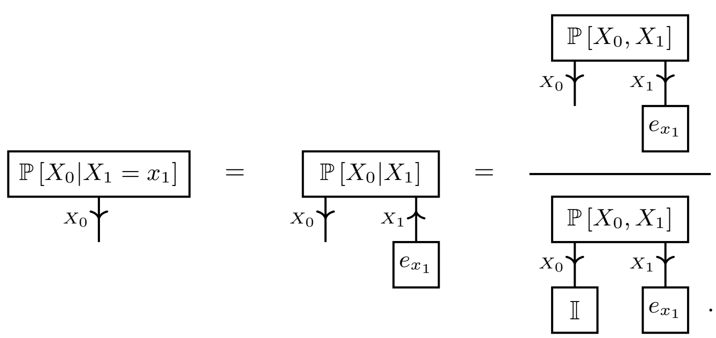


Figure 6: Depiction of conditional probability distributions a) by an edge with the incoming variable X_1 and the outgoing variable X_1, which is decorated by b) the directed tensor $\mathbb{P}[X_0|X_1]$.

Since conditional probabilities are normations of probability tensors, they are directed and therefore depicted by directed hyperedges (see Figure 6). For any $x_1 \in [m_1]$ we depict the slice $\mathbb{P}[X_0|X_1 = x_1]$ defined by a normation operation as



As we have done before for marginal distribution, we relate Def. 20 with a more convenient coordinatewise definition of conditional probabilities. For any indices $x_0 \in [m_0]$ and $x_1 \in [m_1]$ we have

$$\mathbb{P}[X_0 = x_0|X_1 = x_1] = \frac{\mathbb{P}[X_0 = x_0, X_1 = x_1]}{\langle \mathbb{P}[X_0, X_1] \rangle [X_1 = x_1]} = \frac{\mathbb{P}[X_0 = x_0, X_1 = x_1]}{\sum_{x_0 \in [m_0]} \mathbb{P}[X_0 = x_0, X_1 = x_1]} \ .$$

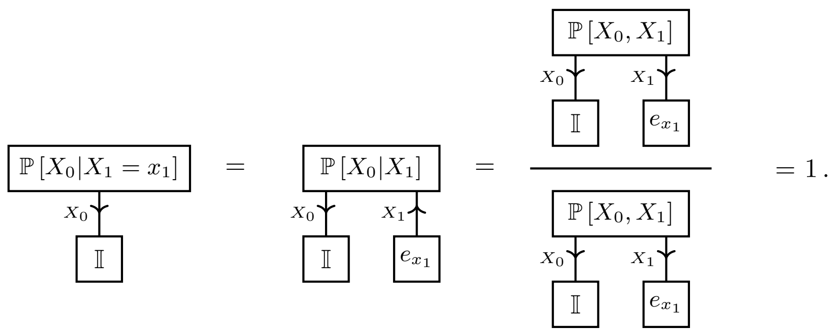
The distribution of X_0 conditioned on X_1 is the normed collection of slice of the probability distribution $\mathbb{P}[X_0, X_1]$. Each slice of the conditioned distribution with respect to incoming variables is a probability distribution itself, as we show next.

Theorem 2. *For any $x_1 \in [m_1]$ the tensor $\mathbb{P}[X_0|X_1 = x_1]$ is a probability tensor.*

Proof. As a normation of a non-negative tensor, the conditional probability $\mathbb{P}[X_0|X_1 = x_1]$ and any of its slices is also a non-negative tensor. Further, we have for any $x_1 \in [m_1]$

$$\begin{aligned}\langle \mathbb{P}[X_0|X_1 = x_1] \rangle [\varnothing] &= \sum_{x_0 \in [m_0]} \mathbb{P}[X_0 = x_0|X_1 = x_1] \\ &= \frac{\sum_{x_0 \in [m_0]} \mathbb{P}[X_0 = x_0, X_1 = x_1]}{\sum_{x_0 \in [m_0]} \mathbb{P}[X_0 = x_0, X_1 = x_1]} \\ &= 1\,,\end{aligned}$$

and therefore each slice is normed. We can visualize this calculation exploiting our diagramatic notation as



Since for any $x_1 \in [m_1]$ the slice $\mathbb{P}[X_0|X_1 = x_1]$ is non-negative and contracts to 1, we conclude that it is a probability distribution. ☐

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

Theorem 3. *Any tensor with non-negative coordinates is a conditional distribution tensor, if and only if it is directed with the condition variables incoming and the other outgoing.*

Proof. "$\Rightarrow$": By The. 2 a conditional probability tensor $\mathbb{P}[X_0|X_1]$ is the normation of a tensor and by The. 91 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.

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

$$\mathbb{P}[X_\mathcal{V}] = \frac{1}{\prod_{v \in \mathcal{V}^{\text{in}}} m_v} \cdot T[X_\mathcal{V}] \tag{5}$$

is a probability tensor, since

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

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

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

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

The. 3 specifies a broad class of tensors to represent conditional probabilities. In combination with The. 97, which states that relational encodings are directed, we get that any relational encoding of a function is a conditional probability tensor.

4.4 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 4 (Bayes Theorem). *For any probability distribution $\mathbb{P}[X_0, X_1]$ with positive $\mathbb{P}[X_1]$ we have*

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

Proof. This theorem follows from the more generic contraction equation The. 92 to be shown in Chapter 13. We note that by positivity of $\mathbb{P}[X_1]$, the tensor network $\mathbb{P}$ is normable with respect to X_1. The. 92 therefore implies choosing $\mathcal{V} = \{0, 1\}$, $\mathcal{V}^{\text{in}} = \{1\}$ and $\mathcal{V}^{\text{out}} = \{0\}$, that For our tensor

$$\begin{aligned}\mathbb{P}[X_0, X_1] &= \langle \langle \mathbb{P}[X_0, X_1] \rangle [X_0|X_1], \langle \mathbb{P}[X_0, X_1] \rangle [X_1] \rangle [X_0, X_1] \\ &= \langle \mathbb{P}[X_0|X_1], \mathbb{P}[X_1] \rangle [X_0, X_1] .\end{aligned}$$

□

Following the insight of the Bayes The. 4, probability distributions of arbitrary numbers of variables can be decomposed as a contraction of conditional probabilities, as we show in the next theorem.

Theorem 5 (Chain Rule). *For any probability distribution $\mathbb{P}\left[X_{[d]}\right]$ we have*

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

provided that all conditional probability distributions exist.

Proof. The claim can be derived by an iterative application of the Bayes The. 4 theorem. We will proof this statement in more generality in Chapter 13 as The. 93, deducing it from the generalization of the Bayes The. 4 by The. 92. The claim here then follows from The. 93 using $\mathcal{V} = [d]$ and $T[X_{\mathcal{V}}] = \mathbb{P}\left[X_{[d]}\right]$, since

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

□

We observe, that the chain rule provides a generic decomposition scheme of probability distributions into conditional distributions. The conditional distribution to $k = d - 1$, which appears in the chain decomposition, is in the same tensor space as the decomposed distribution $\mathbb{P}\left[X_{[d]}\right]$. To achieve our main goal of tensor network decompositions, which is an efficient storage format of the decomposed tensor, we need to further sparsify the appearing conditional probabilities (to be more precise, we aim at basis+ CP decompositions, to be introduced in Chapter 15). These simplification require additional assumptions on the distribution, which we will introduce in the next section.

4.5 Independent Variables

Independence leads to severe sparsifications of conditional probabilities and is therefore the key assumption to gain sparse decompositions of probability distributions. Before showing such decomposition schemes, we first provide a coordinatewise definition of independent variables.

Definition 21 (Independence). *We say that X_0 is independent of X_1 with respect to a distribution $\mathbb{P}[X_0, X_1]$, if for any values $x_0 \in [m_0]$ and x_1 the distribution satisfies*

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

We state next an equivalent independence criterion based on a contraction equation of probability distributions.

Theorem 6 (Independence Criterion as a Contraction Equation). *The variable X_0 is independent from X_1 with respect to a probability distribution $\mathbb{P}[X_0, X_1]$, if and only if*

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

Proof. By The. 1 we know that marginal probabilities are equivalent to contracted probability distributions, i.e. $\mathbb{P}[X_0] = \langle\{\mathbb{P}\}\rangle[X_0]$. By orthogonality of one-hot encodings we have that

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

is equivalent to

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

We reorder the summations and arrive at

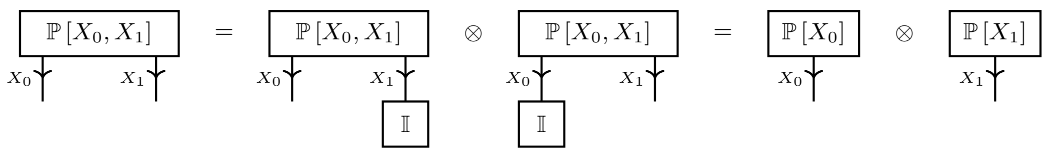
$$\sum_{x_0,x_1} \mathbb{P}[X_0 = x_0, X_1 = x_1] \cdot e_{x_0,x_1}[X_0, X_1] = \left(\sum_{x_0} \mathbb{P}[X_0 = x_0]\, e_{x_0}[X_0]\right) \cdot \left(\sum_{x_1} \mathbb{P}[X_0 = x_0] \cdot e_{x_1}[X_1]\right)$$

which is by Lem. 18 equal to the claim

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

$\square$

Two jointly distributed variables are by The. 6 independent, if and only if their joint distribution $\mathbb{P}[X_0, X_1]$ is the tensor product of marginal probabilities. Using tensor network diagrams we depict this property by



Let us notice, that the assumption of independence reduces the degrees of freedom from $m_0 \cdot m_1 - 1$ to $(m_0 - 1) + (m_1 - 1)$. The decomposition by marginal distributions furthermore exploits this reduced freedom and provides an efficient storage. Having a joint distribution of multiple variables, which disjoint subsets are independent, we can iteratively apply the decomposition scheme. As a result, the degrees of freedom scaling exponential in the number of distributed variables would be reduced to a linear scaling, by the assumption of independence.

Independence is, as we observed, a strong assumption, which is often too restrictive. It is furthermore an undesired property, when in a supervised learning scenario a target variable has to be predicted based on known feature variables. Conditional independence instead is a less demanding assumption, which still implies efficient tensor network decompositions schemes. We introduce conditional independence as independence of variables with respect to conditional distributions.

Definition 22 (Conditional Independence)**.** *Given a joint distribution of variables X_0, X_1 and X_2, such that $\mathbb{P}[X_2]$ is positive. We say that X_0 is independent of X_1 conditioned on X_2 if for any states $x_0 \in [m_0], x_1 \in [m_1]$ and $x_2 \in [m_2]$*

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

Conditional independence stated in Def. 22 has a close connection with independence stated in Def. 21. To be more precise, X_0 is independent of X_1 conditioned on X_2, if and only if X_0 is independent of X_1 with respect to any slice $\mathbb{P}[X_0, X_1 | X_2 = x_2]$ of the conditional distribution $\mathbb{P}[X_0, X_1 | X_2]$. Analogously to The. 6 for independence, we further find a decomposition criterion for conditional independence. Since conditional independence can be regarded as a property of conditional probabilities, this decomposition criterion also involves conditional probabilities.

Theorem 7 (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 of X_1 conditioned on X_2, if and only if the equation*

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

holds.

Proof. With the same argumentation as in the proof of The. 6, we notice that the contraction equation holds, if and only if for any $x_0 \in [m_0]$, $x_1 \in [m_1]$ and $x_2 \in [m_2]$

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

This is equivalent to conditional independence by Def. 22. $\square$

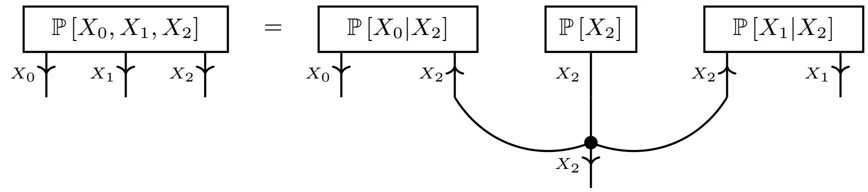


Figure 7: Diagrammatic visualization of the contraction equation in Cor. 1. Conditional independence of X_0 and X_1 given X_2 holds if the contraction on the right ride is equal to the probability tensor on the left side.

We can further exploit conditional independence to find tensor network decompositions of probabilities, as we show as the next corollary.

Corollary 1. *If and only if X_0 is independent of X_1 conditioned on X_2 the probability distribution $\mathbb{P}$ satisfies (see Figure 7)*

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

Proof. With the Bayes The. 4 it holds that

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

Decomposing the first tensor in the contraction, The. 7 implies, that X_0 is independent of X_1 conditioned on X_2, if and only if

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

□

Let us now recall our motivation of the study of conditional independence, namely to find sparsifications of conditional probabilities as those appearing in chain decompositions The. 5. As we state as the next theorem, such sparsifications follow from conditional independence.

Theorem 8. *Whenever X_0 is independent of X_1 given X_2, we have for any $x_1 \in [m_1]$*

$$\mathbb{P}[X_0|X_1 = x_1, X_2] = \mathbb{P}[X_0|X_2] .$$

Proof. By the Bayes The. 4 we have for any indices to the variables

$$\mathbb{P}[X_0 = x_0|X_1 = x_1, X_2 = x_2] = \frac{\mathbb{P}[X_0 = x_0, X_1 = x_1|X_2 = x_2]}{\langle \mathbb{P}[X_0, X_1 = x_1|X_2 = x_2] \rangle [\varnothing]}$$

If X_0 is independent of X_1 given X_2 it follows that

$$\begin{aligned}\mathbb{P}[X_0 = x_0|X_1 = x_1, X_2 = x_2] &= \frac{\mathbb{P}[X_0 = x_0|X_2 = x_2] \cdot \mathbb{P}[X_1 = x_1|X_2 = x_2]}{\langle \mathbb{P}[X_0, X_1 = x_1|X_2 = x_2] \rangle [\varnothing]} \\ &= \mathbb{P}[X_0 = x_0|X_2 = x_2] .\end{aligned}$$

□

Following our motivation of sparse decompositions, we now combine this result with the generic chain rule, to show Markov Chain decompositions.

Theorem 9 (Markov Chain)**.** *Let there be a set of variables X_k where $k \in [d]$, and let us denote for $k \in [d]$ by $X_{[k]}$ the collection of variables $X_0, \ldots, X_{k-1}$. Let us assume, that for any $k \in [d]$ with $k \geq 2$ the variable X_k is independent of $X_{[k-1]}$ conditioned on X_{k-1}, then*

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

We depict this decomposition in Figure 8.

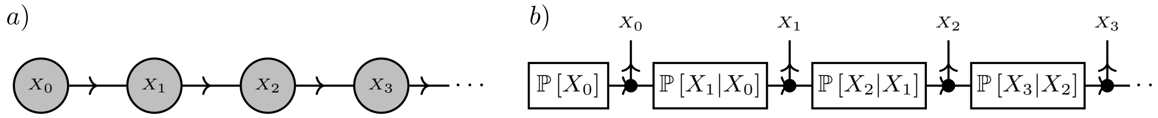


Figure 8: Depiction of a Markov Chain Decomposition by a a) hypergraph with the nodes $\mathcal{V} = [d]$ and edges $\mathcal{E} = \left\{\{0\} \cup \{\{k, k+1\} \, : \, k \in [d], \, k > 1\}\right\}$ and b) a decorating Tensor Network representing the sparsified conditional probabilities.

Proof. By the chain rule shown in The. 5 we have

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

Using that X_k is conditional independent of $X_{[k-1]}$ conditioned on X_{k-1} we further have by The. 8

$$\mathbb{P}\left[X_k | X_{[k]}\right] = \mathbb{P}\left[X_k | X_{k-1}\right] \otimes \mathbb{I}\left[X_{[k-1]}\right] \, .$$

Composing both equalities and omitting the trivial tensors shows the claim. □

The assumption of X_k being independent of $X_{[k-1]}$ conditioned on X_{k-1} is called the Markov property and the corresponding collection of random variables is called a Markov Chain. The. 9 states an efficient decomposition of the probability distribution into a concatenated product of matrices representing conditional probability distributions. Marginal distributions of Markov Chains can therefore consecutively be computed by matrix-vector products, that is for $k \in [d]$ with $k \geq 1$

$$\mathbb{P}\left[X_k\right] = \left\langle \mathbb{P}\left[X_k | X_{k-1}\right], \mathbb{P}\left[X_{k-1}\right] \right\rangle \left[X_k\right] \, .$$

The conditional probability matrices are therefore called stochastic transition matrices.

We notice, that the decomposition scheme of The. 9 hints at an efficient representation of $\mathbb{P}\left[X_{[d]}\right]$ based on transition matrices. While $\mathbb{P}\left[X_{[d]}\right]$ is a tensor in a space of dimension

$$\prod_{k \in [d]} m_k \, ,$$

the sum of the dimension of the transition matrices is

$$m_0 + \sum_{k \in [d], \, k \geq 1} m_k \cdot m_{k-1} \, .$$

We therefore observe a linear increase of the storage demand of the transition matrices in the order d, whereas a naive storage of $\mathbb{P}\left[X_{[d]}\right]$ by its coordinates would have an exponentially demand.

The Markov Chain serves as a toy example drawing on a restrictive chain arrangement of conditional independencies. In the following section, we will investigate decomposition schemes, which relax this assumption and draw on more general collections of conditional independencies. The computation of marginal distribution by consecutive transition matrix multiplications will then be replaced by more general tensor network contractions.

4.6 Graphical Models

Graphical models provide a more generic framework to relate conditional dependency assumptions on a distribution with tensor network decompositions. Following the tensor network formalism we in this section introduce graphical models based on hypergraphs. First, we study Markov Networks in most generality and then connect with conditional probabilities in the discussion of Bayesian Networks.

4.6.1 Markov Networks

We now define Markov Networks based on hypergraphs, to establish a direct connection with tensor network decorating the hypergraph. In a more canonical way, Markov Networks are instead defined by graphs, where instead of the edges the cliques are decorated by factor tensors (see for example Koller and Friedman (2009)).

Definition 23 (Markov Network). *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network of non-negative tensors decorating a hypergraph $\mathcal{G}$. Then the Markov Network $\mathbb{P}^{\mathcal{G}}$ to $\mathcal{T}^{\mathcal{G}}$ is the probability distribution of X_v defined by the tensor*

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

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 tensor network $\mathcal{T}^{\mathcal{G}}$.

The marginalization of a Markov Network to $\mathcal{T}^{\mathcal{G}}$ on subsets of variables $X_{\tilde{\mathcal{V}}}$ is

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

This can be derived from The. 121, 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

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

While we have directly defined Markov Networks as decomposed probability distributions, we now want to derive assumptions on a distribution assuring that such decompositions exist. As we will see, the sets of conditional independencies encoded by a hypergraph are captured by its seperation properties, as we define next.

Definition 24 (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 the more canonical graph-based definition of Markov Networks Koller and Friedman (2009).

Definition 25 (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. 11.

Theorem 10 (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. "$\Rightarrow$": 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

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

By The. 7, it now follows that X_A is independent of X_B conditioned on X_C.

"$\Leftarrow$": 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 (see Clifford and Hammersley (1971)), which we will proof later in Sect. 13.4 of Chapter 13. ☐

From the proof of The. 10 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 Koller and Friedman (2009)).

4.6.2 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. 6 by introducing the property of acyclicity for hypergraphs.

Definition 26. *A directed path is a sequence $v_0, \dots v_r$ such that for any $l \in [r]$ there is an hyperedge $e = (e^{\mathrm{in}}, e^{\mathrm{out}}) \in \mathcal{E}$ such that $v_l \in e^{\mathrm{in}}$ and $v_{l+1} \in e^{\mathrm{out}}$. 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*

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

and its non-descendants $\mathrm{NonDes}(v)$ 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 27 (Bayesian Network)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic hypergraph with edges of the form*

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

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

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

which represents the probability distribution

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

By definition each tensor decorating a hyperedge is directed with $X_{\mathrm{Pa}(v)}$ 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. 94 that their contraction defines a probability distribution.

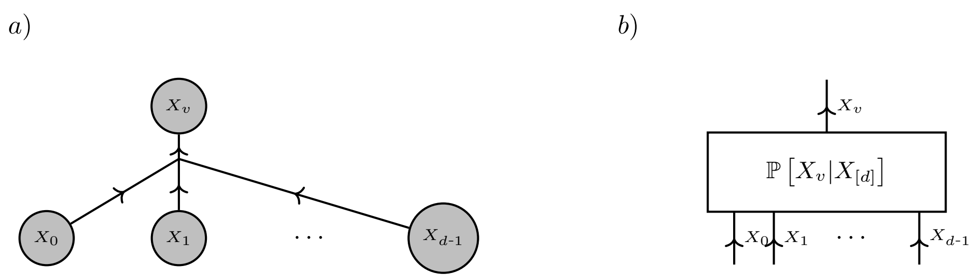


Figure 9: Example of a Factor of a Bayesian Network to the node X_v with parents $X_0, \dots, X_{d-1}$, as an directed edge a) which is decorated by a directed tensor b).

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 11 (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

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

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 The. 8 and get

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

"$\Leftarrow$": To show the converse direction, let there be a Bayesian Network $\mathbb{P}\left[X_{\mathcal{V}}\right]$ 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

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

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. 7, it now follows that X_v is independent of $\mathrm{NonDes}(v)$ conditioned on $\mathrm{Pa}(v)$. □

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

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

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

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

$|\mathcal{V}| - 1 \to |\mathcal{V}|$: 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 121, 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 12 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.

4.6.4 Hidden Markov Models

Hidden Markov Models are examples of Bayesian Networks, constructed as follows. Let us recall Markov Chains as investigated in The. 9 and extend them by observation variables E_k for $k \in [d]$, representing limited observations of the state variables X_k. To be more precise, we assume the following conditional independencies:

- As for Markov Chains, we assume that for $k \in [d]$ with $k \geq 1$ the variable X_k is independent of $X_{[k-1}$ and $E_{[k-1]}$ conditioned on X_{k-1}
- In addition, for we assume that for $k \in [d]$ the observation variable E_k is independent of $X_{[k}$ and $E_{[k]}$ conditioned on X_k

From this conditional independence assumption, we apply the Chain Rule The. 5 given the order of variables

$$X_0, E_0, X_1, E_1, \dots, X_{d-1}, E_{d-1}$$

and get

$$\begin{aligned}
\mathbb{P}\left[X_{[d]}, E_{[d]}\right] = \big\langle & \{\mathbb{P}\left[X_0\right], \mathbb{P}\left[E_0|X_0\right]\} \\
& \cup \{\mathbb{P}\left[X_k|X_{[k]}, E_{[d]}\right] \,:\, k \in [d]\} \\
& \cup \{\mathbb{P}\left[E_k|X_{[k+1]}, E_{[d]}\right] \,:\, k \in [d]\}\big\rangle\left[X_{[d]}, E_{[d]}\right].
\end{aligned}$$

We now apply the conditional independence assumptions to sparsify the appearing conditional distributions by application of The. 8. This results in the decomposition (see Figure 10b)

$$\begin{aligned}
\mathbb{P}\left[X_{[d]}, E_{[d]}\right] = \big\langle & \{\mathbb{P}\left[X_0\right], \mathbb{P}\left[E_0|X_0\right]\} \\
& \cup \{\mathbb{P}\left[X_k|X_{k-1}\right] \,:\, k \in [d]\} \\
& \cup \{\mathbb{P}\left[E_k|X_k\right] \,:\, k \in [d]\}\big\rangle\left[X_{[d]}, E_{[d]}\right].
\end{aligned}$$

In addition to the stochastic transition matrices $\mathbb{P}\left[X_k|X_{k-1}\right]$ appearing in Markov Chains, we further have stochastic observation matrices $\mathbb{P}\left[E_k|X_k\right]$ for $k \in [d]$. Their contraction with marginal distribution of the respective state varibles delivers the marginal distribution of the observation matrix by

$$\mathbb{P}\left[E_k\right] = \langle \mathbb{P}\left[E_k|X_k\right], \mathbb{P}\left[X_k\right]\rangle\left[E_k\right]$$

We notice, that this is a Bayesian Network on a directed acyclic hypergraph $\mathcal{G}$ (see Figure 10a) consistent in nodes $\{X_{[d]}\} \cup \{E_{[d]}\}$ to each state and observation variables, and the directed hyperedges by

- $(\varnothing, \{X_0\})$, decorated by the intial marginal distribution of X_0
- $(\{X_{k-1}\}, \{X_k\})$ for $k \in [d]$ with $k \geq 1$, decorated by stochastic transition matrices
- $(\{X_k\}, \{E_k\})$ for $k \in [d]$, decorated stochastic observation matrices

While we have derived this directed graph structure directly based on the chain rule decomposition with sparsified conditional distributions, it also follows from the more generic hypergraph characterization of Bayesian Networks through separability by The. 11.

4.7 Exponential Families

Exponential families are collections of probability distributions, generalizing the previous discussed graphical models Wainwright and Jordan (2008); Murphy (2022). The probability distributions, which are members of an exponential family, share the computation of the probability tensor based on a boolean base measure, marking the support of the distribution, and a statistic function containing features. They differ only by canonical parameters which weight the features at a given state to calculate the respective probability. Exponential families consist the most generic distributions investigated in this work and will also serve as a generic framework in the discussion of probabilistic reasoning in Chapter 5, as well as for neuro-symbolic models in Part II.

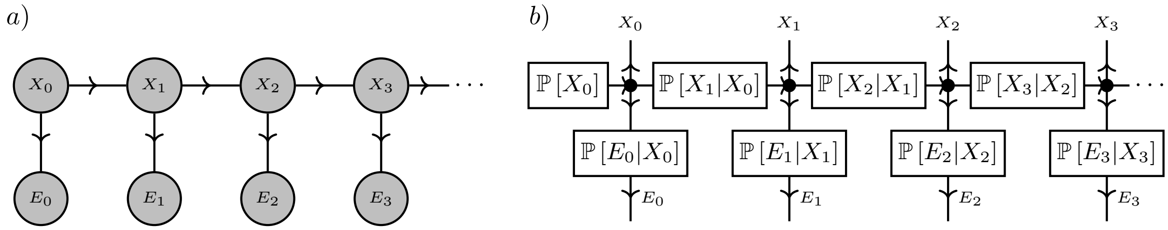


Figure 10: Decomposition of a probability distribution in the Hidden Markov Model, consistent of state variables X_k and observation variables E_k. Given the models conditional independence assumptions, the distribution is a Bayesian Network on the directed hypergraph a). The hypergraph is decorated by the network of conditional probability tensors b), which are interpreted as stochastic transition matrices $\mathbb{P}[X_k|X_{k-1}]$ and stochastic observation matrices $\mathbb{P}[E_k|X_k]$.

Definition 28. *Given a statistic function*

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

and a boolean base measure

$$\nu : \bigtimes_{k\in[d]} [m_k] \to \{0,1\}$$

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*

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

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

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

We used the selection encoding to represent the weighted summation over the statistics, that is the tensor (see Def. 15)

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

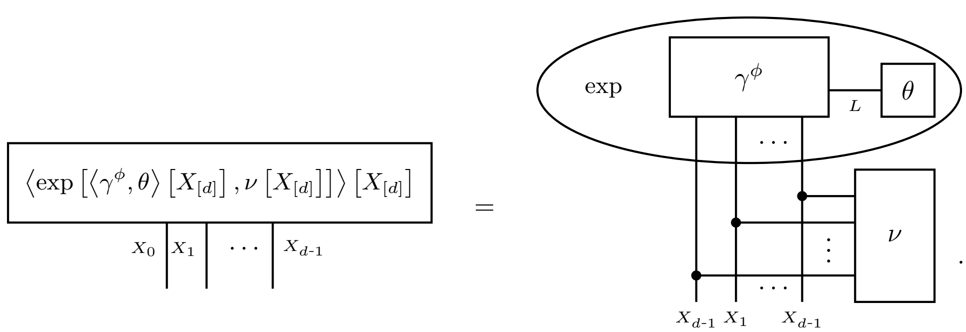
defined for $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ and $l \in [p]$ as

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

The selection encoding represent the weighted sum of the statistic coordinates by the canonical parameter vector $\theta[L]$ as a contraction

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

For more details on this representation scheme, we refer to The. 105 in Chapter 13. Up to normation, we sketch the probability distribution of any member by the tensor network diagram



We here denote by an ellipsis the coordinatewise transformation by the exponential function (see Sect. 13.2). Since such coordinatewise transformation are nonlinear, they are a caveat for efficient contraction of the diagram.

Since we restrict the discussion to finite state spaces, the distribution $\mathbb{P}^{(\phi,\theta,\nu)}$ is well-defined for any $\theta[L] \in \mathbb{R}^p$. 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 Wainwright and Jordan (2008).

As before, we restrict to boolean base measures, which have to satisfy $\langle \nu \rangle[\varnothing] \neq 0$ for respective distributions to exist. 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. 18). In Chapter 9 we will investigate distributions, where the base measures and the sufficient statistics share a common decomposition framework.

Lemma 1. *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

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

□

A further useful criterion is that of minimality of an exponential family, as we define next.

Definition 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 non-vanishing vector $V[L]$ and a scalar $\lambda \in \mathbb{R}$ with*

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

If a statistic is not minimal, we can omit coordinates of it without affecting the expressivity $\Gamma^{\phi,\nu}$. As long as we find a non-vanishing vector $V[L]$ and $\lambda \in \mathbb{R}$ as in Def. 29, we can choose a coordinate ϕ_l such that $V[L=l] \neq 0$, conclude that the coordinate is linear dependent on the others and drop it as redundant.

4.7.1 Tensor Network Representation

As we have observed, the selection encoding formalism can efficiently represent the energy tensor to a member of an exponential family, but through coordinatewise transform by the exponential does not provide an efficient decomposition scheme of the probability distribution itself. We now overcome this problem with usage of the relational encoding formalism to represent members of exponential families by a single contraction without nonlinear transforms.

Theorem 13 (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_l taking values in* $[|\mathrm{im}(\phi_l)|]$ *(see for more details on this scheme Chapter 14), given an interpretation map*

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

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

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

and have

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

Proof. We embed the image of ϕ in the cartesian product of the coordinate images

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

and design enumerate the embedded image of ϕ by the variables $Y_{[p]}$. The. 100, to be shown in Chapter 14, implies

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

Here we denote by $\langle \cdot, \theta\left[L\right]\rangle$ the dual function to $\theta\left[L\right]$, which assigns to vectors their contraction with $\theta\left[L\right]$. Its restriction onto the vectors in $\bigtimes_{l\in[p]} \mathrm{im}\,(\phi_l)$ is the tensor satisfying

$$\exp\left[\langle \cdot, \theta\rangle\right]|_{\mathrm{im}(\phi)}\left[Y_{[p]}\right] = \bigotimes_{l\in[p]} \exp\left[\cdot\theta\left[L=l\right]\right]|_{\mathrm{im}(\phi_l)}\left[Y_l\right] = \bigotimes_{l\in[p]} W^l[Y_l] .$$

We further have (see The. 101 in Chapter 14)

$$\rho^{\phi}\left[Y_{[p]}, X_{[d]}\right] = \left\langle \left\{\rho^{\phi_l}\left[Y_l, X_{[d]}\right] \,:\, l\in[p]\right\}\right\rangle \left[Y_{[p]}, X_{[d]}\right] .$$

Refining the above decomposition of $\exp\left[\left\langle \gamma^{\phi}, \theta \right\rangle \left[X_{[d]}\right]\right]$ by these further decompositions we arrive at the claim. $\square$

In the proof of The. 13 we have observed, that the relational encoding $\rho^{\phi}\left[Y_{[p]}, X_{[d]}\right]$ of the statistics decomposed into a tensor network of relational encodings $\rho^{\phi_l}\left[Y_l, X_{[d]}\right]$ to the coordinate of the statistic. We can exploit further decomposition mechanisms, which will be discussed in full detail in Chapter 14, to find even sparser decompositions. This is for example the case, when the coordinates of the statistic are compositions of functions depending on small numbers of variables. When the coorindates of the statistic furthermore share similar parts in their compositions, these parts can be shared in the decomposition. We will investigate such sparsification mechanisms in more detail in Chapter 9, where the coordinates of the statistic are propositional formulas with a natural decomposition by their syntactical description.

The tensor network representation of an exponential family by The. 13 is a Markov Network consistent of two types of cores. First, we refer to the relational encodings ρ^{ϕ_l} of the coordinates of a statistic as computation cores. Our intuition is that they compute the hidden variable Y_l, based on Basis Calculus (see Chapter 14), which encode the value of the coordinate with respect to the image interpretation map I_l. We notice, that since they are directed with Y_l being the only outgoing variable, they do not influence any contraction with open variables $X_{[d]}$, unless further tensors sharing the variable Y_l are present in the contraction. The influence of the contraction is performed by the activation cores $W^l[Y_l]$, which exploit the computed statistic variable and provide in combination with the relational encoding a factor

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

to the Markov Network reduced to the observed variables $X_{[d]}$. When the canonical parameter is vanishing at a coordinate, that is $\theta\left[L=l\right] = 0$, then this factor is trivial, since $W^l[Y_l] = \mathbb{I}\left[Y_l\right]$ and as a consequence of the directionality of relational encodings we have

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

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

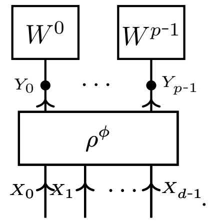
By The. 13 any member of an exponential family is represented by the normed contraction of a collection of unary activation cores contracted with the computation network $\rho^{\phi}\left[Y_{[p]}, X_{[d]}\right]$. We understand these activation cores as a member of a simple Markov Network distributing 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 call this graph the elementary graph, since it also corresponds with elementary tensor network formats consistent of tensor products of vectors. A straightforward generalization of probability distributions representable by exponential families then allows for arbitrary decomposition formats for activation tensors, as we define next.

Definition 30. *Given a statistic $\phi : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}^p$, and a hypergraph $\mathcal{G} = ([p], \mathcal{E})$ with nodes associated to the coordinates of the statistic, we define the by ϕ and $\mathcal{G}$ computable family of distributions by*

$$\Lambda^{\phi,\mathcal{G}} = \left\{ \left\langle \left\{\rho^{\phi}\left[Y_{[p]}, X_{[d]}\right]\right\} \cup \left\{T^e\left[Y_e\right]\right\}\right\rangle \left[X_{[d]}|\varnothing\right] \,:\, T^e\left[Y_e\right] \in \bigotimes_{l\in e}\mathbb{R}^{n_l},\, 0\left[Y_e\right] \prec T^e\left[Y_e\right] \right\} .$$

Note that we restrict to non-negative activation cores by demanding $0\left[Y_e\right] \prec T^e\left[Y_e\right]$, a notation which will be introduced in more detail in Chapter 7 as partial order of tensors. We refer to any member $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\phi,\mathcal{G}}$ as a by ϕ and $\mathcal{G}$ computable distribution.

For unary activation cores, that is for the elementary graph $\mathcal{G}^{\mathrm{EL}}$, any member of $\Lambda^{\phi,\mathcal{G}^{\mathrm{EL}}}$ has up to a normation factor a tensor network decomposition by the diagram

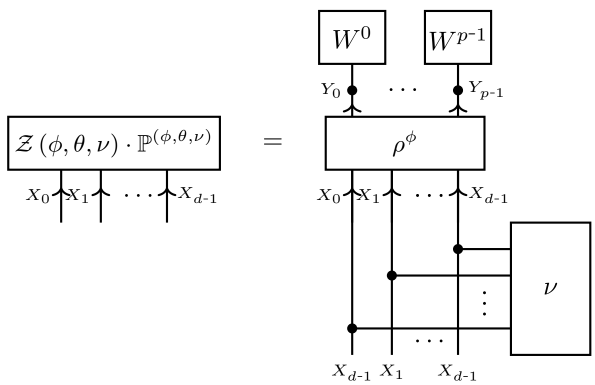


Comparing this representation scheme with The. 13, we conclude as the next corollary, that any member of an exponential family with trivial base measure can be represented by an elementary activation tensors.

Corollary 2 (Corollary of The. 13)**.** *For any statistic* $\phi : \bigtimes_{k\in[d]}[m_k] \to [p]$ *and trivial base measure* $\nu\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$ *we have*

$$\Gamma^{\phi,\mathbb{I}} \subset \Lambda^{\phi,\mathcal{G}^{\mathrm{EL}}} .$$

For elements of the exponential family with general boolean base measure we have with the activation cores constructed in The. 13



where the partition function represents the normalizing contraction of the tensor network. Let us note, that when choosing activation cores with nontrivial support, we can also prepare boolean base measures and in principle extend Cor. 2 to families of nontrivial base measures. We will investigate such schemes later in Chapter 9, where we call them hybrid logic networks.

In comparison with the selection encoding representation of energy tensors, we have prepared a contraction without non-linear transforms, which represents the probability distributions being members of an exponential family. However, relation encoding come with the expense of introducing more auxiliary variables compared with selection encodings. To be more precise, while selection encodings bundle the coordinates of the statistic in single selection variables, relation encodings create for each state $l \in [p]$ of these selection variable an own auxiliary variable Y_l, which enumerated the image of the coordinate and can therefore be of high dimension. Thus, selection encodings offer in general a more efficient storage format coming at the expense of nonlinear operations in the computation of probabilities. We later will encounter situations, where selection encodings are feasible while relation encodings are not, when applying the formalism of formula selecting networks (see Chapter 8) in neuro-symbolic reasoning (see Chapter 10).

Based on Cor. 2 a further natural question is, whether $\Gamma^{\phi,\mathbb{I}}$ is a proper subset of $\Lambda^{\phi,\mathcal{G}^{\mathrm{EL}}}$. This is the case for most statistics ϕ, since members of exponential families are positive with respect to their base measure, which is in the corollaries setting trivial, while in $\Lambda^{\phi,\mathcal{G}^{\mathrm{EL}}}$ we allow also for activation cores with vanishing coordinates, which in general do not produce positive distributions. The only statistics where $\Gamma^{\phi,\mathbb{I}}$ is not a proper subset of $\Lambda^{\phi,\mathcal{G}}$ are along this argumentation constant, since then the activation cores are one-dimensional vectors and vanishing coordinates are prohibited by the need for normalizability. We will follow these intuitions in the discussion of logical reasoning, starting with Chapter 7, and will use the formats $\Lambda^{\phi,\mathcal{G}}$ as hybrid formats storing probability distributions and logical knowledge bases.

While we have restricted our discussion on the elementary decomposition of the activation tensor, further decomposition schemes have interesting interpretations as well. Given a CP decomposition of the activation tensor (see for more details Chapter 15), the corresponding distributions are weighted mixture distributions built from the elementary decompositions. In general, the expressivity increases monotonously with the introduction of additional auxiliary variables and hyperedges in the representation format of activation tensors.

4.7.2 Mean Parameters

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

Definition 31. *Let there be an exponential family defined by a statistic ϕ and a boolean base measure. We call the tensor*

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

the mean parameter tensor to a distribution $\mathbb{P}\left[X_{[d]}\right]$. The set

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

is called the polytope of realizable mean parameters. Here we denote by $\Gamma^{\delta,\nu}$ the set of all probability distributions representable with respect to ν (see Def. 17). 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 inverse, that is a map

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

with $\mathbb{P}^{(\phi, B^{(\phi,\nu)}(F^{(\phi,\nu)}(\theta)),\nu)} = \mathbb{P}^{(\phi,\theta,\nu)}$ for any $\theta[L] \in \mathbb{R}^p$, a backward map of the exponential family.

While introduced here as a property of a distribution, the mean parameters will be central to the discussion of probabilistic inference in Chapter 5.

4.7.3 Markov Networks as Exponential Families

As we have claimed before, exponential families can be regarded as a generalization of graphical models. We here show this claim by a construction of exponential families representing Markov Networks on constant hypergraphs.

Theorem 14 (Exponential Representation of Markov Networks)**.** *Let there be a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a coloring of the nodes by dimensions m_v, we define a sufficient statistics*

$$\phi^{\mathcal{G}} \,:\, \bigtimes_{v \in \mathcal{V}} [m_v] \to \bigtimes_{e \in \mathcal{E}} \left(\bigtimes_{v \in e} [m_v] \right)$$

by a cartesian product $\phi^{\mathcal{G}} = \bigtimes_{e \in \mathcal{E}} \phi_e$ of statistics

$$\phi_e : \bigtimes_{v \in \mathcal{V}} [m_v] \to \bigtimes_{v \in e} [m_v]$$

defined by the restriction of indices to the respective edge, that is for $x_{\mathcal{V}} \in \bigtimes_{v \in \mathcal{V}} [m_v]$

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

Given any Markov Network with positive tensors $\{T^e \,:\, e \in \mathcal{E}\}$ decorating the hyperedges of $\mathcal{G}$ we define

$$\theta[L] = \bigtimes_{e \in \mathcal{E}} \theta_e[X_e]$$

where

$$\theta_e[X_e] = \ln\left[T^e[X_e]\right]$$

and L enumerates a concatenation of the states of X_e. Then, the Markov Network is in the member of the exponential family with trivial base measure, sufficient statistic $\phi^{\mathcal{G}}$ and parameters θ.

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

$$\prod_{e\in\mathcal{E}} T^e [X_e = x_e]$$

$$= \exp\left[\sum_{e\in\mathcal{E}} \ln [T^e [X_e = x_e]]\right]$$

$$= \exp\left[\sum_{e\in\mathcal{E}} \theta_e [X_e = x_e]\right]$$

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

By contraction, we further have

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

we thus get with the above

$$\langle\{T^e : e \in \mathcal{E}\}\rangle [X_{\mathcal{V}}] = \exp [\langle \phi (X_{\mathcal{V}}, L), \theta [L]\rangle [X_{\mathcal{V}}]] . \tag{6}$$

This implies, that the contraction of the tensors in the Markov Network coincides with the exponential of the energy tensor of the constructed member of the exponential family. It follows for the normalization, that

$$\langle\{T^e : e \in \mathcal{E}\}\rangle [X_{\mathcal{V}}|\varnothing] = \langle \exp [\langle\theta, \phi\rangle [X_{\mathcal{V}}]]\rangle [X_{\mathcal{V}}|\varnothing] . \tag{7}$$

We thus conclude, that the Markov Network coincides with the constructed member of the exponential family. □

The mean parameter of the Markov Network exponential family is the cartesian product of the marginals $\mu_e [X_e]$. They are often refered to as beliefs in the literature, as introduced by Pearl (1988). For Markov Networks on tree hypergraphs, and their embedding into junction tree formats, the corresponding mean parameter polytope can be characterized by local consistency constraints. More precisely, it can be shown, that for the statistic constructed in The. 14, in case of tree hypergraphs, the

$$\mathcal{M}_{\phi^{\mathcal{G}}} = \Big\{\mu [L] = \left(\mu_e [X_e]\right)_{e\in\mathcal{E}} : \forall e, \tilde{e} \in \mathcal{E} : \langle \mu_e [X_e]\rangle [X_{e\cap\tilde{e}}] = \langle \mu_e [X_e]\rangle [X_{e\cap\tilde{e}}],$$
$$\forall e : 0 [X_e] \prec \mu_e [X_e] \wedge \langle \mu_e [X_e]\rangle [\varnothing] = 1\Big\}.$$

That is, the polytope of realizable mean parameters consists of those non-negative and normed beliefs, which are coinciding on the contraction of shared variables. Capturing these constraints by Lagrange parameters and performing optimization of certain objectives then results in message-passing schemes, as we will discuss in Chapter 17. If the hypergraph is not minimally connected, this constructed polytope is only an outer bound of the true mean parameter polytope, but still serves as a motivation of loopy belief propagation schemes (see Chapter 4 in Wainwright and Jordan (2008)).

4.7.4 Representation of generic distributions

The formalism of exponential families can capture any probability distribution, when applying statistic functions of large expressivity. Taking for the statistic the identity function $\delta \left[X_{[d]}, L_{[d]}\right]$ defined as

$$\delta \left[X_{[d]} = x_{[d]}, L_{[d]} = l_{[d]}\right] = \begin{cases} 1 & \text{if} \quad x_{[d]} = l_{[d]} \\ 0 & \text{else} \end{cases},$$

we can represent any positive probability distribution $\mathbb{P}\left[X_{[d]}\right]$ as a member of the exponential family $\Gamma^{\delta,\mathbb{I}}$. To see this, it is enough to choose

$$\theta \left[L_{[d]}\right] = \left\langle \ln \left[\mathbb{P}\left[X_{[d]}\right]\right], \delta \left[X_{[d]} = x_{[d]}, L_{[d]} = l_{[d]}\right]\right\rangle \left[L_{[d]}\right],$$

where the contraction with δ copies the variables $X_{[d]}$ to $L_{[d]}$. The energy tensor of this member of $\Gamma^{\delta,\mathbb{I}}$ is then

$$\left\langle \theta \left[L_{[d]}\right], \delta \left[X_{[d]} = x_{[d]}, L_{[d]} = l_{[d]}\right]\right\rangle \left[X_{[d]}\right] = \ln \left[\mathbb{P}\left[X_{[d]}\right]\right]$$

and thus

$$\mathbb{P}^{\delta,\theta,\mathbb{I}}[X_{[d]}] = \left\langle \exp\left[\ln\left[\mathbb{P}\left[X_{[d]}\right]\right]\right] \right\rangle \left[X_{[d]}|\varnothing\right] = \mathbb{P}\left[X_{[d]}\right] \,.$$

We further note, that the mean parameter of this constructed element of $\Gamma^{\delta,\mathbb{I}}$ is

$$\mu\left[L_{[d]}\right] = \left\langle \mathbb{P}^{\delta,\theta,\mathbb{I}}[X_{[d]}], \delta\left[X_{[d]} = x_{[d]}, L_{[d]} = l_{[d]}\right] \right\rangle \left[L_{[d]}\right] = \left\langle \mathbb{P}\left[X_{[d]}\right], \delta\left[X_{[d]} = x_{[d]}, L_{[d]} = l_{[d]}\right] \right\rangle \left[L_{[d]}\right] \,,$$

and thus coincides with the distribution itself, after a relabelling of the distributed variables. Let us notice, that this family also correspond with the Markov Network on the maximal hypergraph $\mathcal{G}^{\max} = (\mathcal{V}, \{\mathcal{V}\})$. We will further revisit this family in Chapter 9, where we will refer to it by the minterm family in order to connect with terminology developed for logical reasoning in Chapter 7.

4.8 Empirical Distributions

To prepare for reasoning on data in the next section, we now derive tensor network representation for empirical distributions, which are defined based on observed states $\left((x_0^j, \ldots, x_{d-1}^j) \,:\, j \in [m]\right)$ of a factored system.

Definition 32. *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

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

where we introduced as single incoming for the relational encoding of the sample selector map the sample selecting variable J taking values in $[m]$.

The relational encoding of the sample selector map has a decomposition by

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

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

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

We can therefore interpret each coordinate of the empirical distribution as the relative frequency of the corresponding state in the observed 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. 15.1.2 as basis CP decompositions. We now exploit this structure to find efficient tensor network decompositions (see Figure 11) based on matrices encoding its variables.

Theorem 15. *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

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

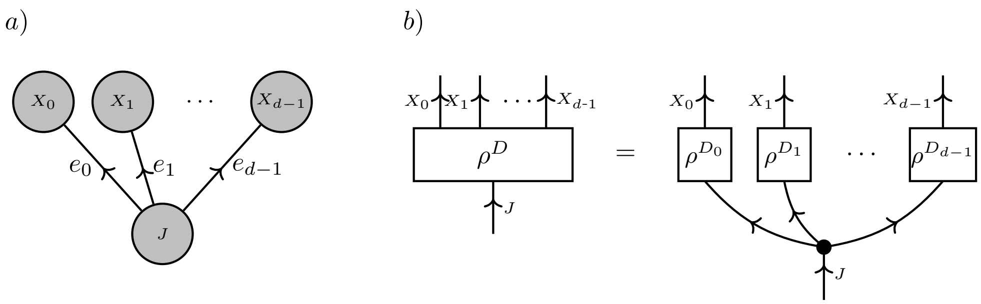


Figure 11: Decomposition of the relational encoding of a sample selector map to a dataset $\left((x_0^j, \ldots, x_{d-1}^j) \,:\, j \in [m]\right)$. a) Interpretation as a sample selection variable J selecting states for the variables $X_{[d]}$ according to the enumerated dataset. b) Corresponding decomposition of the relational encoding ρ^D into a tensor network in the basis CP Format (see Sect. 15.1.2), where $T^{e_k} = \rho^{D_k}$.

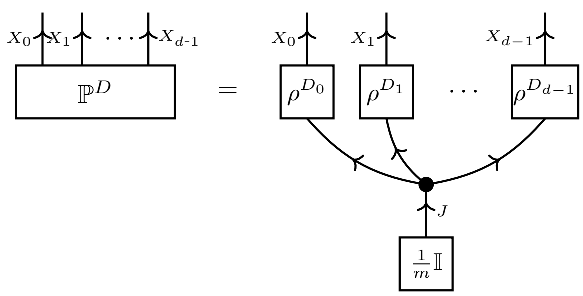
We then have

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

and

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

In a contraction diagram this decomposition is represented as



.

Proof. The first claim is a special case of Theorem 116, to be shown in Chapter 14. To show the second claim we notice

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

With the first claim it now follows that

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

$\square$

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 15 we notice that the scalar $\frac{1}{m}$ could be assigned with any core in a representation of $\mathbb{P}^D$, and the core $\mathbb{I}\left[J\right]$ is thus redundant in the contraction representation. However, creating the core $\frac{1}{m}\mathbb{I}\left[J\right]$ provides us with a simple interpretation of the empirical distribution. We can understand $\frac{1}{m}\mathbb{I}\left[J\right]$ 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 11), the variable J serves as single parent for each categorical variable X_k.

4.9 Discussion and Outlook

This chapter has established a foundational treatment of probability distributions by tensors, and motivated tensor network decompositions along classical approaches towards graphical models. To show this correspondence, we defined both tensor networks and graphical models based on the same hypergraph. This then enabled us to define Markov Networks simply as the normations of tensor networks with non-negative coordinates. In the literature, tensor networks are, however, often treated as being dual to the graphs defining graphical models (see e.g. Robeva and Seigal (2019)). The duality becomes clear, when one interpretes the tensors as nodes and their common variables as edges, as might be natural given the applied notation of wiring diagrams to represent tensor networks. We in this work avoid the discussion of this ambiguity, and treat tensors as decorations of hyperedges.

In the literature, the tensors decorating hyperedges are often refered to as "factors" and their coordinatewise logarithm as "features" Koller and Friedman (2009). With the scope of this work, we avoided such further terminology.

Further, graphical models follow a tradition of definition on graphs, instead of hypergraphs. Tensors, or "factors", are then assigned to maximal cliques. We observed that the notion of maximality is an important assumption, for example in the proof of the Hammersley-Clifford theorem, and therefore introduced the property of clique capturing hypergraphs (see Def. 25) to connect with this graph-based formalism.

While we here restricted our discussion to finite state spaces to each variable, probability distributions can in general be defined for arbitrary measureable spaces. Joint distributions of these more generic variables still have a tensor structure. The discussion of them, however, needs to be more careful, since integrals might diverge and tensors therefore not be normable. By restriction in this work to finite state spaces of factored systems, we where able to exclude such situations.

5 Probabilitic Inference

After having investigated sparse decomposition schemes of probability distributions into tensor networks, we now exploit these schemes to derive efficient reasoning schemes. We first introduce by queries a generic scheme to retrieve information by contractions, and introduce the method of maximum likelihood estimation related to entropy optimization. Then we focus on inference tasks in exponential families, which have been introduced as a generalization of graphical models in the previous chapter.

5.1 Queries

The efficient retrieval of information stored in probability distributions has to exploit the available decomposition schemes. To avoid the instantiation of a distribution based on its decomposition, we directly define deductive reasoning schemes by contractions, which can be executed using the available decomposition.

5.1.1 Querying by functions

We now formalize queries by retrieving expectations of functions given a distribution specified by probability tensors. We exploit basis calculus in defining categorical variables Y_f to tensors f, which are enumerating the set $\mathrm{im}\,(f)$. More details on this scheme are provided in Chapter 14, see Def. 79 therein.

Definition 33. *The marginal query of a probability distribution $\mathbb{P}\left[X_{[d]}\right]$ by a query statistic*

$$f : \bigtimes_{k\in[d]} [m_k] \to \mathcal{U}^f$$

is the vector $\mathbb{P}\left[Y_f\right]$, where Y_f is an image enumerating variable (see Chapter 14 for more details), defined as the contraction

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

If $\mathcal{U}^f \subset \mathbb{R}$, and the statistic f is therefore a tensor, we further define the expectation query of $\mathbb{P}\left[X_{[d]}\right]$ by f as

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

Given another query statistic $g : \bigtimes_{k\in[d]}[m_k] \to \mathcal{U}^g$, which image is enumerated by a variable Y_g, the conditional query of the probability distribution $\mathbb{P}\left[X_{[d]}\right]$ by the statistic f conditioned on g is the matrix $\mathbb{P}\left[Y_f|Y_g\right] \in \mathbb{R}^{|\mathrm{im}(f)|} \otimes \mathbb{R}^{|\mathrm{im}(g)|}$ defined as the normation

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

We notice, that marginal and conditional queries generalize the schemes of marginalization and conditioning on the distributed variables. As such, marginal distributions are marginal queries with respect to a identity query statistic acting on the respective variables. Conditional distributions can be similarly retrieved by two identity statistics, representing the incoming and outgoing variables.

Expectation queries return the expectation of a real-valued feature $f : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}$. When understanding f as a random variable given the probability measure $\mathbb{P}\left[X_{[d]}\right]$, the expectation query returns the expectation of that random variable. Expectation queries are further contractions of marginal queries with the identity function restricted on the image of f, since

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

This contraction equations follows from the more general Cor. 10, which will be shown in Chapter 14.

5.1.2 Mode Queries

A different kind of queries are mode queries, which we formalize by the searches of the indices to maximal coordinates in a tensor.

Definition 34. *Given a tensor $T\left[X_{[d]}\right]$ the mode query is the problem*

$$\mathrm{argmax}_{x_{[d]}} T\left[X_{[d]} = x_{[d]}\right] \,. \tag{8}$$

We can pose mode queries as convex optimization problems. To this end we recall, that the set of all probability distributions is a convex hull of the one-hot encoded states, that is

$$\Gamma^{\delta,\mathbb{I}} = \mathrm{conv}\left(e_{x_{[d]}}\left[X_{[d]}\right] \,:\, x_{[d]} \in \bigtimes_{k\in[d]}[m_k]\right) \,.$$

With this we have

$$\begin{aligned}
\max_{x_{[d]}} T\left[X_{[d]} = x_{[d]}\right] &= \max_{x_{[d]}} \left\langle T\left[X_{[d]}\right], e_{x_{[d]}}\left[X_{[d]}\right]\right\rangle\left[\varnothing\right] \\
&= \max_{\mu\left[X_{[d]}\right]\in\Gamma^{\delta,\mathbb{I}}} \left\langle T\left[X_{[d]}\right], \mu\left[X_{[d]}\right]\right\rangle\left[\varnothing\right] \,.
\end{aligned}$$

We note that the maximation over $\Gamma^{\delta,\mathbb{I}}$ is a convex optimization problem and the maxima are taken at

$$\mathrm{argmax}_{\mu\left[X_{[d]}\right]\in\Gamma^{\delta,\mathbb{I}}} \left\langle T\left[X_{[d]}\right], \mu\left[X_{[d]}\right]\right\rangle\left[\varnothing\right] = \mathrm{conv}\left(e_{x_{[d]}}\left[X_{[d]}\right] \,:\, x_{[d]} \in \mathrm{argmax}_{x_{[d]}} T\left[X_{[d]} = x_{[d]}\right]\right) \,.$$

We will further apply this generic trick to approach mode queries when studying inference problems for exponential families in the following sections.

5.1.3 Energy representations

For exponential families (see Sect. 4.7) we have observed, that often energy tensors have feasible tensor network representations, whereas the corresponding probability distributions can get infeasible. We therefore investigate here schemes to answer queries based on the energy tensor instead of the distribution.

Theorem 16. *Let $E\left[X_{[d]}\right]$ be an energy tensor and $\mathbb{P}\left[X_{[d]}\right] = \left\langle \exp\left[E\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right]$ the corresponding distribution. For 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. To show the theorem, we use a generic simplification property of coordinatewise transforms, which we will show as Lem. 19 in Chapter 13 and get

$$\left\langle \exp\left[E\left[X_A, X_B\right]\right]\right\rangle\left[X_A, X_B = x_B\right] = \left\langle \exp\left[E\left[X_A, X_B = x_B\right]\right]\right\rangle\left[X_A\right]$$

Based on this we get

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

$\square$

Importantly, The. 16 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. In this more generic situation, we would need to sum over exponentiated coordinates, that is

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

Mode queries on probability distributions in an energy representation can always be reduces to mode queries on the energy tensor. This is due to the monotonicity of the exponential function, which implies

$$\begin{aligned}
\mathrm{argmax}_{x_{[d]} \in \times_{k\in[d]}[m_k]} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] &= \mathrm{argmax}_{x_{[d]} \in \times_{k\in[d]}[m_k]} \left\langle \exp\left[E\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]} = x_{[d]} | \varnothing\right] \\
&= \mathrm{argmax}_{x_{[d]} \in \times_{k\in[d]}[m_k]} \exp\left[E\left[X_{[d]} = x_{[d]}\right]\right] \\
&= \mathrm{argmax}_{x_{[d]} \in \times_{k\in[d]}[m_k]} E\left[X_{[d]} = x_{[d]}\right] .
\end{aligned}$$

Since we are only interested in identifying the index of the maximum coordinate, and not its value, we have further dropped the normalization term by partition functions. When one instead need to get the value of the maximal, the partition function cannot be ignored.

5.2 Sampling

Let us here investigate how to draw samples from a prbability distribution, based on queries on it. Naive methods, such as drawing a random number in $[0, 1]$, adding iteratively the coordinates and stopping when the sum exceeds the random variables, are infeasible when having large tensor orders causing exponential increases of the coordinate number. We recall, that the number of coordinates of $\mathbb{P}\left[X_{[d]}\right]$ is $\prod_{k\in[d]} m_k$, which increases exponentially in the number d of the variables. Efficient methods instead have to exploit tensor network decompositions of the decompositions.

5.2.1 Exact Methods

The first insight to derive efficient sampling algorithms is to sample a single variable in each step. Forward sampling (see Algorithm 1) exploits to this end the generic chain decomposition (see The. 5) of a probability distribution, namely

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

It then samples iteratively a state x_k for the variable X_k conditioned on the previously sampled states, that is from the conditonal distribution

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

The generic chain decomposition thereby ensures that probability of getting a state $x_{[d]}$ by this procedure coincides with $\mathbb{P}\left[X_{[d]} = x_{[d]}\right]$.

Algorithm 1 Forward Sampling

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

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

end for

Forward Sampling is especially efficient, when sampling from a Bayesian Network respecting the topological order of its nodes. More technically, when the parents $\mathrm{Pa}(k)$ of a node k are contained in the preceding variables $[k]$, we apply the conditional independence assumption (more precisely The. 8 in combination with The. 11) to get

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

Since this conditional probability coincides with a local tensor in the Bayesian Network, we can avoid to contract the network for preparing the conditional distribution. Different to more general Markov Networks, forward sampling from Bayesian Network can therefore be done efficiently by reduction to conditional queries answerable using local tensors. We note that it is important to sample in the topological order induced by the underlying directed hypergraph, since the computation of generic conditional distributions is also for Bayesian Networks NP-hard (see Chapter 13 in Koller and Friedman (2009)). Sampling along the topological variable order requires only tractable to answer conditional queries on the Bayesian Network.

5.2.2 Gibbs Sampling

While we have seen that forward sampling can be performed efficiently on Bayesian Networks, Gibbs sampling can be also performed efficiently for Markov Networks. Gibbs sampling Algorithm 2 overcomes the intractability problems of sampling steps during forward sampling at the expense of repetitions of the sampling step. When performing finite repetitions, Gibbs sampling in general samples from an approximate distribution to the one desired. It can be shown, that these approximate distribution tend to one desired in the asymptotic limit of infinite repetitions of the sampling step (see Chapter 12 in Koller and Friedman (2009)).

Algorithm 2 Gibbs Sampling

---

for $k \in [d]$ **do**
  Draw x_k from an initialization distribution.
end for
while Stopping criterion is not met **do**
  for $k \in [d]$ **do**
    Draw x_k from the conditional distribution

$$\mathbb{P}\left[X_k | X_{[d]/\{k\}} = x_{[d]/\{k\}}\right]$$

  end for
end while

The central problem of forward sampling on Markov Networks has been the need for global contractions to answer the required conditional queries, which originates from large numbers of variables to be marginalized out. When avoiding the marginalization of variables, and conditioning on them instead, global contractions can be avoided. To be more precise, for any tensor network $\mathcal{T}^{\mathcal{G}}$ on $\mathcal{G} = ([d], \mathcal{E})$ and any $k \in [d]$ we have

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_k, X_{[d]/\{k\}}\right] = \left\langle \left\{ T\left[e\right] X_k, X_{e/\{k\}} = x_{e/\{k\}} \,:\, e \in \mathcal{E},\, k \in e \right\} \right\rangle \left[X_k\right] \cdot \prod_{e \in \mathcal{E},\, k \notin e} T\left[e\right] X_e = x_e \,.$$

As a consequence, we get for the Markov Network $\mathbb{P} = \mathbb{P}^{\mathcal{G}}$ to $\mathcal{T}^{\mathcal{G}}$, that

$$\begin{aligned} \mathbb{P}\left[X_k | X_{[d]/\{k\}} = x_{[d]/\{k\}}\right] &= \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_k, X_{[d]/\{k\}} | \varnothing\right] \\ &= \left\langle \left\{ T\left[e\right] X_k, X_{e/\{k\}} = x_{e/\{k\}} \,:\, e \in \mathcal{E},\, k \in e \right\} \right\rangle \left[X_k | \varnothing\right] \,. \end{aligned}$$

The conditional queries on a Markov Network asked in Gibbs Sampling can therefore be answered by contractions only of those tensors containing the variable X_k. To find further locally answerable conditional queries, we need to condition only on the neighbored variables, refered to as Markov blanket, such that the other variables are conditionally independent. This follows from the characterization of the conditional independences eminent in Markov Networks, which has been shown in The. 10, and can be used to design further tractable sampling schemes for Markov Networks.

We can further answer these conditional queries efficiently, when we perform Gibbs sampling on a probability distribution in an energy representation, that is $\mathbb{P}\left[X_{[d]}\right] = \left\langle \exp\left[E\left[X_{[d]}\right]\right] \right\rangle \left[X_{[d]} | \varnothing\right]$. Using The. 16, we have

$$\mathbb{P}\left[X_k | X_{[d]/\{k\}} = x_{[d]/\{k\}}\right] = \left\langle \exp\left[E\left[X_k, X_{[d]/\{k\}} = x_{[d]/\{k\}}\right]\right] \right\rangle \left[X_A | \varnothing\right]$$

We note, that the main property of the conditional query exploited here, is that all variables but the one sampled appear as a condition and none is marginalized out. In the scheme of forward sampling, where most of the variables are marginalized out in many queries, we cannot apply this trick and would have to perform sums over exponentiated coordinates to the variables marginalized out.

5.2.3 Simulated Annealing

Simulated annealing is an adapted sampling scheme that targets mode queries rather than generating representative samples from a distribution. It employs an annealing process that gradually transforms the probability distribution by increasingly favoring high-likelihood configurations, thereby improving the chances of sampling a solution to a mode query. To be more precise, let there be a distribution in energy representation, that is $\mathbb{P}\left[X_{[d]}\right] = \left\langle \exp\left[E\left[X_{[d]}\right]\right] \right\rangle \left[X_{[d]} | \varnothing\right]$. Whe introduce a parameter $\beta \in \mathbb{R}$, which we understand as the inverse temperature, and anneal the distribution through scaling its energy by this parameter. In the limit of $\beta \to \infty$, for each state $x_{[d]} \in \bigtimes_{k \in [d]} [m_k]$ the annealed distribution behaves as

$$\left\langle \exp\left[\beta \cdot E^{X_{[d]}}\left[\right]\right] \right\rangle \left[X_{[d]} = x_{[d]} | \varnothing\right] \to \left\langle \mathbb{I}_{\mathrm{argmax}_{x_{[d]} \in \bigtimes_{k \in [d]} [m_k]} E\left[X_{[d]} = x_{[d]}\right]} \right\rangle \left[X_{[d]} = x_{[d]} | \varnothing\right] \,.$$

In this limit, the annealed distribution tends to the uniform distribution of the maximal coordinates, that is the uniform distribution of the set

$$\mathrm{argmax}_{x_{[d]} \in \times_{k \in [d]}[m_k]} E\left[X_{[d]} = x_{[d]}\right] = \mathrm{argmax}_{x_{[d]} \in \times_{k \in [d]}[m_k]} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] .$$

To integrate annealing into Gibbs sampling, one chooses a parameter β for each repetition of a sampling step and sample from the conditioned annealed distribution $\left\langle \exp\left[\beta \cdot E\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$. Through increasing β during the algorithm, the samples are drawn towards states with larger coordinates in $\mathbb{P}\left[X_{[d]}\right]$. However, when β is large, the sampling procedure can get stuck in local maxima, whereas small β are in favor of overcoming such. The inverse temperature is thus understood as a tradeoff parameter between the exploration of new regions of the state space and increasing the coordinate of the sample by local coordinate optimization. It is therefore typically chosen low in the beginning of the sampling algorithm and then sequentially increased to find maximal coordinates. Due to this typical increasement of the inverse temperature strategy, the algorithm is refered to as simulated annealing.

5.3 Maximum Likelihood Estimation

So far we have been concerned with deductive reasoning task, that is retrieve information from a given distribution or drawing a sample. We now turn to inductive reasoning tasks, where a probability distribution is estimated given data. To present the generic framework of maximum likehihood estimation in the tensor network contraction formalism, we introduce the likelihood loss exploiting the structure of empirical distribution, and then provide interpretations in terms of entropies.

5.3.1 Likelihood Loss

Following the notation in Sect. 4.8, datasets $\left((x_0^j, \dots, x_{d-1}^j) \, : \, j \in [m]\right)$ are collections of observed samples

$$D(j) = (x_0^j, \dots, x_{d-1}^j) \in \bigtimes_{k \in [d]} [m_k]$$

of a factored system. The likelihood of a probability distribution $\mathbb{P}\left[X_{[d]}\right]$ to produce an observed sample is then

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

We further introduce the likelihood of $\mathbb{P}\left[X_{[d]}\right]$ with respect to a dataset as

$$\mathbb{P}\left[\left((x_0^j, \dots, x_{d-1}^j) \, : \, j \in [m]\right)\right] := \prod_{j \in [m]} \mathbb{P}\left[X_{[d]} = D(j)\right] .$$

The likelihood draws on the assumption, that each datapoint in the dataset has been drawn independently from the same distribution. When this generating distribution coincides with $\mathbb{P}\left[X_{[d]}\right]$, then the probability of generating a dataset by this scheme is the likelihood. In inductive reasoning, the true distribution $\mathbb{P}\left[X_{[d]}\right]$ is unknown and needs to be approximatively estimated based on data and a learning hypothesis. We will therefore compute and compare the likelihood of distributions, which in general do not coincide with the distribution generating the data. It is thus important to not understand the likelihood as a probability, which is only true for the generating distribution, as pointed out in Chapter 2 in MacKay (2003).

Let us now transform the likelihood to find an representation involving the empirical distribution (see Def. 32), for which efficient tensor network decompositions have been derived in The. 15. Applying a scaled logarithm we get

$$\begin{aligned}
\frac{1}{m} \cdot \ln\left[\mathbb{P}\left[\left((x_0^j, \dots, x_{d-1}^j) \, : \, j \in [m]\right)\right]\right] &= \frac{1}{m} \cdot \ln\left[\prod_{j \in [m]} \mathbb{P}\left[X_{[d]} = D(j)\right]\right] \\
&= \frac{1}{m} \sum_{j \in [m]} \ln\left[\mathbb{P}\left[X_{[d]} = D(j)\right]\right] \\
&= \left\langle \ln\left[\mathbb{P}\left[X_{[d]}\right]\right], \mathbb{P}^D\left[X_{[d]}\right]\right\rangle [\,] .
\end{aligned}$$

Let us notice, that this transform of the likelihood is monotoneous and therefore does not influence the position of the maximum, when optimizing the likelihood. Motivated by this property, we use the transformed form to define the loss-likelihood loss and maximum likelihood estimation.

Definition 35. *The log-likelihood loss of a distribution $\mathbb{P}$ given a dataset $\left((x_0^j, \ldots, x_{d-1}^j) \,:\, j \in [m]\right)$ is the functional*

$$\mathcal{L}_D\left(\mathbb{P}\right) = -\left\langle \ln\left[\mathbb{P}\left[X_{[d]}\right]\right], \mathbb{P}^D\left[X_{[d]}\right]\right\rangle\left[\right].$$

Having a hypothesis $\Gamma \subset \Gamma^{\delta,\mathbb{I}}$, that is a set of probability distributions, the maximum likelihood estimation is the problem

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

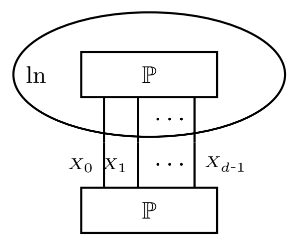
5.3.2 Entropic Interpretation

The Shannon entropy, which has been introduced in the seminal paper Shannon (1948), is a foundational concept in various research fields beyond statistical learning, such as information theory or statistical physics. While a detailed discussion is out of the scope of this work, we here only provide computation schemes of the entropy based on contractions of distributions.

Definition 36 (Shannon entropy)**.** *The Shannon entropy of a distribution $\mathbb{P}\left[X_{[d]}\right]$ is the quantity*

$$\mathbb{H}\left[\mathbb{P}\right] = \sum_{x_{[d]}\in \times_{k\in[d]}[m_k]} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] \cdot \left(-\ln\left[\mathbb{P}\left[X_{[d]} = x_{[d]}\right]\right]\right) = \left\langle \mathbb{P}, -\ln\left[\mathbb{P}\right]\right\rangle\left[\varnothing\right] .$$

We represent the Shannon entropy by the tensor network diagram

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

where we denote a coordinatewise transform by the logarithm as an ellipsis (see Sect. 13.2).

We here make the convention $\ln[0] = -\infty$ and $0\cdot\ln[0] = 0$, to have the Shannon entropy well-defined for distributions with non-trivial support.

Among the distributions in the same tensor space, the uniform distribution maximizes the Shannon entropy

$$\mathbb{H}\left[\langle\mathbb{I}\rangle\left[X_{[d]}|\varnothing\right]\right] = \sum_{k\in[d]} \ln\left[m_k\right]$$

and the one-hot encodings to states $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ minimize the Shannon entropy

$$\mathbb{H}\left[e_{x_{[d]}}\left[X_{[d]}\right]\right] = 0 .$$

The Shannon entropy measures the information content of a distribution and is therefore a central tool for regularization in statistical learning (see for an introduction Chapter 2 in MacKay (2003)). We therefore exploit this information content as a regularizer to identify a distribution among those coinciding in the answer to a collection of expectation queries. To be more precise, let there be for $l \in [p]$ query tensors f^l (see Def. 33) and $\mathbb{P}^D$ an empirical distribution. The problem of maximal entropy with respect to coinciding expectation queries with $\mathbb{P}^D$ is then posed as

$$\mathrm{argmax}_{\mathbb{P}\in\Gamma^{\delta,\mathbb{I}}}\mathbb{H}\left[\mathbb{P}\right] \quad \text{subject to} \quad \forall l \in [p] : \left\langle \mathbb{P}, f^l\right\rangle\left[\varnothing\right] = \left\langle \mathbb{P}^D, f^l\right\rangle\left[\varnothing\right]$$

where $\Gamma^{\delta,\mathbb{I}}$ is the set of probability distributions given a factored system. We study instances of this maximal entropy problem later in Sect. 5.6.3, where we show that its solution is a member of the exponential family, which statistic is build by the query tensors. We will further provide connections between the problems of maximal entropy and maximum likelihood estimation.

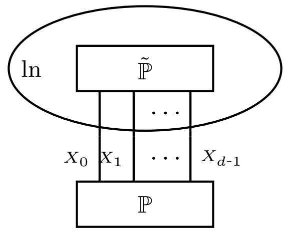
While the Shannon entropy is a property of a single distribution, the cross-entropy is a straight forward generalization towards pairs of distributions. We first introduce this quantity and then interpret the log-likelihood loss based on the cross-entropy.

Definition 37 (Cross entropy and Kullback Leibler divergence)**.** *The cross-entropy between two distributions $\mathbb{P}\left[X_{[d]}\right]$ and $\tilde{\mathbb{P}}[X_{[d]}]$ defined with respect to the same factored system is the quantity*

$$\mathbb{H}\left[\mathbb{P},\tilde{\mathbb{P}}\right] = \sum_{x_0,\ldots,x_{d-1}} \mathbb{P}\left[X_0=x_0,\ldots,X_{d-1}=x_{d-1}\right]\cdot\left(-\ln\left[\tilde{\mathbb{P}}[X_{[d]}=x_{[d]}]\right]\right) = \left\langle \mathbb{P},-\ln\left[\tilde{\mathbb{P}}\right]\right\rangle[\varnothing]\,.$$

The cross-entropy is captured by the tensor network diagram

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



The Kullback-Leiber divergence between $\mathbb{P}\left[X_{[d]}\right]$ and $\tilde{\mathbb{P}}[X_{[d]}]$ is the quantity

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

Let us notice, that we have $\mathbb{H}\left[\mathbb{P},\tilde{\mathbb{P}}\right]=\infty$ if and only if there is a state $x_{[d]}\in\bigtimes_{k\in[d]}[m_k]$ such that $\mathbb{P}\left[X_{[d]}=x_{[d]}\right]>0$ and $\tilde{\mathbb{P}}[X_{[d]}=x_{[d]}]=0$.

The Gibbs inequality (for a proof see for example Chapter 2 in **?**) states that for any distributions $\mathbb{P}\left[X_{[d]}\right]$ and $\tilde{\mathbb{P}}[X_{[d]}]$ we have

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

where equality holds if any only if $\mathbb{P}=\tilde{\mathbb{P}}$ This ensures, that the Kullback-Leiber Divergence between any distributions is positive vanishes only if both distributions coincide.

We in the next lemma provide an entropic interpretation of maximum likehlihood estimation as defined in Def. 35.

Lemma 2. *The maximum likelihood estimation Problem $\mathrm{P}^{\mathcal{L}_D}_{\Gamma,\mathbb{P}^D}$ is equivalent to the minimization of cross-entropy and Kullback-Leibler divergence, that is*

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

Proof. Comparing the log-likelihood loss in Def. 35 with the cross-entropy in Def. 37, we get

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

which established the equivalence of maximum likelihood estimation and cross-entropy minimization. Futher, since

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

and $\mathbb{H}\left[\mathbb{P}^D\right]$ is a constant offset in the objective, maximum likelihood estimation is equivalent to the minimization of the Kullback-Leibler divergence. □

Example 4 (Cross entropy with respect to exponential families)**.** *If $\tilde{\mathbb{P}}$ is a member of an exponential family, we have*

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

The last term vanishes, given the convention $0\cdot\ln[0]=0$, if and only if for any $x_{[d]}$ with $\nu\left[X_{[d]}=x_{[d]}\right]=0$ we have $\mathbb{P}\left[X_{[d]}=x_{[d]}\right]=0$, and is infinite instead. Therefore, the cross entropy between a distribution and a member of an exponential family is finite, if and only if the distribution is representable with respect to the base measure ν (see Def. 17). If $\mathbb{P}$ is representable with respect to ν, we can abbreviate the cross-entropy to

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

5.4 Polytope of mean parameters

To prepare for the discussion of forward and backward inference in exponential families, we in this section investigate the polytope of mean parameters. Given a pair of a statistic $\phi : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}^p$ and a boolean base measure $\nu\left[X_{[d]}\right]$, the polytope of mean parameters is the set (see Def. 31)

$$\mathcal{M}_{\phi,\nu} = \mathrm{conv}\left(\left\langle \gamma^{\phi}\left[X_{[d]}, L\right], \mathbb{P}\left[X_{[d]}\right]\right\rangle[] : \mathbb{P}\left[X_{[d]}\right] \in \Gamma^{\delta,\nu}\right) ,$$

where by $\Gamma^{\delta,\nu}$ we note the set of by ν representable distributions (see Def. **??**). We in this section, we will further characterize the polytope of realizable mean parameters.

The mean parameters are computated as collections of expectation queries to ϕ_l, which are answered against distributions in $\Gamma^{\delta,\nu}$. For any $l \in [p]$ we have for the mean parameter $\mu[L]$ reproduced by a distribution $\mathbb{P}\left[X_{[d]}\right]$

$$\mu[L=l] = \mathbb{E}[\phi_l] = \left\langle \gamma^{\phi}\left[X_{[d]}, L=l\right], \mathbb{P}\left[X_{[d]}\right]\right\rangle[\varnothing] .$$

5.4.1 Representation by convex hulls

First of all, we provide a simple characterization of the sets of mean parameters as the convex hull of the slices to the selection encoding of the statistic. Such sets are refered to as $\mathcal{V}$-polytopes Ziegler (2013).

Theorem 17. *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*

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

Proof. First we realize that the characterization of by ν representable distributions is a standard simplex extended by trivial coordinates, that is

$$\Gamma^{\delta,\nu} = \mathrm{conv}\left(e_{x_{[d]}}\left[X_{[d]}\right] \,:\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right) .$$

This follows from the fact, that the support of any by ν representable distribution is contained in the support of ν. Further, each representable distribution is contained in the convex hull of the one-hot encoded support elements, since any distribution is normed.

The polytope of mean parameters is a linear transform of the elements in $\Gamma^{\delta,\nu}$, since the contraction with γ^{ϕ} is linear. It follows that

$$\begin{aligned}\mathcal{M}_{\phi,\nu} &= \mathrm{conv}\left(\left\langle \gamma^{\phi}\left[X_{[d]}, L\right], e_{x_{[d]}}[L]\right\rangle[] : \nu\left[X_{[d]} = x_{[d]}\right] = 1\right)\\ &= \mathrm{conv}\left(\gamma^{\phi}\left[X_{[d]} = x_{[d]}, L\right] \,:\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right) .\end{aligned}$$

□

5.4.2 Representation as intersecting half-spaces

For any vector $a[L] \in \mathbb{R}^p$ and a scalar $b \in \mathbb{R}$, we call the set

$$\{\mu[L] \,:\, \langle \mu[L], a[L]\rangle[\varnothing] \leq b\} \subset \mathbb{R}^p$$

a half-space of $\mathbb{R}^p$. Bounded intersections of finitely many half-spaces are called $\mathcal{H}$-polytopes Ziegler (2013).

We state next, that the polytope $\mathcal{M}_{\phi,\nu}$ of mean parameters is a $\mathcal{H}$-polytopes.

Theorem 18. *The set $\mathcal{M}_{\phi,\nu}$ is for any statistic ϕ and base measure ν a $\mathcal{H}$-polytope, that is 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. By The. 17, the set $\mathcal{M}_{\phi,\nu}$ is the convex hull of a finite set of vectors and is therefore a $\mathcal{V}$-polytope. We therefore apply the main theorem for polytopes **?**, which states the equivalence of $\mathcal{V}$-polytopes and $\mathcal{H}$-polytope, for which a proof can be found as Theorem 1.1 in Ziegler (2013). Therefore, $\mathcal{M}_{\phi,\nu}$ is also a $\mathcal{H}$-polytope and has is thus the intersection of finitely many half-spaces. □

The determination of the half-space parametrizing $((a_i[L], b_i) \,:\, i \in [n])$ is, however, in general difficult and the main reason for the intractability of probabilistic inference (see e.g. Wainwright and Jordan (2008)).

5.4.3 Characterization of the interior

Theorem 19. *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*

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

Proof. "$\Rightarrow$": By The. 26 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.

"$\Leftarrow$": Since by assumption the statistics is minimal, the convex set $\mathcal{M}_{\phi,\nu}$ is full dimensional (see e.g. Appendix B in Wainwright and Jordan (2008)). We thus use a well-known property for full-dimensional convex sets (see Rockafellar (1997); Hiriart-Urruty and Lemarechal (1993)), 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

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

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

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

□

5.4.4 Characterization of the boundary by faces

Let us now continue with the investigation of the faces of the mean parameter polytope, which we define analogously to Def. 2.1 in Ziegler (2013).

Definition 38. *Given a mean parameter polytope $\mathcal{M}_{\phi,\nu}$ in the half space representation of The. 18, and any subset $\mathcal{I} \subset [n]$ we say that the set*

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

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

While all inequalities in a half-space representation are satisfied for any element of the polytope, we define faces by the additional sharp satisfaction of a subset of the half-space inequalities.

We notice, that the faces build the boundary of $\mathcal{M}_{\phi,\nu}$. This can be easily verified, since for any vector $\mu[L] \in \mathcal{M}_{\phi,\nu}$, for which no halfspace inequalities hold sharply, also a neighborhood satisfies the halfspace inequalities. If any halfspace inequality holds sharply, in the other case, the vector is a member of the corresponding face.

If ϕ is not minimal with respect to ν, we find a non-vanishing vector $V[L]$ and a scalar $\lambda \in \mathbb{R}$ such that

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

This implies, that any probability distribution $\mathbb{P}\left[X_{[d]}\right]$ representable with ν satisfies

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

Any $\mu[L] \in \mathcal{M}_{\phi,\nu}$ then satisfies

$$\langle \mu[L], V[L]\rangle [\varnothing] = \lambda .$$

Thus, the polytope $\mathcal{M}_{\phi,\nu}$ is contained in an affine linear subspace and has vanishing interior. We can further understand this equation as two half-space inequalities

$$\langle \mu[L], V[L]\rangle [\varnothing] \leq \lambda \quad \text{and} \quad \langle \mu[L], V[L]\rangle [\varnothing] \geq \lambda ,$$

which can be integrated into any half-space representation. We conclude, that in the case of non-minimal statistics, the whole polytope $\mathcal{M}_{\phi,\nu}$ is a face itself, since it satisfies these half-space inequalities sharply.

Let us now investigate, that subsets of $\mathcal{M}_{\phi,\nu}$ are the solutions of linear optimization problems constrained on $\mathcal{M}_{\phi,\nu}$. Since such solution sets are intersections of the boundary of $\mathcal{M}_{\phi,\nu}$ with half-spaces, they are faces. In the next theorem we show, how linear optimization problems are constructed to match a given face.

Theorem 20. *For any non-empty face $Q^{\mathcal{I}}_{\phi,\nu}$ to a subset $\mathcal{I} \subset [n]$ there is a vector $\theta[L]$, which we call a normal of the face, such that*

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

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

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

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. Since the face is non-empty, there is a $\mu[L]$ with

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

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

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

it follows that

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

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

As we show next, also a converse statement holds, namely that for any vector $\theta[L]$ we find a face $Q^{\mathcal{I}}_{\phi,\nu}$, such that the $\theta[L]$ is a face normal to that face.

Theorem 21. *For any $\theta[L]$ we find a subset $\mathcal{I} \subset [n]$, such that*

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

Proof. We first notice, that

$$\mathrm{argmax}_{\mu\in\mathcal{M}_{\phi,\nu}} \langle\theta[L],\mu[L]\rangle[\varnothing] = \mathrm{conv}\left(\gamma^{\phi}\left[X_{[d]}=x_{[d]},L\right] \,:\, x_{[d]}\in\mathrm{argmax}_{x_{[d]}\in\times_{k\in[d]}[m_k]}\left\langle\theta[L],\phi\left(x_{[d]}\right)\right\rangle[\varnothing]\right) .$$

Further, since the contraction with $\theta[L]$ is linear, the set $\mathrm{argmax}_{\mu\in\mathcal{M}_{\phi,\nu}}\langle\theta[L],\mu[L]\rangle[\varnothing]$ is contained in the boundary of the polytope $\mathcal{M}_{\phi,\nu}$. We can conlcude, that the set is a face, that is we find a subset $\mathcal{I}\subset[n]$ with

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

□

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

5.4.5 Base measure refinement

For mean parameters $\mu[L]$ outside the interior of $\mathcal{M}_{\phi,\nu}$ we know by The. 19, 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 22. *Let there be a statistic ϕ, which is minimal with respect to a base measure ν, and $\mu[L]\notin\mathcal{M}_{\phi,\nu}{}^{\circ}$. Then there is a vector $\theta[L]\in\mathbb{R}^p$ with*

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

and all distributions reproducing the 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

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

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. 20, in which also a construction procedure is provided given a half-space representation (see The. 18). Now, we have

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

and thus

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

Thus, any distribution reproducing meanparam is a convex combination of the one-hot encodings of the states in $\mathrm{argmax}_{x_{[d]}}\left\langle\theta[L],\gamma^{\phi}X_{[d]}=x_{[d]},L\right\rangle[\varnothing]$, 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 39. *The base measure to the face of $\mathcal{M}$ with normal θ is*

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

The. 22 implies that any mean parameter on a face of $\mathcal{M}_{\phi,\nu}$ can be reproduced with a distribution representable 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 by Algorithm 3 a procedure to refine the base measure until the reduced mean parameter is in the interior of a reduced mean parameter polytope.

Theorem 23. *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 $\tilde{\mu}\in\left(\mathcal{M}_{\tilde{\phi},\tilde{\nu}}\right)^{\circ}$.*

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 $\mu \notin (\mathcal{M}_{\phi,\nu})^{\circ}$ **do**

  while ϕ not minimal with respect to ν (see Def. 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

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. 22 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 5 (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 24. *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 $\tilde{\nu}, \tilde{\phi}$ of the Base Measure Refinement Algorithm 3 we have

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

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. 39 depends only on ϕ. Providing a more technical argument, we have

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

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

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

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

5.5 Forward Mapping in Exponential Families

Forward mappings are expectation queries of the coordinates ϕ_l asked against members of the exponential family itself, which are represented by canonical parameters $\theta[L] \in \mathbb{R}^p$. Each expectation query thereby computes a coordinate of the corresponding mean parameter. For any $\theta[L]$ we have a closed form representation of this expectation query 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 Wainwright and Jordan (2008), we can characterize the forward mapping to exponential families as a variational problem and provide an alternative characterization to this contraction.

5.5.1 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 25. *We have*

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

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

Proof. For a proof see Theorem 3.4 in Wainwright and Jordan (2008). □

We already know by The. 19, that distribution representable by ν reproduce the mean parameters in the interior of $\mathcal{M}_{\phi,\nu}$. We now state that the elements of the corresponding exponential family $\Gamma^{\phi,\nu}$, which are by construction representable by ν, are expressive enough to reproduce the whole interior of $\mathcal{M}_{\phi,\nu}$.

Theorem 26. *For any statistics ϕ the image $\mathrm{im}\left(F^{(\phi,\nu)}\right)$ of the forward map is the interior of the convex polytope $\mathcal{M}_{\phi,\nu}$. If the statistic ϕ is in addition minimal with respect to ν (see Def. 29), the forward map is a bijection between $\mathbb{R}^p$ as the set of canonical mean parameters and $\mathcal{M}_{\phi,\nu}$.*

Proof. For a proof see Theorem 3.3 in Wainwright and Jordan (2008). □

5.5.2 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

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

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

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

thus the entropy term is neglectable for large β. A more precise argument is using a limit of the maxima and can be found in Theorem 8.1 in Wainwright and Jordan (2008)

5.5.3 Mean Field Method

We rewrite

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

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

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

Theorem 27 (Update equations for the mean field approximation)**.** *Keeping all legs but one constant, the problem*

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

is solved at

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

Proof. We have

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

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

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

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.

5.5.4 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

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

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

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

Theorem 28 (Update equations for the structured variational approximation)**.** *The Markov Network $\mathcal{T}^{\mathcal{G}}$ with hypercores $\{T^e[X_e] : e \in \mathcal{E}\}$ 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 $O(\mathcal{T}^{\mathcal{G}}) = \langle E, \langle \mathcal{T}^{\mathcal{G}} \rangle [X_{\mathcal{V}} | \varnothing] \rangle [\varnothing] + \mathbb{H}[\langle \mathcal{T}^{\mathcal{G}} \rangle [X_{\mathcal{V}} | \varnothing]]$. To proof the theorem, we use Lem. 24, which shows a characterization of the derivative of functions

We have

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

Further we have

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

We define the tensor

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

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

The objective has then a representation as

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

Let us now differentiate all terms. With Lem. 24 we now get

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

Further we have

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

and (see Proof of 23)

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

Together, the first order condition

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

is equal to all y_e satisfying

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

We notice, that by normation

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

and that the scalar

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

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

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

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

$$\lambda = \exp[\lambda_1]\,.$$

□

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

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

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

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

5.6 Backward Mapping in Exponential Families

Lemma 2 further implies, that the Maximum Likelihood Problem is the M-Projection (see Chapter 8 in Koller and Friedman (2009)) of a distribution $\mathbb{P}^*$ onto a set Γ of probability tensors is

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

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

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.

5.6.1 Variational Formulation

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

Theorem 30. *Let there be a sufficient statistic ϕ. The map $B^{(\phi,\nu)}: \mathbb{R}^p \rightarrow \mathbb{R}^p$ defined as*

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

is a backward mapping.

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

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

and thus

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

The first order condition is therefore

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

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

5.6.2 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 31. *Let there be any exponential family, a mean parameter vector $\mu^* \in \mathrm{im}\left(F^{(\phi,\nu)}\right)$ 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*

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

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. 30, 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 4

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

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 Lem. 2 the maximum likelihood estimator. □

5.6.3 Connection with Maximum Entropy

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

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

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

Theorem 32. *Let ϕ be a statistic and ν a base measure. For any $\mu^* \in (\mathcal{M}_{\phi,\nu})^\circ$ 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 $\mu^* \in (\mathcal{M}_{\phi,\nu})^\circ$, The. 19 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 Koller and Friedman (2009). 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. 32 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 $\mu^* \notin (\mathcal{M}_{\phi,\nu})^\circ$, the mean paramater is by The. 19 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 33. *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. 23 and the above Lemma. □

The. 33 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. 23.

5.6.4 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. We 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 3. *For any sufficient statistic ϕ a parameter vector θ and a $l \in [p]$ we define*

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

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

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

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

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

□

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. 10.3.

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

5.7 Discussion

Further in Wainwright and Jordan (2008): Convex Duality. Forward mapping coincides with gradient, i.e. $\mu = \nabla A^{(\phi,\nu)}(\theta)$.

In Wainwright and Jordan (2008): 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. $\theta = \nabla\left(A^{(\phi,\nu)}\right)^*(\mu)$.

6 Propositional Logic

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.

6.1 Encoding of Booleans

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

6.1.1 Representation by coordinates

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

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

defined as

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

In Def. 75 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 4. *I is a homomorphism between the groups*

$$\left(\{\mathrm{False}, \mathrm{True}\}, \wedge\right) \quad \textit{and} \quad \left(\{0, 1\}, \cdot\right) .$$

Proof. It suffices to notice, that for arbitrary $z_0, z_1 \in \{\mathrm{False}, \mathrm{True}\}$ 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 Foundation (2025)).

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

$$I(\neg z) = 1 - I(z) \,. \tag{9}$$

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. 14.6. 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 (9) 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

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

defined as

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

results is a homomorphism between the groups

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

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 $\{0, 1\}$.

6.1.2 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

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

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

6.1.3 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 13 and Chapter 14. 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. 97.

6.2 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.

6.2.1 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 40. *A propositional formula $f\left[X_{[d]}\right]$ depending on d atoms X_k is a boolean-valued tensor*

$$f\left[X_{[d]}\right] : \bigtimes_{k\in[d]} [2] \to \{0,1\} \subset \mathbb{R}\,.$$

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

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

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

The propositional formulas coincide therefore with the boolean tensors (see Def. 11).

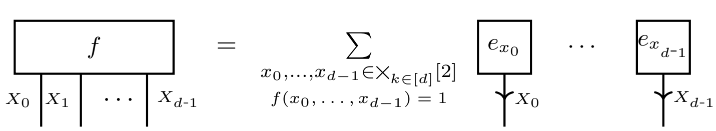
Since propositional formulas are binary valued tensors, the generic decomposition of Lem. 18 simplifies to

$$f\left[X_{[d]}\right] = \sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} f\left[X_0 = x_0,\ldots,X_{d-1} = x_{d-1}\right] \cdot e_{x_{[d]}}\left[X_{[d]}\right] \tag{10}$$

$$= \sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2]\,:\, f[X_0=x_0,\ldots,X_{d-1}=x_{d-1}]=1} e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]\,. \tag{11}$$

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 14 (see Def. 75).

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.

6.2.2 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. 40). 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. 14) of that map to represent the formula (see Figure 12).

Given a factored system with d atoms $X_{[d]}$ and a propositional formula f, the relational encoding of f (see Def. 14) 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

$$\rho^f\left[Y_f, X_{[d]}\right] = \sum_{x_{[d]}\in \times_{k\in[d]}[2]} e_{x_{[d]}}\left[X_{[d]}\right] \otimes e_{f\left[X_{[d]}=x_{[d]}\right]}\left[Y_f\right] . \tag{12}$$

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 5. *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 12)

$$\rho^f\left[Y_f, X_{[d]}\right] = e_0\left[Y_f\right] \otimes \left(\sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=0} e_{x_{[d]}}\left[X_{[d]}\right]\right) \tag{13}$$

$$+ e_1\left[Y_f\right] \otimes \left(\sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=1} e_{x_{[d]}}\left[X_{[d]}\right]\right) \tag{14}$$

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 10, 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 encoding 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. 98).

6.3 Syntax of Propositional Formulas

Relational encodings of propositional formulas are especially useful when representing function compositions by the representation of their components (see The. 99). 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.

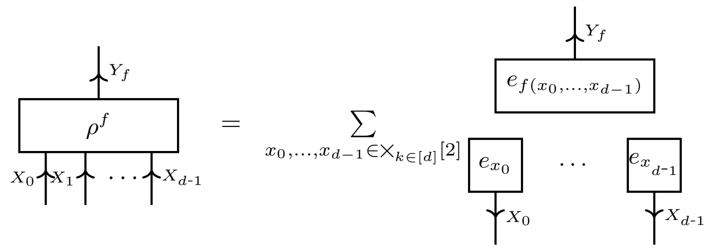


Figure 12: Relational encoding of a propositional formula. The encoding is a sum of the one hot encodings of all states of the factored system in a tensor product with basis vectors, which encode whether the state is a model of the formula. The tensor is directed, since any contraction with an encoded state results in the basis vector evaluating the formula, which we called basis calculus.

6.3.1 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 41. *The tensors $f_k\left[X_{[d]}\right]$ defined for $x_{[d]} \in \times_{k\in[d]}[2]$ 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.

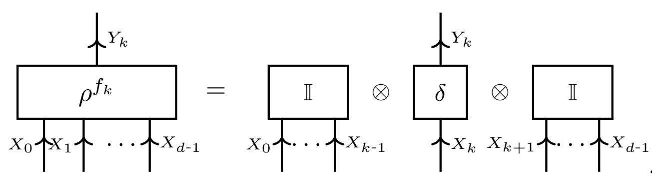
Theorem 34. *Any atomic formula $f_k\left[X_{[d]}\right]$ is represented as*

$$f_k\left[X_{[d]} = x_{[d]}\right] = \langle e_1\left[X_k\right]\rangle\left[X_{[d]}\right] = e_1\left[X_k\right] \otimes \mathbb{I}\left[X_{[d]/\{k\}}\right] \, .$$

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

$$\begin{aligned}
\rho^{X_k}\left[Y_k, X_{[d]}\right] &= \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right] \otimes e_{f_k[X_0=x_0,\ldots,X_{d-1}=x_{d-1}]}\left[Y_k\right] \\
&= \left(e_{0,0}\left[X_k, Y_k\right] + e_{1,1}\left[X_k, Y_k\right]\right) \otimes \mathbb{I}\left[X_l \,:\, l \neq k\right] \\
&= \langle \delta\left[X_k, Y_k\right]\rangle\left[X_{[d]}, Y_k\right] \, .
\end{aligned}$$

□

6.3.2 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 6 unary ($d = 1$) and binary ($d = 2$) connectives.

Example 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* $\Leftrightarrow : [2] \times [2] \to [2]$

$$\Leftrightarrow [Y_f, Y_h] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

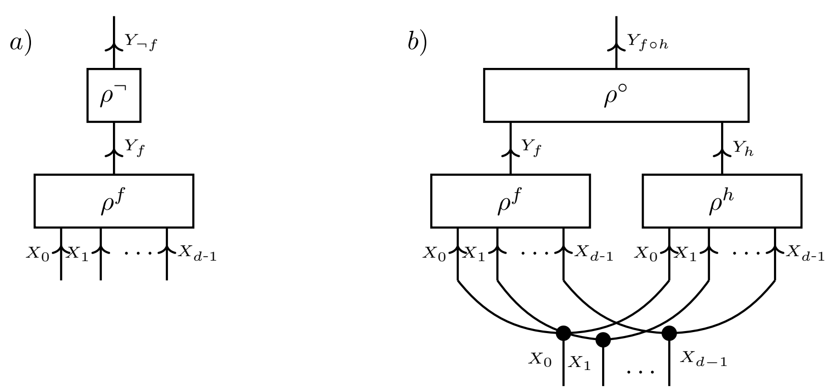


Figure 13: a) Relational encoding of a negated formula f as a tensor network of the encoded formula and the encoded connective $\neg$. b) Relational encoding of a composition of formulas f, h by a connective $\circ \in \{\wedge, \vee, \oplus, \Rightarrow, \Leftrightarrow\}$. The encoding is a contraction of encodings to f, h and $\circ$.

We now show how formulas consisting of connectives acting on other formulas can be represented by basis calculus. 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 13)

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

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 35 (Composition of Formulas)**.** *Let there be a formula $f\left[X_{[d]}\right]$, which has a syntactical decomposition into connectives $\{\circ_l\left[Y_{\mathcal{V}^l}\right] : l \in [p]\}$ taking their inputs by variables $Y_{\mathcal{V}^l} \subset Y_{\mathcal{V}}$ and output by a variable $Y_{\circ_l}$ We here denote by $\mathcal{F}$ the set of sub-formulas and use a boolean variable Y_h for each $h \in \mathcal{F}$. In particular, we denote for each atom in $\mathcal{F}$ the corresponding boolean variable by Y_k. It then holds*

$$\rho^{f}\left[Y_f, X_{[d]}\right] = \left\langle \{\rho^{\circ_l}\left[Y_{\circ_l}, Y_{\mathcal{V}^l}\right] : l \in [p]\} \cup \{\delta\left[Y_k, X_k\right] \,:\, k \in [d]\} \right\rangle \left[Y_f, X_{[d]}\right] \,.$$

Proof. When a variable in $Y_{\mathcal{F}}$ 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. 125 can be applied). This follows from an iterative application of The. 99 to be shown in Chapter 14. □

Remark 3 (d-ary connectives such as $\wedge$ and $\vee$)**.** *Since the decomposition of relational encoding can be applied to generic function compositions (see The. 99), 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 14.

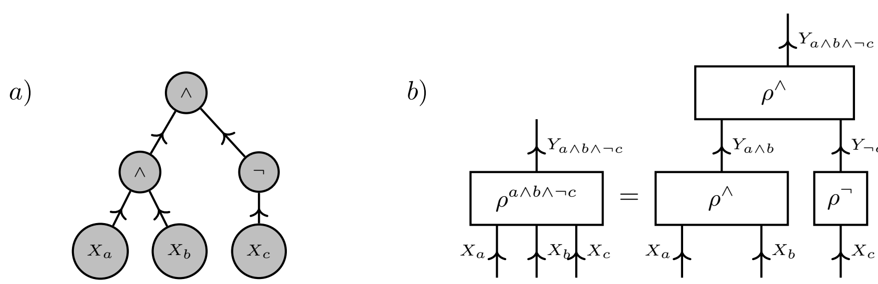


Figure 14: Decomposition of the formula tensor to $f = a \wedge b \wedge \neg c$ into unary (matrix) and binary (third order tensor) cores. a) Visualization of f as a graph. b) Tensor Network decomposition of f. We can make use of the invariance of a Hadamard product with a constant tensor $\mathbb{I}$ and thus not draw axis to atoms not affected by a formula.

Remark 4 (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. 14.5. 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.*

6.3.3 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. 40 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 42 (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 $x_{[d]} \in \times_{k\in[d]}[2]$ 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 $x_0, \dots, x_{d-1} \in \times_{k\in[d]}[2]$ 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 6. *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 $x \in \times_{k\in[d]}[2]$ 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 $x_0, \dots, x_{d-1}$ where $x_k = 0$ if $k \in \mathcal{V}^0$ and $x_k = 1$ if $k \in \mathcal{V}^1$ and define

$$Z^{\vee}_{x_0,\dots,x_{d-1}} = Z^{\vee}_{\mathcal{V}^0,\mathcal{V}^1} \quad \text{and} \quad Z^{\wedge}_{x_0,\dots,x_{d-1}} = Z^{\wedge}_{\mathcal{V}^0,\mathcal{V}^1}.$$

Corollary 3. *Minterms are basis elements of the tensor space, that is for any $x_{[d]} \in \times_{k\in[d]}[2]$ 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 $x_{[d]} \in \times_{k\in[d]}[2]$ we have

$$Z^{\vee}_{x_{[d]}} = \mathbb{I}\left[X_{[d]}\right] - e_{x_{[d]}}\left[X_{[d]}\right].$$

Proof. Follows from Lem. 6, 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 36. *For any boolean tensor $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^2$ with leg-dimensions two we have*

$$T\left[X_{[d]}\right] = \left(\bigvee_{x_{[d]}\,:\,T\left[X_{[d]}=x_{[d]}\right]=1} Z^{\wedge}_{\{k:x_k=0\},\{k:x_k=0\}} \right)\left[X_{[d]}\right]$$

and

$$T\left[X_{[d]}\right] = \left(\bigwedge_{x_{[d]}\,:\,T\left[X_{[d]}=x_{[d]}\right]=0} Z^{\vee}_{\{k:x_k=0\},\{k:x_k=0\}} \right)\left[X_{[d]}\right].$$

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. 36 are also called canonical normal forms to propositional formulas $T\left[X_{[d]}\right]$.

Remark 5 (Efficient Representation in Propositional Syntax)**.** *The decomposition in The. 36 is a basis CP decomposition of the binary tensor and will further be investigated in Chapter 15. The formulas constructed in the proof of The. 36 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. 36. We will provide such constructions in Chapter 15, where we show that dropping the demand of directionality and investigating binary CP Decompositions will improve the sparsity of the polynomial formula representation.*

6.3.4 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.

6.4 Discussion and Outlook

Further study of representing Knowledge Bases based on Tensor Networks of its formulas in Sect. 9.2 (see The. 57).

7 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 with the derivation of contraction criteria for logical entailment. We interpret formulas by distributions and extend logical entailment towards probabilistic reasoning.

7.1 Entailment in Propositional Logics

Entailment is the central consequence relation among logical formulas. Let us define this relation first based on the models of a knowledge base and a test formula.

Definition 43 (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*

$$\forall x_{[d]} \in \bigtimes_{k\in[d]} [2] : \left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \Rightarrow \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

If $\mathcal{KB} \models \neg f$ holds, we say that $\mathcal{KB}$ contradicts f.

To use the tensor network formalism for the decision of entailment, we will in the following develop three equivalent criteria for entailment.

7.1.1 Deciding Entailment by contractions

First of all, we can decide entailment based on vanishing contractions with the negated test formula.

Theorem 37 (Contraction Criterion of Entailment)**.** *We have $\mathcal{KB} \models f$ if and only if*

$$\langle \mathcal{KB}, \neg f \rangle [\varnothing] = 0 .$$

Proof. "$\Leftarrow$": If for a $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have $\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1$ but not $\left(f\left[X_{[d]} = x_{[d]}\right] = 1\right)$, we would have $\left(\neg f\left[X_{[d]} = x_{[d]}\right] = 1\right)$ and

$$\langle \mathcal{KB}, \neg f \rangle [\varnothing] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[2]} \mathcal{KB}\left[X_{[d]} = x_{[d]}\right] \cdot f\left[X_{[d]} = x_{[d]}\right] > 1 .$$

Thus, whenever the contraction vanishes, we have

$$\forall x_{[d]} \in \bigtimes_{k\in[d]} [2] : \left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \Rightarrow \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

"$\Rightarrow$": Conversely, if the contraction $\langle \mathcal{KB}, \neg f \rangle [\varnothing]$ does not vanish, we would find $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ 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. $\square$

The contraction criterion can be extended to the decision of contradiction as well, since $\mathcal{KB} \models \neg f$ is equivalent to $\langle \mathcal{KB}, f \rangle [\varnothing] = 0$. Therefore, entailment and contradiction can be decided simultaneously by a single contraction, as we state next.

Theorem 38. *Given propositional formulas $\mathcal{KB}$ and f we build*

$$T[Y_f] = \left\langle \mathcal{KB}\left[X_{[d]}\right], f\left[X_{[d]}, Y_f\right] \right\rangle [Y_f] \, .$$

Then $\mathcal{KB} \models f$ is equivalent to $T[Y_f = 0] = 0$, and $\mathcal{KB} \models \neg f$ is equivalent to $T[Y_f = 1] = 0$.

Proof. This follows from The. 37 using that

$$\langle \mathcal{KB}, \neg f \rangle [\varnothing] = T[Y_f = 0]$$

and

$$\langle \mathcal{KB}, f \rangle [\varnothing] = T[Y_f = 1] \, .$$

□

7.1.2 Deciding Entailment by partial ordering

Logical entailment can be understood by subset relations of the models of the respective formulas. This perspective can be applied with subset encodings in Chapter 14. The subset relation corresponds with partial ordering of its encoded tensors, as will be shown in The. 95. For two propositional formulas, we denote to this end $f \prec h$ (see Def. 76), if and only if for all $x_{[d]} \in \times_{k\in[d]}[2]$

$$f\left[X_{[d]} = x_{[d]}\right] \leq h\left[x_{[d]}\right] \, .$$

Theorem 39 (Partial Ordering Criterion of Entailment)**.** *We have $\mathcal{KB} \models f$ if and only if $\mathcal{KB}\left[X_{[d]}\right] \prec f\left[X_{[d]}\right]$.*

Proof. Since both $\mathcal{KB}$ and f are boolean tensors, we have for any $x_{[d]} \in \times_{k\in[d]}[2]$ that

$$\mathcal{KB}\left[X_{[d]} = x_{[d]}\right], f\left[X_{[d]} = x_{[d]}\right] \in \{0, 1\} \, .$$

Thus,

$$\forall x_{[d]} \in \bigtimes_{k\in[d]}[2] : \mathcal{KB}\left[X_{[d]} = x_{[d]}\right] \leq f\left[X_{[d]} = x_{[d]}\right]$$

is equivalent to

$$\forall x_{[d]} \in \bigtimes_{k\in[d]}[2] : \left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \Rightarrow \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right) .$$

This states that $\mathcal{KB}\left[X_{[d]}\right] \prec f\left[X_{[d]}\right]$ is equivalent to $\mathcal{KB} \models f$. □

7.1.3 Redundancy of entailed formulas

Another interpretation of entailment is by redundancy of a formula in a Knowledge Base. This is especially interesting for the sparse representation of Knowledge Bases.

Theorem 40 (Redundancy Criterion of Entailment)**.** *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 formula f we have

$$\mathbb{I}\left[X_{[d]}\right] = f\left[X_{[d]}\right] + \neg f\left[X_{[d]}\right]$$

and thus

$$\mathcal{KB}\left[X_{[d]}\right] = \left\langle \mathcal{KB}\left[X_{[d]}\right], \mathbb{I}\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right] = \left\langle \mathcal{KB}\left[X_{[d]}\right], f\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right] + \left\langle \mathcal{KB}\left[X_{[d]}\right], \neg f\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right] \, .$$

Now, by The. 37 we have $\mathcal{KB} \models f$, if and only if $\left\langle \mathcal{KB}\left[X_{[d]}\right], \neg f\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right] = 0$, which is thus equal to

$$\mathcal{KB}\left[X_{[d]}\right] = \left\langle \mathcal{KB}\left[X_{[d]}\right], f\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right] \, .$$

□

7.1.4 Contraction Knowledge Base

We exploit the contraction and redundancy criteria of entailment to sketch an implementation of a propositional Knowledge Base in Algorithm 6. Here the function $\mathrm{ASK}(f)$ returns, whether a formula f is entailed or contradicted by a Knowledge Base. If the formula is neither entailed or contradicted, we say it is contingent. If it is both, we have $\mathcal{KB}\left[X_{[d]}\right] = 0$ and thus an inconsistent Knowledge Base. Exploiting The. 38 we decide these situations based on a single contraction.

The function $\mathrm{TELL}(f)$ incorporates an additional formula f into a Knowledge Base $\mathcal{KB}$. Here we exploit The. 40 and do not add a formula, which is entailed in order to maintain a sparse representation. The function further refuses to add a formula, which would make the Knowledge Base inconsistent (returns Refused) and only changes the Knowledge Base in case of a contingent formula (returns Added).

Algorithm 6 Contraction Knowledge Base

```
ASK(𝒦ℬ, f)
  T[Y_f] ← ⟨{h[X_[d]] : h ∈ 𝒦ℬ}, ρ^f[Y_f, X_[d]]⟩[Y_f]
  if T[Y_f = 0] = 0 and T[Y_f = 1] = 0  then
    return Inconsistent
  end if
  if T[Y_f = 0] = 0 then
    return Entailed
  end if
  if T[Y_f = 1] = 0 then
    return Contradicted
  end if
  return Contingent
TELL(𝒦ℬ, f)
  answer ← ASK(f)
  if answer is Inconsistent: then
    return Inconsistent
  end if
  if answer is Entailed: then
    return Redundant
  end if
  if answer is Inconsistent or Contradicted: then
    return Refused
  end if
  if answer is Inconsistent or Contradicted: then
    𝒦ℬ ← 𝒦ℬ ∪ {f}
    return Added
  end if
```

7.2 Formulas as Random Variables

In order to present logical entailment as extreme cases of more generic probabilistic reasoning, we now provide probabilistic interpretations of propositional formulas. In the next sections, we will investigate two ways of interpreting relational encodings of formulas as conditional probabilities. The atom centric one, which understands the atomic legs as conditions and calculates the truth of the formula, leads to a direct interpretation of ρ^f as a conditional probability distribution. When instead taking the formula itself centric, we get uniform distributions of its models and the complement, when conditioning on the satisfaction of the formula.

7.2.1 Probabilistic queries by formulas

Let $\mathbb{P}\left[X_{[d]}\right]$ be a joint distribution of atomic variables X_k, where $k \in [d]$, taking variables in $m_k = 2$. Let us then ask a query in the formalism of Def. 33, where the query function is assumed to be a propositional formula. The joint distribution can be extended to a variable Y_f representing the satisfaction of a formula f given an assignment to the atoms, by adding its relational encoding as

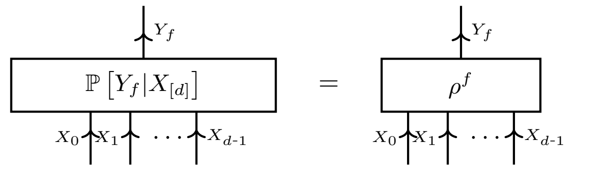
$$\mathbb{P}\left[Y_f, X_{[d]}\right] = \left\langle \rho^f\left[Y_f, X_{[d]}\right], \mathbb{P}\left[X_{[d]}\right]\right\rangle\left[X_{[d]}\right] .$$

Let us note, that this is a normed probability distribution, since $\left\langle \rho^f \left[Y_f, X_{[d]}\right] \right\rangle \left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$ and $\mathbb{P}\left[X_{[d]}\right]$ is normed.

Conditioning this probability distribution on the atoms, we get

$$\mathbb{P}\left[Y_f | X_{[d]}\right] = \rho^f \left[X_{[d]}\right] .$$

We thus interpret the relational encoding of a formula as a conditional probability of f given the assignments to the atoms $X_{[d]}$ and depict this by



To be more precise, we have for any $x_{[d]}$

$$\mathbb{P}\left[Y_f | X_{[d]} = x_{[d]}\right] = \begin{cases} e_0\left[Y_f\right] & \text{if} \quad f\left[X_{[d]} = x_{[d]}\right] = 0, \text{ i.e. } x_{[d]} \text{ is not a model of } f \\ e_1\left[Y_f\right] & \text{if} \quad f\left[X_{[d]} = x_{[d]}\right] = 1, \text{ i.e. } x_{[d]} \text{ is a model of } f \end{cases} .$$

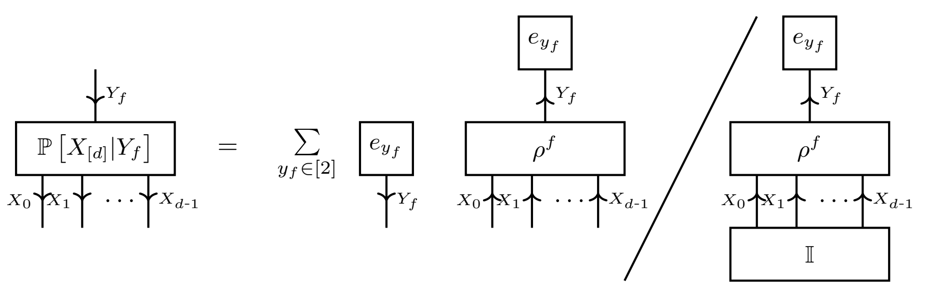
Since the conditional query $\mathbb{P}\left[Y_f|X_{[d]}\right]$ provides an interpretation of ρ^f as a conditional probability, we interpret $\mathbb{P}\left[Y_f\right]$ as a marginal distribution inherited by $\mathbb{P}\left[X_{[d]}\right]$. This is also reflected in the fact that both $\mathbb{P}\left[Y_f|X_{[d]}\right]$ and $\rho^f\left[Y_f, X_{[d]}\right]$ are directed, since the first is a normation by Def. 33 and the second an relational encoding of a formula. Probabilistic queries (see Def. 33), which functions are propositional formulas are thus answered by the satisfaction rate of a propositional formula given a joint distribution of the corresponding atoms.

7.2.2 Uniform distributions of the models

Let us now converse the order of conditioning from $\mathbb{P}\left[Y_f|X_{[d]}\right]$ to $\mathbb{P}\left[X_{[d]}|Y_f\right]$. In this way, we understand a propositional formula as a definition of a joint probability distributions of the atoms, instead of a formulation of a probabilistic query against a joint distribution. To this end, we define by the single tensor core $\{\rho^f\left[Y_f, X_{[d]}\right]\}$ a Markov Network $\mathbb{P}^{\{f\}\cup[d]}\left[Y_f, X_{[d]}\right]$. By definition we have

$$\mathbb{P}^{\{f\}\cup[d]}\left[X_{[d]}|Y_f\right] = \left\langle \rho^f \right\rangle \left[X_{[d]}|Y_f\right] .$$

We depict this construction by:



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

$$\mathbb{P}\left[X_{[d]}|Y_f = 1\right] = \frac{1}{\langle f \rangle [\varnothing]} \sum_{x_{[d]} \in \times_{k\in[d]}[2] \,:\, f\left[X_{[d]}=x_{[d]}\right]=1} e_{x_{[d]}}\left[X_{[d]}\right] . \tag{15}$$

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 (15) is well defined except for the case that $\langle f \rangle [\varnothing] = 0$. In that case we would have $f\left[X_{[d]}\right] = 0\left[X_{[d]}\right]$ 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 (15) correspond within the set of probability distributions with those being constant on their support. While the distributions build a 2^d-1-dimensional manifold, the formulas parametrize by this construction $2^{\left(2^d\right)}$.

7.2.3 Probability of a formula given a Knowledge Base

We 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. **??**

$$\begin{aligned}\mathbb{P}\left[Y_f|Y_{\mathcal{KB}}\right] &= \left\langle \mathbb{P}\left[Y_f|X_{[d]}\right], \mathbb{P}\left[X_{[d]}|Y_{\mathcal{KB}}\right]\right\rangle \left[Y_f, Y_{\mathcal{KB}}\right] \\ &= \left\langle \rho^f, \rho^{\mathcal{KB}}\right\rangle \left[Y_f|Y_{\mathcal{KB}}\right] .\end{aligned}$$

We notice, that we have to assume a satisfiable knowledge base $\mathcal{KB}$ for this construction to be well-defined.

Of special interest is the conditional probability of Y_f given that $Y_{\mathcal{KB}}$ is satisfied, that is

$$\begin{aligned}\mathbb{P}\left[Y_f|Y_{\mathcal{KB}}=1\right] &= \left\langle \{\rho^f, \mathcal{KB}\}\right\rangle \left[Y_f|\varnothing\right] \\ &= \frac{\left\langle \{\rho^f, \mathcal{KB}\}\right\rangle \left[Y_f\right]}{\left\langle \{\mathcal{KB}\}\right\rangle [\varnothing]} .\end{aligned}$$

This conditional probability establishes a connection with the entailment relation of propositional formulas, as we show next.

Theorem 41. *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

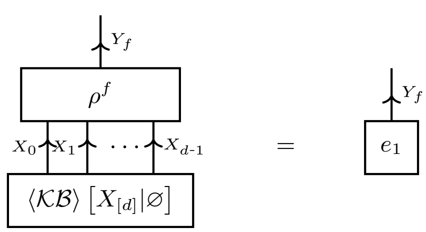
$$\mathbb{P}\left[Y_f=0|Y_{\mathcal{KB}}=1\right] = \frac{\langle \neg f, \mathcal{KB}\rangle [\varnothing]}{\langle \mathcal{KB}\rangle [\varnothing]} .$$

This term vanishes if and only if $\langle \neg f, \mathcal{KB}\rangle [\varnothing]$ vanish. Now, by The. 37 we have $\mathcal{KB} \models f$ if and only if $\langle \mathcal{KB}, \neg f\rangle [\varnothing] = 0$, which is therefore equal to $\mathbb{P}\left[Y_f=0|Y_{\mathcal{KB}}=1\right] = 0$. □

Since any conditional distribution is directed, we have

$$\mathbb{P}\left[Y_f|Y_{\mathcal{KB}}=1\right] = e_0\left[Y_f\right] \text{if} \quad \mathcal{KB} \models \neg f \tag{16}$$
$$e_1\left[Y_f\right] \text{if} \quad \mathcal{KB} \models f \tag{17}$$
$$\notin \left\{e_0\left[Y_f\right], e_1\left[Y_f\right]\right\} \text{else} . \tag{18}$$

We depict the case of entailment $\mathcal{KB} \models f$ by the contraction diagram



.

We can further omit the normation by $\langle \mathcal{KB}\rangle [\varnothing]$ when deciding entailment, and thus drop the assumption of satisfiability of $\mathcal{KB}$, as we state next.

Theorem 42. *Given a formula $\mathcal{KB}$, we have $\mathcal{KB} \models f$ (respectively $\mathcal{KB} \models \neg f$), if and only if*

$$\left\langle \mathcal{KB}, \rho^f\right\rangle \left[Y_f=0\right] = 0 \quad (\textit{ respectively } \left\langle \mathcal{KB}, \rho^f\right\rangle \left[Y_f=1\right] = 0 .$$

Proof. This follows from The. 37 using that

$$\rho^f\left[Y_f=0, X_{[d]}\right] = \neg f\left[X_{[d]}\right] \quad \text{and} \quad \rho^f\left[Y_f=1, X_{[d]}\right] = f\left[X_{[d]}\right] .$$

□

Relating entailment to probability distributions motivates an extension of the entailment provided by Def. 43 to arbitrary probability distributions.

Definition 44. *For any propositional formula $f\left[X_{[d]}\right]$ we say that a probability distribution $\mathbb{P}^{X_{[d]}}$ probabilistically entails f, denoted as $\mathbb{P} \models f$, if*

$$\left\langle \mathbb{P}\left[X_{[d]}\right], \rho^{f}\left[Y_f, X_{[d]}\right]\right\rangle \left[Y_f = 0\right] = 0.$$

If $\mathbb{P} \models \neg f$, that is $\left\langle \mathbb{P}\left[X_{[d]}\right], \rho^{f}\left[Y_f, X_{[d]}\right]\right\rangle \left[Y_f = 1\right] = 0$, we say that $\mathbb{P}$ probabilistically contradicts f.

We note, that when choosing for a formula $\mathcal{KB}$ the uniform distribution $\mathbb{P}\left[X_{[d]}\right] = \left\langle X_{[d]}\right\rangle \left[Y_{\mathcal{KB}} = 1|\varnothing\right]$ among its models, then probabilistic entailment $\mathbb{P} \models f$ of a propositional formula f is by The. 41 equivalent to $\mathcal{KB} \models f$.

7.2.4 Knowledge Bases as Base Measures for Probability Distributions

Let us now further relate the probabilistic entailment provided by Def. 44 with logical entailment, by constructing a corresponding propositional formula to an arbitrary distribution. Given a generic probability distribution $\mathbb{P}$ we can build a Knowledge Base by

$$\mathcal{KB}^{\mathbb{P}} = \mathbb{I}_{\neq 0} \circ \mathbb{P},$$

where $\mathbb{I}_{\neq 0} : \mathbb{R} \rightarrow \mathbb{R}$ denotes the indicator function of the support defined as

$$\mathbb{I}_{\neq 0}(x) = \begin{cases} 0 & \text{if} \quad x = 0 \\ 1 & \text{else} \end{cases}. \tag{19}$$

Probabilistic entailment with respect to $\mathbb{P}$ is then equivalent to entailment with respect to $\mathcal{KB}^{\mathbb{P}}$, as we show next.

Theorem 43. *Any probability distribution $\mathbb{P}\left[X_{[d]}\right]$ probabilistically entails a formula $f\left[X_{[d]}\right]$, if and only if $\mathcal{KB}^{\mathbb{P}} \models f$.*

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 $\mathcal{KB}^{\mathbb{P}}[X_{[d]} = x_{[d]}] = 1$. Therefore the statement

$$\left(\mathcal{KB}^{\mathbb{P}}[X_{[d]} = x_{[d]}] = 1\right) \Rightarrow \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right)$$

is not satisfied. Together, $\mathbb{P} \models f$ does not holds if and only if

$$\forall x_{[d]} \in \bigtimes_{k\in[d]}[2] : \left(\mathcal{KB}^{\mathbb{P}}[X_{[d]} = x_{[d]}] = 1\right) \Rightarrow \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right)$$

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. 17) and positivity (see Def. 18) of distributions with respect to boolean base measures.

Theorem 44. *Let $\mathbb{P}$ be a distribution of boolean variables and let ν be a boolean base measure. Then, $\mathbb{P}$ is representable with respect to ν, if and only if $\mathbb{I}_{\neq 0} \circ \mathbb{P} \models \nu$. Further, $\mathbb{P}$ 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. 17, $\mathbb{P}$ is representable with respect to ν, if and only if

$$\forall x_{[d]} \in \bigtimes_{k\in[d]}[2] : \left(\nu\left[X_{[d]} = x_{[d]}\right] = 0\right) \Rightarrow \left(\mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0\right)$$

This is equal to

$$\forall x_{[d]} \in \bigtimes_{k\in[d]}[2] : \left(\mathbb{I}_{\neq 0} \circ \mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 1\right) \Rightarrow \left(\nu\left[X_{[d]} = x_{[d]}\right] = 1\right)$$

and by definition Def. 43 equal to $\nu \models \mathbb{I}_{\neq 0} \circ \mathbb{P}$.

To proof 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

$$\forall x_{[d]} \in \bigtimes_{k\in[d]} [2] : \left(\nu\left[X_{[d]} = x_{[d]}\right] = 1\right) \Rightarrow \left(\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0\right)$$

This is equal to

$$\forall x_{[d]} \in \bigtimes_{k\in[d]} [2] : \left(\nu\left[X_{[d]} = x_{[d]}\right] = 1\right) \Rightarrow \left(\mathbb{I}_{\neq 0} \circ \mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 1\right)$$

and thus $\nu \models \mathbb{I}_{\neq 0} \circ \mathbb{P}$. □

7.3 Constraint Satisfaction Problems

Let us now explore a more general class of logical inference problems and discuss probabilistic entailment within that class. We then provide further examples based on categorical constraints. Following Chapter 5 in Russell and Norvig (2021), we now define Constraint Satisfaction Problems.

Definition 45. *Let there be a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{T}^{\mathcal{G}}$ be a tensor network of boolean constraint tensors $T^e[X_e]$ to each $e \in \mathcal{E}$, that is*

$$\mathcal{T}^{\mathcal{G}} = \{T^e[X_e] \,:\, e \in \mathcal{E}\}.$$

The Constraint Satisfaction Problem (CSP) to $\mathcal{T}^{\mathcal{G}}$ is the decision whether there is a state $x_{\mathcal{V}}$ such that

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 1\,.$$

We say the CSP is satisfiable, when there is such a state, and unsatisfiable if not.

7.3.1 Deciding entailment on Markov Networks

Deciding entailment on Markov Networks is a general class of contraint satisfaction problems. Here, any factor tensor in the Markov Networks produces a constraint tensor in the respective CSP.

Theorem 45. *Let $\mathbb{P}^{\mathcal{G}}$ be a Markov Network to the Tensor Network $\mathcal{T}^{\mathcal{G}} = \{T^e[X_e] \,:\, e \in \mathcal{E}\}$ on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For each $e \in \mathcal{E}$ we build the factor constraint cores*

$$\tilde{T}^e[e] = \mathbb{I}_{\neq 0} \circ T^e[X_e]\,.$$

Let further $f[X_{\tilde{\mathcal{V}}}]$ be a formula depending on the variables $\tilde{\mathcal{V}}$, and build $\tilde{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \cup \{\tilde{\mathcal{V}}\})$. Then we have that $\mathbb{P}^{\mathcal{G}} \models f$ if and only if the constraint satisfaction problem of $\tilde{\mathcal{G}}$ to the constraint tensors

$$\{\tilde{T}^e \,:\, e \in \mathcal{E}\} \cup \{\neg f\}$$

is unsatisfiable.

Proof. We first show, that

$$\mathbb{I}_{\neq 0} \circ \mathbb{P}^{\mathcal{G}}[X_{\mathcal{V}}] = \left\langle \{\tilde{T}^e \,:\, e \in \mathcal{E}\} \right\rangle [X_{\mathcal{V}}]\,.$$

To this end, let $x_{\mathcal{V}} \in \bigtimes_{v\in\mathcal{V}}[m_v]$ be arbitrary. We have $\mathbb{P}^{\mathcal{G}}[X_{\mathcal{V}} = x_{\mathcal{V}}] = 0$ if and only if at there is an edge $e \in \mathcal{E}$ with $T^e[X_e = x_e]$. But this is equivalent to

$$\left\langle \{\tilde{T}^e \,:\, e \in \mathcal{E}\} \right\rangle [X_{\mathcal{V}}]\,.$$

We thus have for any $x_{\mathcal{V}} \in \bigtimes_{v\in\mathcal{V}}[m_v]$

$$\mathbb{I}_{\neq 0} \circ \mathbb{P}^{\mathcal{G}}[X_{\mathcal{V}} = x_{\mathcal{V}}] = \left\langle \{\tilde{T}^e \,:\, e \in \mathcal{E}\} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}]\,.$$

To continue, we have $\mathbb{P}^{\mathcal{G}} \models f$ if and only if

$$\left\langle \mathcal{T}^{\mathcal{G}}\left[X_{[d]}\right], \neg f\left[X_{[d]}\right] \right\rangle [\varnothing] = 0$$

which is equal to

$$\left\langle \mathbb{I}_{\neq 0} \circ \mathcal{T}^{\mathcal{G}}\left[X_{[d]}\right], \neg f\left[X_{[d]}\right] \right\rangle [\varnothing] = 0 \,.$$

We notice, that this is the unsatisfiability of the claimed Constraint Satisfaction Problem.

□

For any positive tensor T we have

$$\mathbb{I}_{\neq 0} \circ T\left[X_e\right] = \mathbb{I}\left[X_e\right] \,,$$

which does not influence the distribution and can be omitted from the Markov Network. By The. 45, when deciding entailment, we can reduce all tensors of a Markov Network to their support and omit those with full support. Since the support indicating tensors $\mathbb{I}_{\neq 0} \circ T\left[X_e\right]$ are boolean, each is a propositional formula and the Markov Network is turned into a Knowledge Base of their conjunctions. Deciding probabilitic entailment is thus traced back to logical entailment.

Exponential families have a tensor network representation by a Markov Network (see The. 13). However, all factors corresponding with a coordinate of the statistic ϕ have a trivial support, and therefore do not influence the support of the distribution. The only tensors with non-trivial support are those to the boolean base measure ν.

7.3.2 Categorical Constraints

We so far in this chapter 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.

Definition 46 (Categorical Constraint and Atomization Variables)**.** *Given a list $X_0, \dots, X_{m-1}$ of boolean variables and a categorical variable X with dimension m a categorical constraint is a tensor $Z[X, X_{[m]}]$ defined as*

$$Z\left[X_{[m]} = x_{[m]}, X = x\right] = \begin{cases} 1 & \textit{if} \quad x_{[m]} = e_x \\ 0 & \textit{else}\,. \end{cases} \quad \Big(\textit{i.e.}\ \forall k \in [m]\big(x = k\big) \Leftrightarrow \big(x_k = 1\big)\Big)$$

We then call the variables $X_0, \dots, X_{m-1}$ the atomization variables to the categorical variable X.

With The. 116 the relational encoding ρ^Z decomposes in a basis CP format (see Figure 15b) of if its coordinate maps Z^k, where $k \in [m]$, defined as

$$Z^k\left[X_k = x_k, X = x\right] = \begin{cases} 1 & \text{if} \quad x = k \\ 0 & \text{else}\,. \end{cases}$$

Their relational encoding are decomposed as

$$\rho^{Z^k}\left[X_k, X\right] = e_1\left[X_k\right] \otimes e_k\left[X\right] + e_0\left[X_k\right] \otimes \big(\mathbb{I}\left[X\right] - e_k\left[X\right]\big) \,. \tag{20}$$

We further have by The. 116

$$\rho^Z\left[X_{[m]}, X\right] = \left\langle \{\rho^{Z^k}\left[X_k, X\right] \,:\, x \in [m]\} \right\rangle \left[X, X_0, \dots, X_{m-1}\right] \,.$$

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 46. *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*

$$\left\langle T\left[X_{[d]}\right], Z\left[X_A, X\right] \right\rangle \left[X_0 = x_0, \dots, X_{d-1} = x_{d-1}\right] = \begin{cases} T\left[X_0 = x_0, \dots, X_{d-1} = x_{d-1}\right] & \textit{if} \quad x_A = e_x \\ 0 & \textit{else}\,. \end{cases}$$

Here by x_A we denote the restriction of $x_{[d]}$ on the set A.

Proof. For any $x_{[d]}$ we have

$$\left\langle \left\{ T\left[X_{[d]}\right], Z \right\} \right\rangle [X_0 = x_0, \dots, X_{d-1} = x_{d-1}] = T\left[x_{[d]}\right] \cdot Z[X_A = x_A, X = x] .$$

If $x_A = e_x$ we have $Z[X_A = x_A, X = x] = 1$ and thus

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

If $x_A \neq e_x$ then $Z[X_A = x_A, X = x] = 0$ and

$$\left\langle T\left[X_{[d]}\right], Z \right\rangle [X_0 = x_0, \dots, X_{d-1} = x_{d-1}] = 0 .$$

□

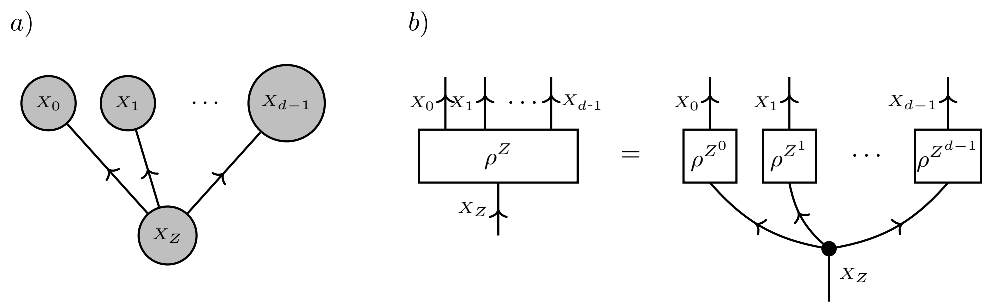


Figure 15: Representation of a categorical constraint in a CP Format tensor network. a) Representation of the dependency of the graphical model. b) Tensor Representation with further network decomposition.

Remark 6 (Constraint Satisfaction Problems of Categorical Constraints)**.** *We can define CSPs by collection of categorical constraints. An example, where the corresponding Constraint Satisfaction Problem is unsatisfiable are the categorical constraints to the three sets*

$$\{X_0, X_1, X_2, X_3\}, \{X_0, X_1\}, \{X_2, X_3\} .$$

Example 7 (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 Russell and Norvig (2021)). 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 16 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.

Deciding whether a Sudoku has a solution is a Constraint Satisfaction Problem Simonis (2005), which is NP-hard Agerbeck and Hansen (2008). Let us notice, that due to this large number of variables and constraints, direct solution of the problem by a global contraction is not feasible. For efficient algorithmic solutions, we instead refer to Sect. 7.4.

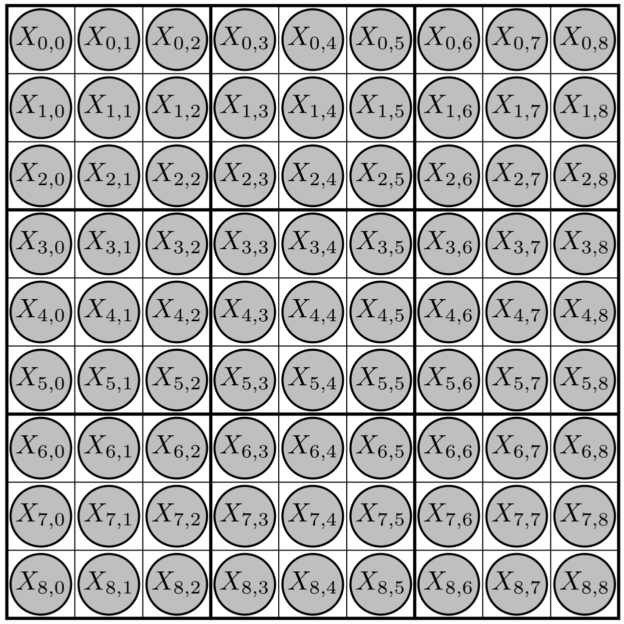


Figure 16: Sudoku grid of basic categorical variables $X_{i,j}$, here drawn in the standard case of $n = 3$, each with dimension $m = n^2 = 9$. Each basic categorical variables has n^2 corresponding atomization variables, which are further atomization variables to the row, column and squares constraints. Instead of depicting those constraints by hyperedges in a variable dependency graph, we here just indicate their existence through row, column and squares blocks.

7.4 Deciding Entailment by local contractions

When having a Constraint Satisfaction Problem on a large number of variables, which are densely connected by constraint tensors, direct exploitation of the global entailment criterion in The. 37 will be infeasible. 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.

7.4.1 Monotonicity of entailment

Vanishing local contractions provide sufficient but not necessary criterion to decide entailment, as we show in the next theorem.

Theorem 47 (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 $\mathbb{P}^{\tilde{\mathcal{G}}} \models f$.*

To prove the theorem, we first establish the following lemma that states if a contraction of non-negative tensors vanishes, the vanishing of a contraction over a subset of these tensors is a sufficient criterion.

Lemma 7. *For any non-negative tensor network $\mathcal{T}^{\mathcal{G}}$ on $\mathcal{G}$ and $\tilde{\mathcal{E}} \subset \mathcal{E}$ we have the following. For $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ with $\tilde{\mathcal{V}} = \cup_{e\in\tilde{\mathcal{E}}} e$ and the tensor network $\mathcal{T}^{\tilde{\mathcal{G}}}$ with tensors coinciding on $\tilde{\mathcal{E}}$ with those in $\mathcal{T}^{\mathcal{G}}$ we have*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing] = 0$$

if $\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [\varnothing] = 0$.

Proof. Since the tensor network $\mathcal{T}^{\tilde{\mathcal{G}}}$ is non-negative, we have whenever $\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [\varnothing] = 0$ that

$$\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] = 0 \left[X_{\tilde{\mathcal{V}}}\right] .$$

It follows with the commutation of contractions (see The. 121 in Chapter 17), that

$$\begin{aligned}\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}] &= \left\langle \{T^e \,:\, e \in \mathcal{E}/\tilde{\mathcal{E}}\} \cup \{\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}]\} \right\rangle [X_{\mathcal{V}}] \\ &= \left\langle \{T^e \,:\, e \in \mathcal{E}/\tilde{\mathcal{E}}\} \cup \{0\, [X_{\tilde{\mathcal{V}}}]\} \right\rangle [X_{\mathcal{V}}] \\ &= 0\end{aligned}$$

Thus, also the contraction of $\mathcal{T}^{\mathcal{G}}$ vanishes in this case. □

Proof of The. 47. We use Lem. 7 on the subset $\mathcal{T}^{\tilde{\mathcal{G}}}$ of the cores $\mathcal{T}^{\mathcal{G}}$ to the Markov Network $\mathbb{P}^{\mathcal{G}}$, which itself defines the Markov Network $\mathbb{P}^{\tilde{\mathcal{G}}}$. Whenever $\mathbb{P}^{\tilde{\mathcal{G}}} \models f$ for a formula f, then we have by The. 37

$$\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \cup \{\neg f\} \right\rangle [\varnothing] = 0\,.$$

It follows with Lem. 7 that also

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{\neg f\} \right\rangle [\varnothing] = 0\,.$$

and therefore $\mathbb{P}^{\mathcal{G}} \models f$. □

Remark 7. *To make use of The. 47 we can exploit any entailment criterion. However, there is no general statement about entailment possible, when the local entailment does not hold. The. 47 therefore just provides a sufficient but not necessary criterion of entailment with respect to $\mathbb{P}^{\mathcal{G}}$.*

7.4.2 Knowledge Cores

To store preliminary conclusions, we define auxiliary knowledge cores storing constraints on variables $e \subset \mathcal{V}$. They are understood as logical formulas to the atomization variables $(X_e = x_e)$ of the respective formulas

$$K^e [X_e] = \bigvee_{x_e \,:\, K^e[X_e = x_e]} \bigwedge_{v \in e} (X_e = x_e)\ .$$

Definition 47. *Let $\mathcal{T}^{\mathcal{G}}$ be a constraint satisfaction problem. We say that a knowledge core $K^e [X_e]$ is sound for $\mathcal{T}^{\mathcal{G}}$, if*

$$\mathbb{I}_{\neq 0} \circ \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_e] \prec K^e [X_e]$$

and complete for $\mathcal{T}^{\mathcal{G}}$ if in addition

$$\mathbb{I}_{\neq 0} \circ \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_e] = K^e [X_e]\ .$$

7.4.3 Knowledge Propagation

We now provide a solution algorithm for constraint satisfaction problems by propagating local contractions.

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 **??**.

Each chosen subset $\tilde{\mathcal{E}} \in \mathcal{U}$ is understood as a local knowledge base, which is then applied for local entailment. The knowledge cores are understood as messages, which propagate information from different regions of a tensor network (see Chapter 17).

There are different ways of implementing Algorithm 7, by choosing the set $\mathcal{U}$ of constraint sets $\tilde{\mathcal{E}}$ and domain $\mathcal{E}^k$. The AC-3 algorithm (see Mackworth (1977)) is a specific instance, where knowledge cores are assigned to single variables and propagation is performed on single constraint cores.

Theorem 48. *At any state of the Knowledge Propagation Algorithm 7, we have that each knowledge core $K^{\tilde{e}}$ is sound for $\mathcal{T}^{\mathcal{G}}$. After each update in Algorithm 7, $K^{\tilde{e}}$ is further monotonically decreasing with respect to the partial ordering.*

Algorithm 7 Knowledge Propagation (KP)

Boolean Tensor Network $\mathcal{T}^{\mathcal{G}}$ on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set of domain edges $\mathcal{E}^k$ and a set $\mathcal{U}$ of subsets of $\mathcal{E}$
Initialize for all $e \in \mathcal{E}^k$:

$$K^e [X_e] = \mathbb{I} [X_e]$$

Initialize a queue

$$\mathcal{Q} = \mathcal{U}$$

while $\mathcal{Q}$ is not empty **do**
Choose a set of edges from the queue

$$\tilde{\mathcal{E}} \leftarrow \mathcal{Q}.\mathrm{pop}()$$

for $e \in \mathcal{E}^k$ with $e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing$ **do**
Contract

$$T [X_e] = \mathbb{I}_{\neq 0} \circ \left\langle \{T^{\tilde{e}} [X_{\tilde{e}}] \,:\, \tilde{e} \in \tilde{\mathcal{E}}\} \cup \{K^e [X_e] \,:\, e \in \mathcal{E}^k, \, e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing\} \right\rangle [X_e]$$

if $T [X_e] \neq K^e [X_e]$ **then**

$$K^e [X_e] \leftarrow T [X_e]$$

for $\tilde{\mathcal{E}} \in \mathcal{U}$ with $e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing$ **do**

$$\mathcal{Q}.\mathrm{push}(\tilde{\mathcal{E}})$$

end for
end if
end for
end while

Proof. We show the first claim by induction over the update steps in Algorithm 7. At the start, where $K^e [X_e] = \mathbb{I} [X_e]$, we trivially have

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{K^e [X_e] \,:\, e \in \mathcal{E}^k\} \right\rangle [X_{\mathcal{V}}] = \left\langle \mathcal{T}^{\mathcal{G}} \cup \{\mathbb{I} [X_e] \,:\, e \in \mathcal{E}^k\} \right\rangle [X_{\mathcal{V}}] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}] \,.$$

Let us now assume, that for a state of cores $\{K^e \,:\, e \in \mathcal{E}^k\}$ the first claim holds and let $\tilde{\mathcal{E}} \subset \mathcal{E}$ be chosen for the update of $K^{\tilde{e}}$. By the invariance under adding the support of subcontractions, which we will proof in more detail as The. 124 in Chapter 17, we have for the update

$$\tilde{K}^{\tilde{e}}[X_{\tilde{e}}] = \mathbb{I}_{\neq 0} \circ \left\langle \{T^e \,:\, e \in \tilde{\mathcal{E}}\} \cup \{K^e \,:\, e \in \mathcal{E}^k, \, e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing\} \right\rangle [X_{\tilde{e}}]$$

that

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{K^e [X_e] \,:\, e \in \mathcal{E}^k\} \right\rangle [X_{\mathcal{V}}] = \left\langle \mathcal{T}^{\mathcal{G}} \cup \{K^e [X_e] \,:\, e \in \mathcal{E}^k\} \cup \{\tilde{K}^{\tilde{e}}[X_{\tilde{e}}]\} \right\rangle [X_{\mathcal{V}}] \,.$$

Thus, the first claim holds also after the update of the core to $\tilde{e}$.

We further have with the monotonocity of boolean contraction (see The. 124) that for any update of $K^{\tilde{e}}$ by $\tilde{K}^{\tilde{e}}$

$$\tilde{K}^{\tilde{e}}[X_{\tilde{e}}] = \mathbb{I}_{\neq 0} \circ \left\langle \{T^e \,:\, e \in \tilde{\mathcal{E}}\} \cup \{K^e \,:\, e \in \mathcal{E}^k, \, e \cap \bigcup_{e \in \tilde{\mathcal{E}}} e \neq \varnothing\} \right\rangle [X_{\tilde{e}}] \prec K^{\tilde{e}} [X_{\tilde{e}}] \,.$$

Thus, each Knowledge Core is monotoneously decreasing at each update, with respect to the partial tensor ordering.

From the first claim we further have for any $\tilde{e} \in \tilde{\mathcal{E}}$

$$\left\langle \{K^{\tilde{e}} [X_{\tilde{e}}]\} \cup \mathcal{T}^{\mathcal{G}} \cup \left\{K^e \,:\, e \in \mathcal{E}^k / \{\tilde{e}\}\right\} \right\rangle [X_{\tilde{e}}] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{e}}]$$

And thus in combination with the monotonocity of boolean contraction (see The. 124) that

$$\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{e}}]\right) \prec K^{\tilde{e}}\left[X_{\tilde{e}}\right] .$$

□

Let us now show that Knowledge Propagation always terminates. We can further characterize the knowledge cores at termination.

Definition 48. *We say that a set of knowledge cores $\{K^e : e \in \mathcal{E}^k\}$ is consistent with a set $\{T^e : e \in \tilde{\mathcal{E}}\}$, if for any $e \in \mathcal{E}^k$*

$$K^e\left[X_e\right] = \mathbb{I}_{\neq 0} \circ \left\langle \{T^e : e \in \tilde{\mathcal{E}}\} \cup \{K^e : e \in \mathcal{E}^k\} \right\rangle [X_e] .$$

This property is similar to the completeness of a knowledge core, when interpreting the other knowledge cores and the constraints $\{T^e : e \in \tilde{\mathcal{E}}\}$ as posing a Constraint Satisfaction Problem.

Theorem 49. *Knowledge Propagation Algorithm 7 always terminates. At termination we further for each $\tilde{\mathcal{E}} \in \mathcal{U}$ and $e \in \mathcal{E}^k$ with $e \cap \bigcup_{e\in\tilde{\mathcal{E}}} e \neq \varnothing$, that the knowledge cores $\{K^e : e \in \mathcal{E}^k, e \cap \bigcup_{e\in\tilde{\mathcal{E}}} e \neq \varnothing\}$ are consistent with $\{T^e : e \in \tilde{\mathcal{E}}\}$.*

Proof. For each knowledge core, there are finitely many boolean tensor precessing it with respect to the partial order. Therefore, since they are monotonously decreasing, each knowledge core can only be varied finitely many times during the algorithm. In total the algorithm can run only finitely many times in the second for loop, where new sets of edges are pushed into the queue. Therefore the while loop will always terminate.

When after a single pass through the while loop with chosen $\tilde{\mathcal{E}} \in \mathcal{U}$, the set $\tilde{\mathcal{E}}$ is not pushed back into $\mathcal{Q}$, we have for any $e \in \mathcal{E}^k$ with $e \cap \bigcup_{e\in\tilde{\mathcal{E}}} e \neq \varnothing$ that

$$K^e\left[X_e\right] = \mathbb{I}_{\neq 0} \circ \left\langle \{T^e : e \in \tilde{\mathcal{E}}\} \cup \{K^e : e \in \mathcal{E}^k, e \cap \bigcup_{e\in\tilde{\mathcal{E}}} e \neq \varnothing\} \right\rangle [X_e] .$$

Whenever the contraction on the right hand side changes during the algorithm, the set $\tilde{\mathcal{E}}$ is pushed into $\mathcal{Q}$. At termination of the algorithm, $\mathcal{Q}$ is empty, and the claimed consistency therefore has to hold. □

We can exploit the Knowledge Propagation Algorithm 7 for the solution of Constraint Satisfaction Problems, by taking $\mathcal{T}^{\mathcal{G}}$ as the tensor network of constraint tensors. Whenever a knowledge core vanishes, we can conclude that the Constraint Satisfaction Problems is not satisfiable, as we show next.

Corollary 4. *Let us for a Constraint Satisfaction Problem encoded by $\mathcal{T}^{\mathcal{G}}$ run Knowledge Propagation Algorithm 7. Whenever for any $\tilde{e} \in \mathcal{E}$ we have $K^{\tilde{e}}\left[X_{\tilde{e}}\right] = 0\left[X_{\tilde{e}}\right]$, then the Constraint Satisfaction Problem is not satisfiable.*

Proof. Whenever $K^{\tilde{e}}\left[X_{\tilde{e}}\right] = 0\left[X_{\tilde{e}}\right]$, then we have by The. 48

$$\mathbb{I}_{\neq 0}\left(\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{e}}]\right) \prec 0\left[X_{\tilde{e}}\right]$$

and therefore

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing] = 0 .$$

□

When the Knowledge Propagation Algorithm 7 converges in a given implementation and no knowledge core vanishes, we can however not conclude that the Constraint Satisfaction Problem is not satisfiable. However, for any index tuple $x_{\mathcal{V}}$ to be a solution of the CSP to $\mathcal{T}^{\mathcal{G}}$, we have the necessary condition

$$\forall \tilde{e} \in \tilde{\mathcal{E}} : K^e\left[X_{\tilde{e}} = x_{\mathcal{V}}|_{\tilde{\mathcal{E}}}\right] = 1 ,$$

where by $x_{\mathcal{V}}|_{\tilde{\mathcal{E}}}$ we denote the restriction of the index tuple $x_{\mathcal{V}}$ to the variables included in $\tilde{\mathcal{E}}$. One can use this insight as a starting point for backtracking search, where the assignments to variables $X_{\tilde{\mathcal{V}}}$ are iteratively guessed, based on the restriction that each constraint is locally satisfialbe, i.e. .

$$\forall \tilde{e} \in \tilde{\mathcal{E}} : \left\langle K^e\left[X_{\tilde{e}\cap\tilde{\mathcal{V}}} = x_{\mathcal{V}}|_{\tilde{e}\cap\tilde{\mathcal{V}}}, X_{\tilde{e}/\tilde{\mathcal{V}}}\right] \right\rangle [\varnothing] \neq 0 .$$

One can understand the guess of an assignment x_v to a variable X_v, as it is done during backtracking search, as an inclusion of a constraint

$$K^{\{v\}}[X_v] = e_{x_v}[X_v] \, .$$

Therefore, Knowledge Propagation Algorithm 7 can be integrated with backtracking search, with iterations between propagations of knowledge and guessing of additional variables.

7.4.4 Applications

Let us examplify the usage of Knowledge Propagation on Constraint Satisfaction Problems posed by entailment queries on Markov Networks.

Corollary 5. *Let Algorithm 7 be run on the cores $\mathcal{T}^{\mathcal{G}} \cup \{\rho^f\}$ with an arbitrary design of $\tilde{\mathcal{E}}$. Whenever for a formula $f[X_{\tilde{\mathcal{V}}}]$ and a K^e we have*

$$\left\langle K^e, \rho^f \right\rangle [Y_f = 0] = 0$$

then the Markov Network $\mathcal{T}^{\mathcal{G}}$ probabilistically entails f. If on the contrary

$$\left\langle K^e, \rho^f \right\rangle [Y_f = 1] = 0$$

then the Markov Network $\mathcal{T}^{\mathcal{G}}$ probabilistically entails $\neg f$, that is probabilistically contradicts f.

Proof. This follows from The. 48 ensuring the soundness of Knowledge Propagation and the sufficiency of local entailment. □

Example 8 (Batch decision of entailment)**.** *Let $\mathcal{F}$ be a set of formulas and $\mathbb{P}^{\mathcal{G}}\left[X_{[d]}\right]$ a Markov Network, for which it shall be decided, which formulas in $\mathcal{F}$ are entailed, contradicted or contingent. We can in addition to the cores of the Markov Network create the cores $\{\rho^f\left[Y_f, X_{[d]}\right] \,:\, f \in \mathcal{F}\}$ and prepare the knowledge cores*

$$K^{\{f\}}[Y_f] \, .$$

To decide entailment batchwise, Knowledge Propagation Algorithm 7 can be run. Whenever during the algorithm we have that for a f, then Cor. 5 implies that if

$$K^{\{f\}}[Y_f] = \begin{cases} e_1[Y_f] & \text{then, the formula is entailed by } \mathbb{P}^{\mathcal{G}} \, . \\ e_0[Y_f] & \text{then, the formula is contradicted by } \mathbb{P}^{\mathcal{G}} \, . \\ \mathbb{I}[Y_f] & \text{then no conclusion can be drawn.} \end{cases}$$

Note, that $K^{\{f\}}[Y_f] = 0[Y_f]$ can not happen, since this would mean that $\mathbb{I}_{\neq 0}\left(\mathbb{P}^{\mathcal{G}}\right)$ is inconsistent. Thus, at any stage of Algorithm 7, one of the three holds.

7.4.5 Mimiking Inference Rules by Propagation

While so far we have discussed semantic based entailment, there are inference rules exploiting only logical syntax to infer entailed statements. We here show, that they can be captures by the knowledge propagation scheme, if the sets $\mathcal{U}$ and $\mathcal{E}^k$ are chosen properly.

Whenever

$$\bigvee_{f \in \mathcal{F}} f \models h$$

then

$$e_1[Y_h] = \left\langle \{f\left[X_{[d]}\right] \,:\, f \in \mathcal{F}\} \cup \{\rho^h\left[Y_h, X_{[d]}\right]\} \right\rangle [Y_h] \, ,$$

that is the inference rule can be performed in when $\mathcal{F}$ are in $\mathcal{U}$.

Example 9. *Modul Ponens For example, when for two formulas $f, h \in \mathcal{F}$ we have $f \models h$, then when $K^{\{f\}}[Y_f] = e_1[Y_f]$ we have*

$$e_1[Y_h] = \left\langle (f \Rightarrow h)[Y_f, Y_h], K^{\{f\}}[Y_f] \right\rangle [Y_h] \, ,$$

that is entailment of h can be concluded using a single update.

When we have a Knowledge Base of horn clauses, we run Knowledge Propagation with each horn clause being a constraint core and a knowledge core for any variable. Algorithm 7 therefore resembles the forward chaining algorithm of propositional logics (see Figure 7.15 in Russell and Norvig (2021)). It is known, that forward chaining is complete for Horn Logic. Thus, the knowledge cores returned in that case by Algorithm 7 are complete for the Knowledge Base as a CSP.

7.5 Discussion

Remark 8 (Interpretation of Contractions in Logical Reasoning)**.** *The coordinates of contracted boolean 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 world count.*

This is consistent with the probabilistic interpretation of contractions, when applying the frequentist interpretation of probability and defining normed worldcounts as probabilities.

Remark 9. *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.*

Local contraction approaches in inference, especially when orchestrated by a Knowledge Propagation algorithm, mimik inference rules in syntax-based prove approaches.

Part II

Neuro-Symbolic Approaches

8 Formula Selecting Networks

In this chapter we will investigate efficient schemes to represent collections of formulas with similar structure in one tensor network.

Definition 49. *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 17.

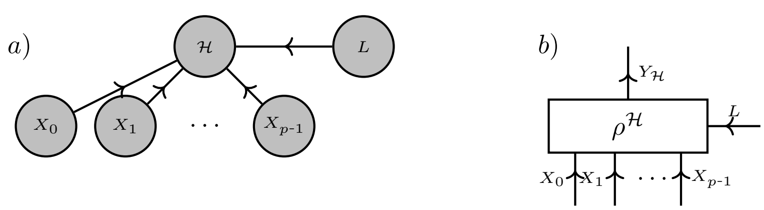


Figure 17: Representation of the Formula Selecting map as a a) Graphical Model with a selection variable $\mathcal{H}$. b) Decorating Tensor Core with selection variable corresponding with an additional axis.

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.

8.1 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.

8.1.1 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 50. *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 18.

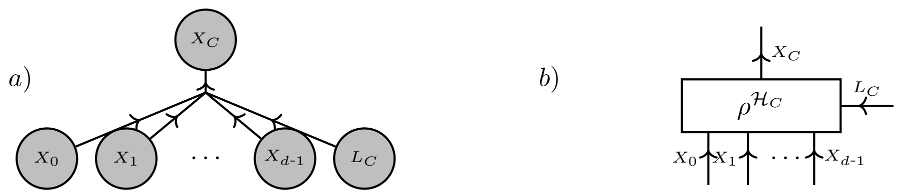


Figure 18: Connective Selector.

Remark 10 (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.*

8.1.2 Variable Selecting Tensor Network

Definition 51. *The selection of one out of p variables in a list $X_{[p]}$ is done by variable selecting maps*

$$\mathcal{H}_V\left[X_{[p]}, L_V\right] : \left(\bigtimes_{l\in[p]} [2]\right) \times [p] \to [2] \tag{21}$$

are defined coordinatewise by

$$\mathcal{H}_V\left[X_0 = x_0, \ldots, X_{p-1} = x_{p-1}, L_V = l_V\right] = x_l . \tag{22}$$

Variable selecting maps appear in the literature as multiplex gates (see Definition 5.3 in Koller and Friedman (2009)).

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 50 (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 19)

$$\rho^{\mathcal{H}_V}\left[Y_V, X_{[p]}, L_V\right] = \left\langle\left\{T^{V_{l_V}}\left[Y_V, X_{l_V}, L_V\right] \,:\, l_V \in [p_V]\right\}\right\rangle\left[Y_V, X_{[p]}, L_V\right] .$$

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

$$\begin{aligned}
&\left\langle \{T^{V_{l_V}}\left[Y_V, X_{l_V}, L_V\right] \,:\, l_V \in [p_V]\} \right\rangle \left[Y_V = y_V, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V\right] \\
&\quad = \prod_{l_V\in[p_V]} T^{V_{l_V}}\left[Y_V = y_V, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V\right] \\
&\quad = T^{V_{\tilde{l}_V}}\left[Y_V = y_V, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V\right] \\
&\quad = \begin{cases} 1 & \text{if} \quad y_V = x_{l_V} \\ 0 & \text{else} \end{cases} \\
&= \rho^{\mathcal{H}_V}\left[Y_V = y_V, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V\right]
\end{aligned}$$

In the second equality, we used that the tensor $T^{V_{l_V}}$ have coordinates 1 whenever $\tilde{l}_V \neq l_V$. $\square$

The decomposition provided by The. 50 is in a CP format, as will be further discussed in Chapter 15. The introduced tensors $T^{V_{l_V}}$ are Boolean, but not directed and therefore encodings of relations but not functions (see Chapter 14).

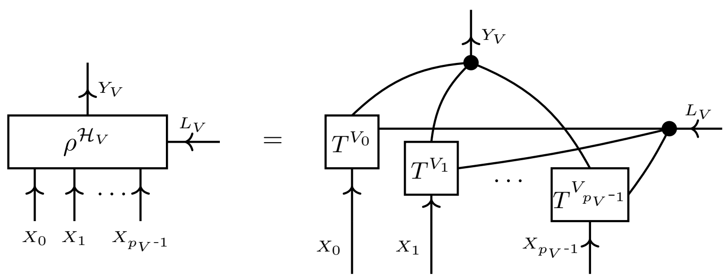


Figure 19: Decomposition of the relational encoding of a variable selecting tensor into a network of tensors defined in The. 50. The decomposition is in a CP-Format (see Chapter 15.

8.2 State Selecting Tensors

As an alternative, one can select a state of a categorical variable X.

Definition 52. *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

$$\mathcal{H}_S\left[X = x, L_S = l_S\right] = \begin{cases} 1 & \textit{if} \quad x = l_S \\ 0 & \textit{else} \end{cases} .$$

State selecting tensors can also be realized by variable selecting tensors. In Sect. 7.3.2 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 $X = 1$.

8.3 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 53. *Let there be a formula $f_{l_0,\dots,l_{n-1}}$ for each index tuple in $l_0,\dots,l_{n-1} \in \bigtimes_{s\in[n]}[p_s]$, where $n, p_0, \dots, p_{n-1} \in \mathbb{N}$. The folded formula selector map (see Figure 20) 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. 51) and connective selector maps (Def. 50). This is especially useful to provide efficient decompositions of relational encodings.

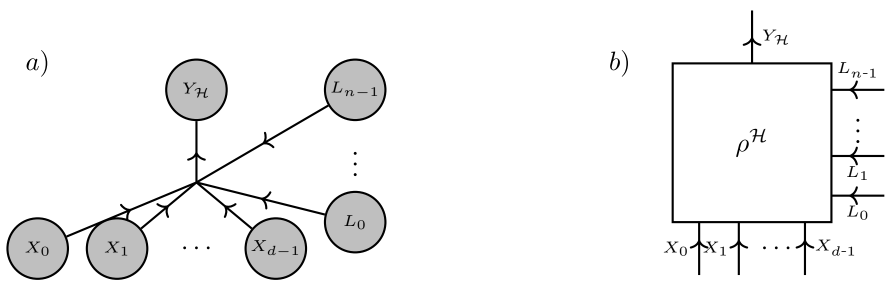


Figure 20: Relational encoding of the folded map $\mathcal{H}$.

8.3.1 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 54. *Given an order $n \in \mathbb{N}$ let there be a connective selector $L_\circ$ selecting connectives of order n and let $\mathcal{H}_{V,0}, \dots, \mathcal{H}_{V,n-1}$ 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 $l_0, \dots, l_{n-1} \in \bigtimes_{s\in[n]}[p_s]$ by

$$\sigma(x_0,\dots,x_{d-1},l_C,l_0,\dots,l_{n-1}) = \mathcal{H}_C(\mathcal{H}_{V,0}(x_0,\dots,x_{d-1},l_0),\dots,\mathcal{H}_{V,n-1}(x_0,\dots,x_{d-1},l_{n-1}),l_C) .$$

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 51. *Decomposition of formula selecting neurons Let σ a logical neuron, defined for a connective selector $L_\circ$ and variable selectors $\mathcal{H}_{V,0}, \dots, \mathcal{H}_{V,n-1}$. Then we have (see Figure 21 for the example of $n = 2$):*

$$\begin{aligned}
&\rho^{\sigma}\left[Y_\sigma, X_{[d]}, L_C, L_{V,0}, \dots, L_{V,n-1}\right] \\
&\quad = \langle\{\rho^{\mathcal{H}_C}\left[Y_\sigma, Y_{V,0}, \dots, Y_{V,n-1}\right], \\
&\qquad \rho^{\mathcal{H}_{V,0}}\left[Y_{V,0}, X_{[d]}, L_{V,0}\right], \dots, \rho^{\mathcal{H}_{V,n-1}}\left[Y_{V,n-1}, X_{[d]}, L_{V,n-1}\right]\}\rangle\left[Y_\sigma, X_{[d]}, L_C, L_{V,0}, \dots, L_{V,n-1}\right] .
\end{aligned}$$

Proof. By composition The. 99. □

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

$$\mathcal{F}_S \coloneqq \left\{S\left(Z^1,\dots,Z^d\right) \,:\, Z^k \in \mathcal{M}^k\right\}$$

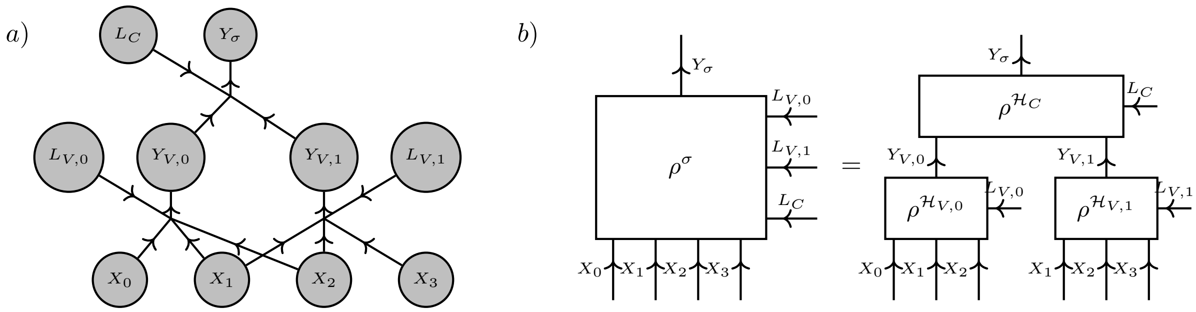


Figure 21: Example of a logical neuron σ of order $n = 2$. a) Selection and categorical variables and their interdependencies visualized in a hypergraph. b) Relational encoding of the logical neuron and tensor network decomposition into variable selecting and connective selecting tensors.

8.3.2 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 55. *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 22 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 $(\{Y_\sigma\}, X_{\text{Pa}(\sigma)} \cup L_\sigma)$.

Theorem 52. *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. 106), we have

$$\left\langle \rho^{\wedge}\left[X_{\wedge}, X_{[d]}\right], e_1\left[X_{\wedge}\right]\right\rangle \left[X_{[d]}\right] = \bigotimes_{k \in [d]} e_1\left[X_k\right]$$

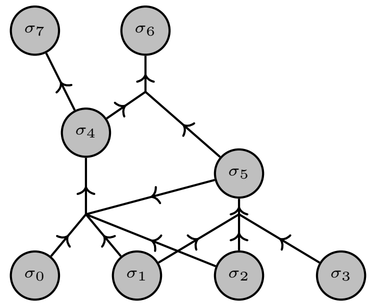


Figure 22: Example of an architecture graph $\mathcal{G}^{\mathcal{A}}$ with input neurons $\mathcal{A}^{\text{in}} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$ and output neurons $\mathcal{A}^{\text{out}} = \{\sigma_6, \sigma_7\}$

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. 51 to decompose each tensor ρ^{σ} into connective and variable selecting components to get a sparse representation of a formula selecting neural network $\mathcal{H}_{\mathcal{A}}$.

8.4 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.

8.4.1 Representation of selection encodings

In technical perspective: FSN provide efficient representation of $\gamma^{\mathcal{F}}$ -> Use for exponential families, structure learning.

Lemma 8. *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*

$$\mathcal{F} : x_0, \ldots, x_{d-1} \rightarrow (f_{l_0,\ldots,l_{n-1}}(x_0, \ldots, x_{d-1}))_{l_0,\ldots,l_{n-1}} .$$

and the formula selecting map

$$\mathcal{H} : x_0, \ldots, x_{d-1}, l_0, \ldots, l_{n-1} \rightarrow f_{l_0,\ldots,l_{n-1}}(x_0, \ldots, x_{d-1}) .$$

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.

8.4.2 Efficient Representation of Formulas

Weight contracted at the selection variables is elementary, then single formula retrieved.

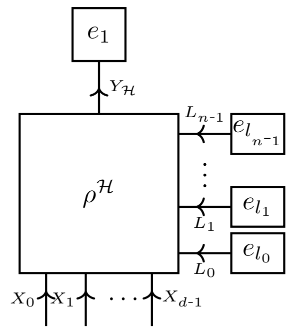
Formula Selecting Neural Networks are means to represent exponentially many formulas with linear (in sum of candidates list lengths) storage. Their contraction with probability tensor networks, is thus a batchwise evaluation of exponentially many formulas. This is possible due to redundancies in logical calculus due to modular combinations of subformulas.

We can retrieve specific formulas by slicing the selection variables, i.e. for $l_0, \ldots, l_{n-1}$ we have

$$f_{l_0,\ldots,l_{n-1}}[X_{[d]}] = \mathcal{H}\left[X_{[d]}, L = l_0, \ldots, l_{n-1}\right] .$$

In a tensor network diagram we depict this by

$f_{l_0,\ldots,l_{n-1}} =$



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.

8.4.3 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

$$\left\langle \mathcal{T}^{\mathcal{G}}, f_{l_{[n]}} \right\rangle [\varnothing] = \left\langle \mathcal{T}^{\mathcal{G}}, \mathcal{H} \right\rangle \left[L_{[n]} = l_{[n]}\right] .$$

8.4.4 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 10 (Average satisfaction of formulas)**.** *The average of the formula satisfactions in $\mathcal{F}$ giben a probability tensor $\mathbb{P}$ is*

$$\frac{1}{\prod_{s\in[n]} p_s} \cdot \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle [\varnothing] .$$

Example 11 (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. 8 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

$$\mathcal{KB} \models \neg \left(\bigvee_{f\in\mathcal{F}} f \right) .$$

8.5 Examples of formula selecting neural networks

8.5.1 Correlation

For example (see Figure 23) 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

$$T\left[L_{V,0}, L_{V,1}\right] = \langle \mathcal{H}_{\mathcal{A}} \rangle \left[L_{V,0}, L_{V,1}\right]$$

to read of covariances as

$$\mathrm{Cov}(X_k, X_l) = T\left[L_{V,0} = k, L_{V,1} = l\right] - T\left[L_{V,0} = k, L_{V,1} = k\right] \cdot T\left[L_{V,0} = l, L_{V,1} = l\right] .$$

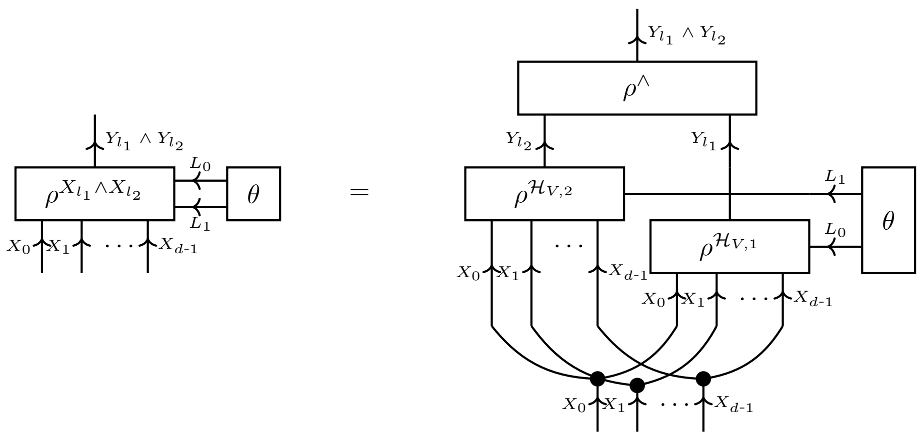


Figure 23: Superposition of the encoded formulas $\rho^{X_{l_1} \wedge X_{l_2}}$ with weight $\theta_{l_1 l_2}$

8.5.2 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 11 (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* $\neg/\mathrm{Id}$ *connective selection neuron to each atom and the upper layer consists of a single dary conjunction.*

8.6 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

$$m_{\mathcal{H}_C} = \left|\cup_{l\in[p]} \mathrm{im}\left(h_l\right)\right| .$$

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

$$m_{\mathcal{H}_C} = \left|\cup_{l\in[p]} \mathrm{im}\left(h_l\right)\right| .$$

State selecting tensors are directly

Example 12 (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. 75. 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.

9 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.

9.1 Markov Logic Networks

Markov Logic Networks exploit the efficiency and interpretability of logical calculus as well as the expressivity of graphical models.

9.1.1 Markov Logic Networks as Exponential Families

We introduce Markov Logic Networks in the formalism of exponential families (see Sect. 4.7).

Definition 56 (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. 40).

Conversely, any binary feature ϕ_l of an exponential family defines a propositional formula (see Def. 40). Thus, any exponential family of distributions of $\bigtimes_{k\in[d]}[2]$, such that $\mathrm{im}\left(\phi_l\right) \subset \{0, 1\}$ 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 12 (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 12.*

9.1.2 Tensor Network Representation

Based on the previous discussion on the representation of exponential families by tensor networks in Sect. 4.7 we now derive a representation for Markov Logic Networks.

Theorem 53 (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 13 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 $\mathrm{im}\left(f_l\right) \subset [2]$ 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 13. □

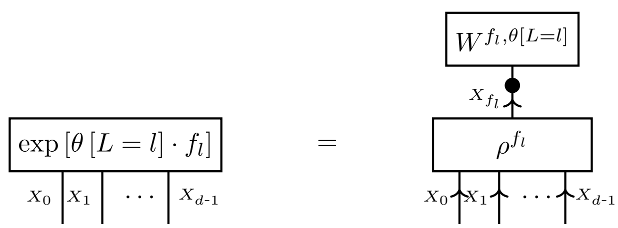


Figure 24: Factor of a Markov Logic Network to a formula f_l, represented as the contraction of a computation core ρ^{f_l} and an activation core $W^{f_l,\theta[L=l]}$. While the computation core ρ^{f_l} prepares based on basis calculus a categorical variable representing the value of the statistic formula 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. *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 $\mathcal{G}^{\mathcal{F}}$ $\{\exp\left[\theta\left[L=l\right] \cdot f_l\right] : l \in [p]\}$.

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 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.

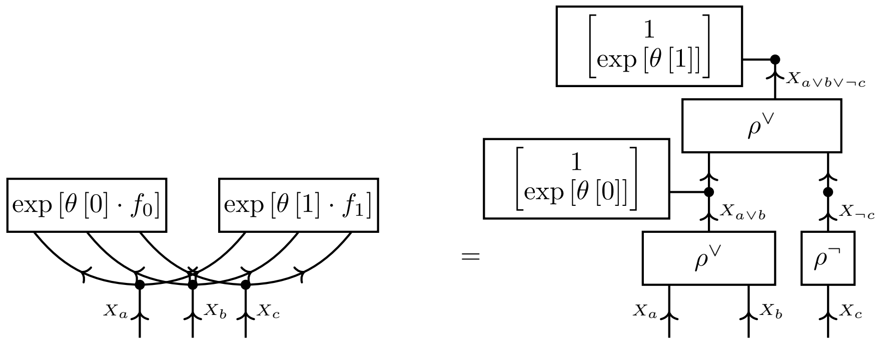


Figure 25: Example of a decomposed Markov Network representation of a Markov Logic Network with formulas $\{f_0 = a \vee b, f_1 = a \vee b \vee \neg c\}$. Since both formulas share the subformula $a \vee b$, their contracted factors have a representation by a connected tensor network.

9.1.3 Energy tensors

With the energy tensor

$$E^{\mathcal{F},\theta}\left[X_{[d]}\right] = \sum_{l\in[p]} \theta\left[L=l\right]\cdot f_l\left[X_{[d]}\right] = \left\langle \gamma^{\mathcal{F}}\left[X_{[d]},L\right],\theta\left[L\right]\right\rangle \left[X_{[d]}\right] \tag{23}$$

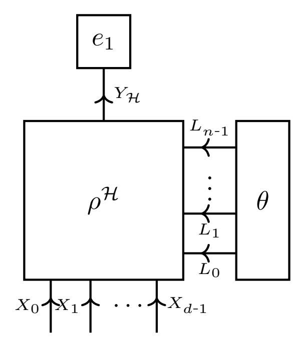
the MLN is the distribution

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \exp\left[E^{\mathcal{F},\theta}\right]\right\rangle \left[X_{[d]}|\varnothing\right] . \tag{24}$$

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:

$\sum_{l_{[n]}\in\times_{s\in[n]}[p_s]} \theta\left[L_{[n]}=l_{[n]}\right] f_{l_{[n]}}$ =



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 11) 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.

9.1.4 Expressivity

Based on Markov Logic Networks containing only maxterms and minterms (see Def. 42), 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 54. *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. 42)

$$\mathcal{F}_\wedge = \{Z^\wedge_{x_0,\ldots,x_{d-1}} : x_0,\ldots,x_{d-1} \in \bigtimes_{k\in[d]} [2]\}$$

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

$$\sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} \left(\ln \mathbb{P}\left[X_0 = x_0,\ldots,X_{d-1} = x_{d-1}\right]\right)\cdot Z^\wedge_{x_0,\ldots,x_{d-1}}[X_{[d]}] = \ln \mathbb{P}\left[X_{[d]}\right] .$$

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

$$\begin{aligned}
&\sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} \left(-\ln \mathbb{P}\left[X_0 = x_0,\ldots,X_{d-1} = x_{d-1}\right]\right)\cdot Z^\vee_{x_0,\ldots,x_{d-1}}[X_{[d]}] \\
&= \left(\sum_{\mathcal{V}_0\subset[d]} \left(-\ln \mathbb{P}\left[X_0 = x_0,\ldots,X_{d-1} = x_{d-1}\right]\right)\cdot \mathbb{I}\left[X_{[d]}\right]\right) \\
&\quad + \left(\sum_{\mathcal{V}_0\subset[d]} \left(\ln \mathbb{P}\left[X_0 = x_0,\ldots,X_{d-1} = x_{d-1}\right]\right)\cdot e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]\right) \\
&= \ln \mathbb{P}\left[X_{[d]}\right] + \lambda\cdot \mathbb{I}\left[X_{[d]}\right] ,
\end{aligned}$$

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 54 is the analogue in Markov Logic to Theorem 36, 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 13 (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?*

9.1.5 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 55. *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

$$\theta\left[L=k\right] = \ln\left[\frac{\langle\mathbb{P}\rangle\left[X_k=1\right]}{\langle\mathbb{P}\rangle\left[X_k=0\right]}\right]$$

Proof. By Theorem 6 we get a decomposition

$$\mathbb{P}\left[X_{[d]}\right] = \bigotimes_{k\in[d]} \mathbb{P}^k\left[X_k\right]$$

where

$$\mathbb{P}^k\left[X_k\right] = \langle\mathbb{P}\rangle\left[X_k\right] \,.$$

By assumption of positivity, the vector $\mathbb{P}^k\left[X_k\right]$ is positive for each $k\in[d]$ and the parameter

$$\theta\left[L=k\right] = \ln\left[\frac{\mathbb{P}^k\left[X_k=1\right]}{\mathbb{P}^k\left[X_k=0\right]}\right]$$

well-defined.

We then notice, that

$$\mathbb{P}^{(\{X_k\},\theta[L=k])}[X_k] = \mathbb{P}^k\left[X_k\right]$$

and therefore with the parameter vector of dimension $p=d$ defined as

$$\theta\left[L\right] = \sum_{k\in[d]} \theta\left[L=k\right]\cdot e_k\left[L\right]$$

we have

$$\begin{aligned}
\mathbb{P}^{(\{X_k\,:\,k\in[d]\},\theta)}[X_{[d]}] &= \bigotimes_{k\in[d]} \mathbb{P}^{(\{X_k\},\theta[L=k])}[X_k] \\
&= \bigotimes_{k\in[d]} \mathbb{P}^k\left[X_k\right] \\
&= \mathbb{P}\left[X_{[d]}\right] \,.
\end{aligned}$$

□

In Theorem 55 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. 9.2).

All atomic formulas can be selected by a single variable selecting tensor, that is

$$E^{(\{X_k\,:\,k\in[d]\},\theta)}[X_{[d]}] = \left\langle\mathcal{H}_V\left[X_{[d]},L\right],\theta\left[L\right]\right\rangle\left[X_{[d]}\right] \,.$$

In case of negative coordiantes $\theta\left[L=k\right]$ 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 $\theta\left[L=k\right]$ by $-\theta\left[L=k\right]$.

9.1.6 Boltzmann machines

A Boltzmann machine is a member of an exponential family with the energy tensor (see e.g. Chapter 43 in MacKay (2003))

$$E^{W,b}[X_0 = x_0, \dots, X_{d-1} = x_{d-1}] = \sum_{k,l\in[d]} W[L_{V,0} = k, L_{V,1} = l] \cdot x_k \cdot x_l + \sum_{k\in[d]} b[L_{V,0} = k] \cdot x_k \, .$$

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

$$E^{W,b}[X_{[d]}] = \left\langle \mathcal{H}_{\mathcal{A}}\left[X_{[d]}, L_C, L_{V,0}, L_{V,1}\right], \theta\left[L_C, L_{V,0}, L_{V,1}\right]\right\rangle \left[X_{[d]}\right] \, .$$

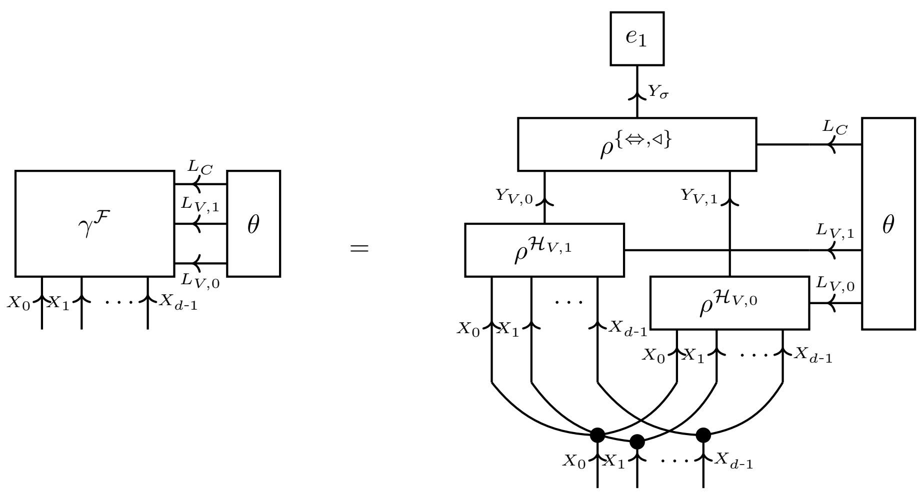


Figure 26: Tensor network representation of the energy of a Boltzmann machine

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.

9.2 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.

9.2.1 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 9. *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

$$\mathcal{Z}\left(\mathcal{F},\theta\right) = \left(\prod_{k\in[d]} m_k\right) - \langle f\rangle[\varnothing] + \langle f\rangle[\varnothing]\cdot\exp[\theta]$$

and therefore for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ with $f\left[X_{[d]} = x_{[d]}\right] = 1$

$$\begin{aligned}\langle \exp[\theta\cdot f]\rangle\left[X_{[d]} = x_{[d]}|\varnothing\right] &= \frac{\exp[\theta]}{\left(\prod_{k\in[d]} m_k\right) - \langle f\rangle[\varnothing] + \langle f\rangle[\varnothing]\cdot\exp[\theta]}\\ &\to \frac{1}{\langle f\rangle[\varnothing]} = \langle f\rangle\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}|\varnothing\right] \, .\end{aligned}$$

For any $x_0,\dots,x_{d-1} \in \bigtimes_{k\in[d]}[2]$ with $f\left[X_{[d]} = x_{[d]}\right] = 0$ we have on the other side

$$\begin{aligned}\langle \exp[\theta\cdot f]\rangle\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}|\varnothing\right] &= \frac{1}{\left(\prod_{k\in[d]} m_k\right) - \langle f\rangle[\varnothing] + \langle f\rangle[\varnothing]\cdot\exp[\theta]}\\ &\to 0 = \langle f\rangle\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}|\varnothing\right] \, .\end{aligned}$$

□

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 56. *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

$$\mathbb{P}^{(\mathcal{F},\beta\cdot\theta)}\left[X_{[d]}\right] \to \langle \mathcal{KB}\rangle\left[X_{[d]}|\right] \, .$$

Proof. Since $\mathcal{KB}$ is satisfiable we find $x_0,\dots,x_{d-1} \in \bigtimes_{k\in[d]}[2]$ with

$$\left\langle \exp\left[\sum_{f\in\mathcal{F}} \beta\cdot\theta_f\cdot f\right]\right\rangle\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}\right] = \exp\left[\beta\cdot\sum_{f\in\mathcal{F}}\theta_f\right]$$

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

$$\langle \mathcal{KB} \rangle \left[X_{[d]} | \varnothing\right] .$$

□

Remark 14 (More generic situation of simulated annealing)**.** *The process of taking $\beta \to \infty$ is known as simulated annealing, see Chapter 5. 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 36).*

9.2.2 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. 14.6). We use $\wedge$ symmetry to represent them as a contraction of the formulas building the Knowledge Base as conjunction.

Theorem 57 (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 15. *$\wedge$ 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.*

9.2.3 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 15 (see Def. 85).

Lemma 10. *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. 10 to show the following sparsity bound on the energy tensor of Markov Logic Networks.

Theorem 58. *Any formula f with a conjunctive normal form of n clauses satisfies*

$$\mathrm{rank}^d\left(f\right) \leq 2^n .$$

For any set $\mathcal{F}$ of formulas each with a conjunctive normal form of n_f clauses satisfies for any θ

$$\mathrm{rank}^d\left(\sum_{f\in\mathcal{F}}\theta_f\cdot f\right) \leq \sum_{f\in\mathcal{F}} 2^{n_f} .$$

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 114 and Lem. 10 to get

$$\mathrm{rank}^d\left(f\right) \leq \prod_{l\in[n]}\mathrm{rank}^d\left(Z^{\vee}_{\mathcal{V}_0^l,\mathcal{V}_1^l}\right) \leq 2^n .$$

Given a collection of formulas $\mathcal{F}$, each with a CNF of n_f clauses we apply Theorem 113 and get

$$\mathrm{rank}^d\left(\sum_{f\in\mathcal{F}}\theta_f\cdot f\right) \leq \sum_{f\in\mathcal{F}}\mathrm{rank}^d\left(f\right) \leq \sum_{f\in\mathcal{F}} 2^{n_f} .$$

□

9.3 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 Wainwright and Jordan (2008).

Definition 57. *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

$$\nu^{\mathcal{KB}}[X_{[d]}] = \left\langle\{f : f\in\mathcal{KB}\}\right\rangle\left[X_{[d]}\right] .$$

Given a set of formulas $\mathcal{F}$, we define the set of hybrid logic networks realizable with $\mathcal{F}$ 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 59. *If any only if $\bigwedge_{f\in\mathcal{KB}} f$ is satisfiable, the tensor*

$$\langle\{f : f\in\mathcal{KB}\}\cup\{\exp[\theta_f\cdot f] : f\in\mathcal{F}\}\rangle\left[X_{[d]}\right]$$

is normable.

Proof. We need to show that

$$\langle\{f : f\in\mathcal{KB}\}\cup\{\exp[\theta_f\cdot f] : f\in\mathcal{F}\}\rangle[\varnothing] > 0\,. \tag{25}$$

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

$$\begin{aligned}\langle\{f : f\in\mathcal{KB}\}\cup\{\exp[\theta_f\cdot f] : f\in\mathcal{F}\}\rangle\left[X_{[d]}=x_{[d]}\right] &= \left(\prod_{f\in\mathcal{KB}} f\left[X_{[d]}=x_{[d]}\right]\right)\cdot\left(\prod_{f\in\mathcal{F}}\exp[\theta_f\cdot f]\left[X_{[d]}=x_{[d]}\right]\right)\\ &= \left(\prod_{f\in\mathcal{F}}\exp[\theta_f\cdot f]\left[X_{[d]}=x_{[d]}\right]\right)\\ &> 0\,.\end{aligned}$$

Condition (25) follows from this and the Hybrid Logic Network is well-defined. □

9.3.1 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 60. *For any hybrid logic network we have*

$$\mathbb{P}^{(\mathcal{F},\theta,\mathcal{KB})}[X_{[d]}] = \left\langle\{\rho^f\left[Y_f,X_{[d]}\right] : f\in\mathcal{F}\cup\mathcal{KB}\}\cup\{e_1[Y_f] : f\in\mathcal{KB}\}\cup\{W^f[Y_f] : f\in\mathcal{F}\}\right\rangle\left[X_{[d]}|\varnothing\right]\,.$$

Proof. By Lem. 5. □

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 53)

$$W^f[Y_f] = \begin{bmatrix}1\\ \exp[\theta[L=l]]\end{bmatrix}[Y_l]$$

the hard formulas get activation cores by unit vectors

$$e_1[Y_f] = \begin{bmatrix}0\\1\end{bmatrix}[Y_l]\,.$$

As shown in Sect. 9.2.1, 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 27.

Remark 16. *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.*

9.3.2 Reasoning Properties

Deciding probabilistic entailment (see Def. 44) with respect to Hybrid Logic Networks can be reduced to the hard logic parts of the network.

Theorem 61. *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

$$\mathcal{KB}\models f\,.$$

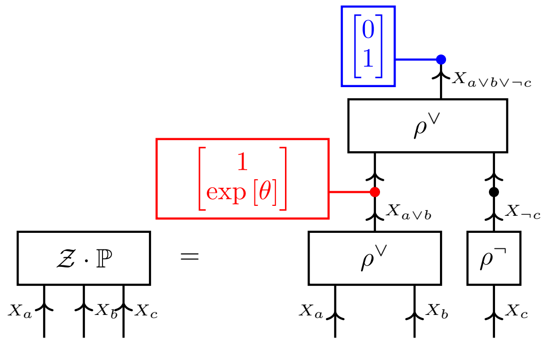


Figure 27: Diagram of a formula tensor with activated heads, containing hard constraint cores and probabilistic weight cores .

Proof. This follows from Theorem 45 on the representation of Hybrid Logic Networks as Markov Networks in Theorem 60. □

Formulas in $\mathcal{F}$, which are entailed or contradicted by $\mathcal{KB}$ are redundant, as we show next.

Theorem 62. *If for a formula f and $\mathcal{KB}$ we have*

$$\mathcal{KB} \models f \quad \text{or} \quad \mathcal{KB} \models \neg f$$

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 40. 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 7. By Theorem 61 only the hard logic parts of a Hybrid Logic Network are required in the ASK operation.

9.3.3 Expressivity

Hybrid Logic Networks extend the expressivity result of Theorem 54 to arbitrary probability tensors, dropping the positivity constraints for Markov Logic Networks.

Theorem 63. *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

$$\begin{aligned}\left\langle \nu^{\mathcal{KB}}, \exp\left[\left\langle \gamma^{\mathcal{F}_\wedge}, \theta\right\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]} = x_{[d]}\right] &= \prod_{l_{[d]}} \exp\left[\theta\left[L_{[d]} = l_{[d]}\right] \cdot Z^{\wedge}_{l_{[d]}}\left[X_{[d]} = x_{[d]}\right]\right] \\ &= \exp\left[\theta\left[L_{[d]} = x_{[d]}\right]\right] \\ &= \mathbb{P}\left[X_{[d]} = x_{[d]}\right] .\end{aligned}$$

□

9.4 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 54, 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 8 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 Russell and Norvig (2021)). Solving such problems is then equivalent to sampling from the worlds in a logical interpretation, and can be approached by the methods of Chapter 7. Among these classed, we have only discussed the Sudoku game in Example 7. Extensions by Hybrid Logic Networks can be interpreted as implementations of preferences among possible solutions by probabilities.

10 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 4. 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.

10.1 Mean parameters of Hybrid Logic Networks

The convex polytope of realizable mean parameters (see Def. 31) is for a statistic $\mathcal{F}$ of propositional formulas and a base measure ν

$$\mathcal{M}_{\mathcal{F},\nu} = \left\{\left\langle \mathbb{P}, \gamma^{\mathcal{F}}\right\rangle[L] \,:\, \mathbb{P} \in \Gamma^{\delta,\nu}\right\} ,$$

where by Γ we denote the set of all probability distributions. By The. 17 the convex polytope has a characterization as a convex hull

$$\mathcal{M}_{\mathcal{F},\nu} = \mathrm{conv}\left(\gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \in \bigtimes_{k\in[d]}[2],\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right) . \tag{26}$$

The mean parameter polytopes are examples of $0/1$-polytopes Ziegler (2000); Gillmann (2007), 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 $\mu\left[L = l\right] \in \{0,1\}$, then $\mu\left[L\right]$ 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

10.1.1 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 64. *Any parameter $\mu\left[L\right] \in \mathcal{M}_{\mathcal{F},\nu}$ is an extreme point of $\mathcal{M}_{\mathcal{F},\nu}$, if and only $\mu\left[L\right]$ 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

$$\mathcal{KB} = \{\neg^{(1-\mu[L=l])} f_l \,:\, l \in [p]\} ,$$

where we denote by $\neg^0$ the identity connective and by $\neg^1 = \neg$ the logical negation.

Proof. "$\Rightarrow$": Let μ be an extreme point of $\mathcal{M}_{\mathcal{F},\nu}$. Since by (26) $\mathcal{M}_{\mathcal{F},\nu}$ is a convex hull of vectors, there exists a $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ such that

$$\mu\left[L\right] = \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right] .$$

By definition of $\gamma^{\mathcal{F}}$, $\mu\left[L\right]$ is a boolean vector and for any $l \in [p]$ we have

$$f_l\left[X_{[d]} = x_{[d]}\right] = \mu\left[L = l\right]$$

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.

"$\Leftarrow$": To show the converse direction, let $\mu\left[L\right]$ 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 $l \in [p]$

$$\gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L = l\right] = f_l\left[X_{[d]} = x_{[d]}\right] = \mu\left[L = l\right]$$

and thus $\mu\left[L\right] = \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]$. With the characterization (26) this establishes in particular $\mu\left[L\right] \in \mathcal{M}_{\mathcal{F},\nu}$. Since $\mu\left[L\right]$ is boolean and therefore an extreme point of the cube $[0,1]^p$, it is also an extreme point of the subset $\mathcal{M}_{\mathcal{F},\nu} \subset [0,1]^p$. □

10.1.2 Mean parameters in the interior

By The. 26 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. *If $\mu\left[L\right] \in \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$, or equivalently the statistic is minimal and $\mu\left[L\right]$ is reproduceable by a distribution positive with respect to ν, then there is $\theta\left[L\right]$ such that $\mathbb{P}^{\mathcal{F},\theta,\nu}$ reproduces $\mu\left[L\right]$.*

10.1.3 Mean parameters outside the interior

By The. 19 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 65 (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. 41 probabilistic entailment $\mathbb{P} \models f_l$ (see Def. 44). 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. 65 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 13 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. 39), which has been shown in more generality in The. 22. For hybrid logic network we characterize the face base measures in the next theorem.

Theorem 66 (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

$$\operatorname{argmax}_{x_{[d]}} \left\langle \theta[L], \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]\right\rangle[\varnothing] .$$

For the first claim: We have since $\gamma^{\mathcal{F}}$ is boolean that

$$\|\theta[L]\|_1 \geq \max_{x_{[d]}} \left\langle \theta[L], \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right] \right\rangle [\varnothing] \ .$$

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 $\gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L = l\right] = \mathbb{I}_{>0}(\theta[L = l])$. 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 $\max_{x_{[d]}} \left\langle \theta[L], \gamma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right] \right\rangle [\varnothing]$ is attained, we have

$$\langle \theta, v \rangle [\varnothing] \in \max_{\mu} \langle \theta, \mu \rangle [\varnothing] \ .$$

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

$$\bigvee_{v[L]\,:\,\langle\theta,v\rangle[\varnothing]\in\max_{\mu}\langle\theta,\mu\rangle[\varnothing]} f_{\mathcal{F},\langle\theta,v\rangle[L]} \ .$$

□

10.1.4 Expressivity of Hybrid Logic Networks

Let us recall, that the set $\Lambda^{\mathcal{F},\mathrm{EL}}$ 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 $\Lambda^{\mathcal{F},\mathrm{EL}}$ (see Def. 57), that is

$$\mathcal{M}_{\mathcal{F},\nu}|_{\Lambda^{\mathcal{F},\mathrm{EL}}} = \left\{ \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle [L] \,:\, \mathbb{P} \in \Lambda^{\mathcal{F},\mathrm{EL}} \right\} \ .$$

While $\mathcal{M}_{\mathcal{F},\nu}|_{\Lambda^{\mathcal{F},\mathrm{EL}}} \subset \mathcal{M}_{\mathcal{F},\nu}$ 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 $\Lambda^{\mathcal{F},\mathrm{EL}}$. In the next example we provide a class of formulas, for which $\Lambda^{\mathcal{F},\mathrm{EL}}$ does not have sufficient expressivity.

Example 13 (Insufficient expressivity of $\Lambda^{\mathcal{F},\mathrm{EL}}$ 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.

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 67. *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. 67 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. 66. 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 10.1.4):

- $\mu[L=l] \in \mathcal{M}_{\mathcal{F},\nu}{}^{\circ}$: 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.

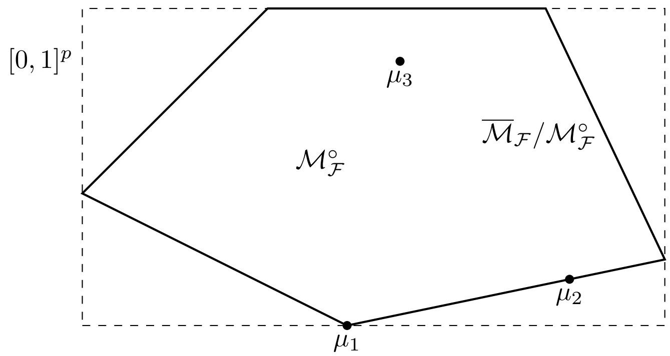


Figure 28: Sketch of the convex polytope $\mathcal{M}_{\mathcal{F},\nu}$ as a subset of the p-dimensional cube $[0,1]^p$ (here as a 2-dimensional projection) with example mean parameters μ_1, μ_2, μ_3. The boundary points $\mu_1, \mu_2 \in \overline{\mathcal{M}}_{\mathcal{F}}/\mathcal{M}^{\circ}_{\mathcal{F}}$ 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.

10.1.5 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.

10.1.6 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 Ziegler (2013))

Example 14 (Atomic formulas)**.** *The assumption of The. 67 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 15 (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!*

10.2 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 5 for the generic exponential families.

10.2.1 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

$$\mathrm{argmax}_{\theta[L]\in\mathbb{R}^p} \quad \langle \theta\left[L\right], \mu\left[L\right] \rangle \left[\varnothing\right] - A^{(\phi,\nu)}(\theta\left[L\right])$$

where

$$\mu\left[L\right] = \left\langle \gamma^{\mathcal{F}}, \mathbb{P} \right\rangle \left[L\right] \quad \text{and} \quad A^{(\phi,\nu)}(\theta\left[L\right]) = \left\langle \exp\left[\left\langle \gamma^{\mathcal{F}}\left[X_{[d]}, L\right], \theta\left[L\right] \right\rangle \left[X_{[d]}\right] \right], \nu \right\rangle \left[\varnothing\right] .$$

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 8. *Let $\mu\left[L\right] = \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle \left[L\right]$ and*

$$\tilde{\mathcal{F}} = \{f_l \,:\, \mu\left[L=l\right] \in \{0,1\}\} \quad , \quad \nu^{\tilde{\mathcal{F}},\mu}\left[X_{[d]}\right] = \bigwedge_{f_l\in\tilde{\mathcal{F}}} \neg^{(1-\mu[L=l])} f_l\left[X_{[d]}\right] .$$

If $\mu\left[L\right]$ 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 $\mathcal{F}$ then coincides with the projection of $\mathbb{P}$ onto $\Gamma^{\mathcal{F}/\tilde{\mathcal{F}},\nu^{\tilde{\mathcal{F}},\mu}}$.

10.2.2 Maximum Entropy in Hybrid Logic Networks

The Maximum Entropy Problem for Markov Logic Networks is

$$\mathrm{argmax}_{\mathbb{P}} \quad \mathbb{H}\left[\mathbb{P}\right] \quad \text{subject to} \quad \left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle \left[L\right] = \mu\left[L\right] \tag{27}$$

Corollary 9 (of The. 33)**.** *Let $\mathbb{P}^D$ be a distribution such that there is a positive distribution $\mathbb{P}$ with $\left\langle \mathbb{P}, \gamma^{\mathcal{F}} \right\rangle \left[L\right] = \left\langle \mathbb{P}^D, \gamma^{\mathcal{F}} \right\rangle \left[L\right]$. 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 $\mu\left[L=l\right] \in \{0,1\}$ 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 68. *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*

- $\mathcal{KB} = \{f_l : l \in [p], \mu\left[L=l\right] = 1\} \cup \{\neg f_l : l \in [p], \mu\left[L=l\right] = 0\}$
- $\mathcal{F} = \{f_l : l \in [p], \mu\left[L=l\right] \in (0,1)\}$
- *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 Wainwright and Jordan (2008)). $\square$

10.3 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. 3 in the following.

Lemma 11. *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 $\mu_D\left[L=l\right] \in (0,1)$ is then satisfied, if*

$$\theta[L=l] = \ln\left[\frac{\mu_D\left[L=l\right]}{(1-\mu_D\left[L=l\right])} \cdot \frac{T\left[X_{f_l}=0\right]}{T\left[X_{f_l}=1\right]} \right] \tag{28}$$

where by $T[X_{f_l}]$ we denote the contraction

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

Proof. Since $\mathrm{im}(f_l) \subset [2]$ we have

$$\mathrm{Id}|_{\mathrm{im}(f_l)} = e_1[X_{f_l}]$$

and the moment matching condition is by Lem. 3 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

$$\exp[\theta[L = l]] \cdot T[X_{f_l} = 1] = (\exp[\theta[L = l]] \cdot T[X_{f_l} = 1] + T[X_{f_l} = 0]) \cdot \mu_D[L = l] .$$

Rearranging the equations this is equal to

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

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 $T[X_{f_l} = 0], T[X_{f_l} = 1] \neq 0$ since Markov Logic networks are positive distributions and $f_l \notin \{\mathbb{I}[X_{[d]}], 0[X_{[d]}]\}$. □

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. 9.2 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 $T[X_{f_l} = 1] = 0$, respectively $T[X_{f_l} = 1] = 0$ 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. 9.2.

Since the likelihood is concave (see Koller and Friedman (2009)), 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 28 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 et al. (2008).

Remark 17 (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 Wainwright and Jordan (2008). Further, when activating both f and $\neg f$.

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
$$T[X_{f_l}] \leftarrow \left\langle \rho^{f_l} \right\rangle [X_{f_l}]$$
  Set
$$\theta[L = l] \leftarrow \ln \left[\frac{\mu_D[L = l]}{(1 - \mu_D[L = l])} \cdot \frac{T[X_{f_l} = 0]}{T[X_{f_l} = 1]} \right]$$
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
$$T[X_{f_l}] = \left\langle \{\rho^{f_l} \,:\, l \in [p]\} \cup \{W^{\tilde{l}} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\} \right\rangle [X_{f_l}]$$
    Set
$$\theta[L = l] = \ln \left[\frac{\mu_D[L = l]}{(1 - \mu_D[L = l])} \cdot \frac{T[X_{f_l} = 0]}{T[X_{f_l} = 1]} \right]$$
  end for
end while

10.4 Forward and backward mappings in closed form

We recall from Chapter 5, 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.

10.4.1 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. 9.1.4 how to represent any distribution as a MLN of maxterms (respectively minterms), see The. 54.

We use the tuple enumeration of the maxterms and minterms by $\bigtimes_{k\in[d]}[2]$ introduced in Sect. 6.3.3. 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 $\mu = \mathbb{I}\left[\right] - \mathbb{P}$. We can use these insights to provide a characterization of the forward and backward maps of the minterm and maxterm family.

Theorem 69. *Given the Markov Logic Networks to the formula sets*

$$\mathcal{F}_\wedge := \{Z^\wedge_{x_0,\ldots,x_{d-1}} : x_0,\ldots,x_{d-1} \in \bigtimes_{k\in[d]} [2]\} \quad \textit{and} \quad \mathcal{F}_\vee := \{Z^\vee_{x_0,\ldots,x_{d-1}} : x_0,\ldots,x_{d-1} \in \bigtimes_{k\in[d]} [2]\}$$

of all minterms, respectively of all mapterms, the forward mapping are

$$F^\wedge(\theta) = \langle \exp\left[\theta\right]\rangle \left[X_{[d]}|\varnothing\right] \quad \textit{and} \quad F^\vee(\theta) = \langle \exp\left[-\theta\right]\rangle \left[X_{[d]}|\varnothing\right] ,$$

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

$$B^\wedge(\mu) = \ln\left[\mu\right] \quad \textit{and} \quad B^{\mathcal{F}_\vee}(\mu) = -\ln\left[\mu\right] .$$

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

$$F^\wedge(\theta) = \left\langle \exp\left[\langle\{\mathcal{F}_\wedge,\theta\}\rangle\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right] = \langle \exp\left[\theta\right]\rangle \left[X_{[d]}|\varnothing\right] .$$

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

$$\begin{aligned} F^\vee(\theta) &= \left\langle \exp\left[\langle\{\mathcal{F}_\wedge,\theta\}\rangle\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right] \\ &= \left\langle \{\exp\left[\langle\{\mathbb{I},\theta\}\rangle\left[X_{[d]}\right]\right], \exp\left[-\langle\theta\rangle\left[X_{[d]}\right]\right]\}\right\rangle \left[X_{[d]}|\varnothing\right] \\ &= \langle \exp\left[-\theta\right]\rangle \left[X_{[d]}|\varnothing\right] \end{aligned}$$

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 $\mu \in \mathrm{im}\left(F^\vee\right)$ 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 $\theta = \ln\left[\mathbb{P}\right]$, respectively by maxterms when we choose $\theta = \mathbb{I} - \ln\left[\mathbb{P}\right]$. Thus, we have identified a subset of 2^d formulas, which is rich enough to fit any distribution.

10.4.2 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

$$\mu\left[L = k\right] = \mathbb{P}\left[X_k = 1\right] .$$

Theorem 70. *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

$$\begin{aligned} F^{[d]}(\theta[L])[L=k] &= \left\langle X_k, \mathbb{P}^{(\mathcal{F}_{[d]},\theta)}\left[X_{[d]}\right]\right\rangle[\varnothing] \\ &= \langle X_k, \langle \exp[\theta[L=k] \cdot X_k] \rangle [X_k|\varnothing] \rangle [\varnothing] \\ &= \frac{\exp[\theta[L=k]]}{1 + \exp[\theta[L=k]]} . \end{aligned}$$

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 15, 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 MacKay (2003).

Remark 18 (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.*

10.5 Constrained parameter estimation in the minterm family

We approach structure learning as constrained parameter estimation in the naive exponential family (see Sect. 4.7.4), 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. 54 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. 63.

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

$$\mathrm{argmax}_{\theta\in\bigotimes_{k\in[d]}\mathbb{R}^{m_k}} \left\langle \theta, \mathbb{P}^D\right\rangle[\varnothing] - A^{(\phi,\nu)}(\theta)$$

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. 63 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.

10.5.1 Parameter Estimation

Projecting onto the markov logic family to the statistic $\mathcal{F}$ is the instance of Problem **??** with the hypothesis choice

$$\Theta^{\mathcal{F}} = \operatorname{span}\left(\{f : f \in \mathcal{F}\}\right)\,.$$

Then, the problem is the parameter estimation problem studied in Sect. 10.2. 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 19 (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.

10.5.2 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 $f \in \mathcal{F}$.

10.6 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 8 and select from them a small subset.

10.6.1 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

$$\operatorname{argmin}_{f\in\mathcal{H}}\operatorname{argmin}_{\theta\in\mathbb{R}^{|\mathcal{F}|+1}}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F}\cup\{f\},\theta,\nu)}\right] . \tag{$\mathrm{P}_{D,\mathcal{F},\mathcal{H}}$}$$

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 Koller and Friedman (2009)).

10.6.2 Gain Heuristic

In the gain heuristic, only the parameters of the new formula are optimized and the others left unchanged. This amounts to

$$\operatorname{argmin}_{f\in\mathcal{H}}\left(\min_{\theta[|\mathcal{F}|]\in\mathbb{R}}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F}\cup\{f\},\theta,\nu)}\right]\right) . \tag{$\mathrm{P}^{\mathrm{gain}}_{D,\mathcal{F},\mathcal{H}}$}$$

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 12. *The gain heuristic objective is an upper bound on the true greedy objective.*

Proof. Since

$$\operatorname{argmin}_{f\in\mathcal{H}}\left(\operatorname{argmin}_{\theta\in\mathbb{R}^{|\mathcal{F}|+1}}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F}\cup\{f\},\theta,\nu)}\right]\right) \leq \operatorname{argmin}_{f\in\mathcal{H}}\left(\operatorname{argmin}_{\theta[|\mathcal{F}|]\in\mathbb{R}}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F}\cup\{f\},\theta,\nu)}\right]\right) .$$

□

Further, this is Problem ($\mathrm{P}_{\Theta,\mathbb{P}^D}$) in the case

$$\Theta = \ln\left[\tilde{\mathbb{P}}\right] + \cup_{f\in\mathcal{F}}\operatorname{span}(f) .$$

Let us choose a formula $f \in \mathcal{F}$ and consider Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ in the case

$$\Theta^f = \ln\left[\tilde{\mathbb{P}}\right] + \operatorname{span}(f) .$$

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 5 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 71. *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

$$\hat{f} \in \operatorname{argmax}_{f\in\mathcal{F}}\mathrm{D}_{\mathrm{KL}}\left[\left\langle\mathbb{P}^D, f\right\rangle[\varnothing] \,||\, \left\langle\tilde{\mathbb{P}}, f\right\rangle[\varnothing]\right]$$

and $\theta_{\hat{f}}$ is the weight of $\hat{f}$ in the solution of Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ with $\Gamma = \tilde{\mathbb{P}} + \operatorname{span}(f)$. 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. 11 taken at

$$\theta_f = \ln\left[\frac{\mu_D}{(1-\mu_D)} \cdot \frac{(1-\tilde{\mu})}{\tilde{\mu}}\right]$$

where

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

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 60 and get

$$\begin{aligned}
\left\langle \tilde{\mathbb{P}}, \exp\left[\theta_f \cdot f\right] \right\rangle [\varnothing] &= \left\langle \tilde{\mathbb{P}}, \rho^f\left[X_f\right], W^f\left[X_f\right] \right\rangle [\varnothing] \\
&= (1-\tilde{\mu}) + \tilde{\mu} \cdot \exp\left[\theta_f\right] \\
&= (1-\tilde{\mu}) + \frac{\mu_D \cdot (1-\tilde{\mu})}{(1-\mu_D)} \\
&= (1-\tilde{\mu}) \cdot \frac{1}{(1-\mu_D)} \, .
\end{aligned}$$

It follows, that

$$\begin{aligned}
A^{\mathcal{F}\cup\{f\},\nu}\left(\tilde{\theta}\cup\{\theta_f\}\right) &= \ln\left[\left\langle \tilde{\mathbb{P}}, \exp\left[\theta_f \cdot f\right] \right\rangle [\varnothing]\right] \\
&= \ln\left[1-\tilde{\mu}\right] - \ln\left[1-\mu_D\right] \, .
\end{aligned}$$

We further have

$$\mu_D \cdot \theta_f = \mu_D \cdot \left[\ln\left[\frac{\mu_D}{(1-\mu_D)} \cdot \frac{(1-\tilde{\mu})}{\tilde{\mu}}\right]\right] = \mu_D \ln\left[\mu_D\right] - \mu_D \ln\left[1-\mu_D\right] + \mu_D \ln\left[1-\tilde{\mu}\right] - \mu_D \ln\left[\tilde{\mu}\right]$$

and arrive at

$$\begin{aligned}
\mathbb{H}\left[\mathbb{P}^D, \tilde{\mathbb{P}}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(f,\theta_f,\tilde{\mathbb{P}})}\right] &= \mu_D \ln\left[\mu_D\right] - \mu_D \ln\left[1-\mu_D\right] + \mu_D \ln\left[1-\tilde{\mu}\right] - \mu_D \ln\left[\tilde{\mu}\right] - \ln\left[1-\tilde{\mu}\right] - \ln\left[1-\mu_D\right] \\
&= \left(-\mu_D \ln\left[\tilde{\mu}\right] - (1-\mu_D)\ln\left[1-\tilde{\mu}\right]\right) - \left(-\mu_D \ln\left[\mu_D\right] - (1-\mu_D)\ln\left[1-\mu_D\right]\right) \, .
\end{aligned}$$

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 $\Theta = \mathrm{span}\left(f\right)$ is thus given by $\mathrm{D}_{\mathrm{KL}}\left[\left\langle \mathbb{P}^D, f \right\rangle [\varnothing] \,||\, \left\langle \tilde{\mathbb{P}}, f \right\rangle [\varnothing]\right]$, we have that Problem **??** in the case $\Theta = \bigcup_{f\in\mathcal{F}} \mathrm{span}\left(f\right)$ is solved at $\hat{\theta} = \theta_{\hat{f}} \cdot \hat{f}$ where

$$\hat{f} = \mathrm{D}_{\mathrm{KL}}\left[\left\langle \mathbb{P}^D, f \right\rangle [\varnothing] \,||\, \left\langle \tilde{\mathbb{P}}, f \right\rangle [\varnothing]\right] \, .$$

□

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.

10.6.3 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

$$\begin{aligned}\nabla_\theta|_0 \mathcal{L}_D\left(\mathbb{P}^{(\delta,f(\theta),\nu)}\right) &= \left\langle \nabla_{f(\theta)}|_{\tilde{\theta}} \mathcal{L}_D\left(\mathbb{P}^{(\delta,f(\theta),\nu)}\right), \nabla_\theta|_0 f(\theta) \right\rangle [\varnothing] \\ &= \left\langle \mathbb{P}^D, \gamma^\phi \right\rangle [L] - \left\langle \mathbb{P}^{(\delta,\tilde{\theta},\nu)}, \gamma^\phi \right\rangle [L] .\end{aligned}$$

We want to choose the formula, which is best aligned with the gradient of the log-likelihood, that is using a formula selecting map $\mathcal{H}$

$$\operatorname{argmax}_{l\in[p]} \left\langle \mathbb{P}^D, \mathcal{H} \right\rangle [L = l] - \left\langle \mathbb{P}^{(\delta,\tilde{\theta},\nu)}, \mathcal{H} \right\rangle [L = l] . \tag{$\mathrm{P}^{\mathrm{grad}}_{D,\mathcal{F},\mathcal{H}}$}$$

This method is known as the gradient heuristic or grafting. The objective of Problem ($\mathrm{P}^{\mathrm{grad}}_{D,\mathcal{F},\mathcal{H}}$) 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 ($\mathrm{P}^{\mathrm{grad}}_{D,\mathcal{F},\mathcal{H}}$) is further equivalent to the formula alignment

$$\operatorname{argmax}_{f\in\mathcal{H}} \left\langle f, \mathbb{P}^D - \tilde{\mathbb{P}} \right\rangle [\varnothing] .$$

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.

10.6.4 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.

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 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.*

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

10.7 Proposal distribution

Let us now understand the likelihood gradient as the energy tensor of a probability distribution, which we call the proposal distribution.

Definition 58 (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 mode query, for which we can apply the methods introduced in Chapter 5, such as Gibbs Sampling or Mean Field Approximations in combination with annealing.

10.7.1 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 Ziegler (2000); Gillmann (2007).

The extreme points are the formulas selectable by the formula selecting map $\mathcal{H}$.

10.8 Discussion

Remark 21 (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*
$$\mathrm{argmax}_{\mathcal{F},\theta} \mathbb{P}^{\mathcal{F},\theta}[\{D(j)\}_{j\in[m]}] \cdot \mathbb{P}[\mathcal{F}, \theta] \ . \tag{29}$$

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.

11 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 5 and Chapter 10 are robust against such effects.

11.1 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.

11.1.1 Fluctuation of the empirical distribution

When drawing random states $D(j)\in\times_{k\in[d]}[m_k]$ 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

$$\mathbb{E}\left[e_{D(j)}\right]=\sum_{x_{[d]}\in\times_{k\in[d]}[m_k]}\mathbb{P}^*\left[X_{[d]}=x_{[d]}\right]e_{x_{[d]}}\left[X_{[d]}\right]=\mathbb{P}^*\left[X_{[d]}\right] .$$

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 $\times_{k\in[d]}[m_k]$. The forwarding of $\times_{k\in[d]}[m_k]$ under the one-hot encoding is a multinomial random variable, see The. 77.

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 $x\in\times_{k\in[d]}[m_k]$ 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.

11.1.2 Mean parameter of the empirical distribution

We now investigate the empirical mean parameter

$$\mu_D\left[L\right]=\left\langle\gamma^\phi\left[X_{[d]},L\right],\mathbb{P}^D\left[X_{[d]}\right]\right\rangle\left[L\right] .$$

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 72. *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.

11.1.3 Noise tensor and its width

Motivated by The. 72, 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 59. *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

$$\mu_D - \mathbb{E}[\mu_D] = \left\langle \gamma^\phi, \mathbb{P}^D - \mathbb{P}^* \right\rangle [L] \, .$$

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 60. *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 Goeßmann (2021) and generic probabilistic bounds will be provided in Sect. 11.4.

11.2 Error bounds based on the noise width

We now derive error bounds for parameter estimation and structure learning, as introduced in Chapter 10. When combined with probabilistic bounds on the noise width, they are probabilistic success guarantees.

11.2.1 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 5 we have characterized those by the backward map acting on the mean parameter. Thus, while we are interested in the expected canonical parameter

$$\theta_* [L] = B^{(\phi,\nu)}(\mu^*[L])$$

we get an estimation by the empirical canonical parameter

$$\theta_D [L] = B^{(\phi,\nu)}(\mu_D [L]) .$$

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 Shalev-Schwartz, Shai and Ben-David, Shai (2014) Since the empirical risk has a linear dependence on μ_D, we have at each θ

$$\begin{aligned}
\mathbb{E}\left[\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\phi,\theta,\nu)}\right]\right] &= \mathbb{E}\left[\langle \mu_D, \theta \rangle [\varnothing] - A^{(\phi,\nu)}(\theta)\right] \\
&= \langle \mathbb{E}[\mu_D], \theta \rangle [\varnothing] - A^{(\phi,\nu)}(\theta) \\
&= \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\phi,\theta,\nu)}\right]
\end{aligned}$$

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 13. *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

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

It follows that

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

and the claim follows from comparison with Def. 59 and Def. 60. $\square$

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 73. *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*

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\phi,\theta_*,\nu)}||\mathbb{P}^{(\phi,\theta_D,\nu)}\right] \leq 2\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right) . \tag{30}$$

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. 13 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

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\phi,\theta_*,\nu)}||\mathbb{P}^{(\phi,\theta_D,\nu)}\right] \leq 2\omega_\Gamma\left(\eta^{\phi,\mathbb{P}^*,D}\right)\square$$

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 74. *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. 13 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. □

11.2.2 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 61. *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 75. *Whenever*

$$\Delta\left(\mu^{*}\right) > 2 \cdot \omega_{\left\{e_{x_{[d]}} : x_{[d]} \in \times_{k \in [d]}[m_{k}]\right\}}\left(\eta^{\phi, \mathbb{P}^{*}, D}\right) ,$$

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 $l^{D} \in \operatorname{argmax}_{l \in [p]} \mu_{D}\left[L = l\right]$ 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

$$\mu^{*}[L = l^{D}] \leq \mu^{*}[L = l^{*}] - \Delta\left(\mu^{*}\right)$$

and

$$\mu_{D}\left[L = l^{D}\right] \geq \mu_{D}\left[L = l^{*}\right] .$$

Comparing both intequalities we get

$$\left(\mu_{D}\left[L = l^{D}\right] - \mu^{*}[L = l^{D}]\right) + \left(-\mu_{D}\left[L = l^{*}\right] + \mu^{*}[L = l^{*}]\right) \geq \Delta\left(\mu^{*}\right) .$$

Estimating the terms in the bracket by the width of the noise tensor with respect to basis vectors, we get

$$2 \cdot \omega_{\left\{e_{x_{[d]}} : x_{[d]} \in \times_{k \in [d]}[m_{k}]\right\}}\left(\eta^{\phi, \mathbb{P}^{*}, D}\right) \geq \Delta\left(\mu^{*}\right) ,$$

which is a contradiction to the assumption. Thus, any mode of the empirical mean parameter is also a model of the expected mean parameter. □

11.3 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 76. *For any $\mathcal{F}$ the marginal distribution of the coordinate $\eta^{\mathcal{F}, \mathbb{P}^{*}, D}\left[L = l\right]$ 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

$$\mu\left[L = l\right] = \mathbb{E}\left[\left\langle \phi_{l}, e_{D(j)}\left[X_{[d]}\right] \right\rangle [\varnothing]\right] = \left\langle \phi_{l}, \mathbb{P}^{*} \right\rangle [\varnothing]$$ □

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.

11.3.1 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 $\eta^{\mathcal{F},\mathbb{P}^*,D}$.

11.3.2 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 77. *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. 76, which holds for generic statistics $\mathcal{F}$.

11.3.3 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 78. *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. 75 with the width bound of The. 80, 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. 75. $\square$

Example 16 (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*

$$\Delta\left(E^{(\mathcal{F},\mathbb{P}^{(\{f\},\theta)}-\langle\mathbb{I}\rangle\left[X_{[d]}|\varnothing\right])}\right) = \frac{1}{2^d - 1 + \exp\left[-\theta\right]}$$

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.

11.3.4 Guarantees for Unconstrained Parameter Estimation

We here the sphere bounds and combine with The. 74.

Theorem 79. *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. 74 with the probabilistic width bound The. 81 to be shown in the next section. $\square$

11.4 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. 76. 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.

11.4.1 Basis Vectors

We first introduce the sub-Gaussian Norm and show how we can exploit it to state concentration inequalities.

Definition 62 (Sub-Gaussian Norm, see Def. 2.5.6 in Vershynin (2018))**.** *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

$$\|X\|_{\psi_2} = \inf\left\{C > 0\,:\,\left\langle\mathbb{P}\left[X_{[d]}\right], \exp\left[\frac{1}{C^2}\cdot T\left[X_{[d]}\right]\right]\right\rangle\left[\varnothing\right] \leq 2\right\}\,.$$

We now show a sub-Gaussian norm bound on the coordinates of the noise tensor.

Lemma 14. *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 Vershynin (2018), 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 80. *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 Vershynin (2018)) 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. 14 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. 80 is furthermore sharp, see the construction of an identically scaling lower bound in Exercise 2.5.11 in Vershynin (2018). Note that the binomials used here tend to normal distributed variables used in the construction therein.

11.4.2 Sphere

For any tensor $\eta\left[L\right]$ and the sphere $\mathcal{S} \subset \mathbb{R}^p$ we have

$$\omega_{\mathcal{S}}\left(\eta\left[L\right]\right) = \left\|\eta\left[L\right]\right\|_2 \,.$$

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 81. *Let $\mu\left[L\right]$ be a deterministic vector with coordinates in $[0,1]$ and $\eta\left[L\right]$ a random vector, which coordiantes are for $l \in [p]$ marginally distributed as*

$$\eta\left[L=l\right] \sim B\left(m, \mu\left[L=l\right]\right) \,.$$

Then we have for any $\epsilon > 0$, $\tau > 0$ and $m \in \mathbb{N}$ with probability at least $1-\epsilon$

$$\left\|\frac{\eta - \mathbb{E}\left[\eta\right]}{m}\right\|_2 \leq \tau$$

provided that

$$m \geq \frac{\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right]}{\epsilon \cdot \tau^2} \,.$$

Proof. Since the squared norm of the noise is the sum of squared centered and averaged Binomials, we have

$$\mathbb{E}\left[\left\|\eta\left[L\right] - \mathbb{E}\left[\eta\left[L\right]\right]\right\|_2^2\right] = m \cdot \left(\sum_{l\in[p]} \mu\left[L=l\right]\left(1-\mu\left[L=l\right]\right)\right)$$

Here we used that the variance of a variable distributed by $B\left(m, \mu\left[L=l\right]\right)$ is $m \cdot \mu\left[L=l\right]\left(1-\mu\left[L=l\right]\right)$.

If follows, that

$$\mathbb{E}\left[\left(\left\|\frac{\eta - \mathbb{E}\left[\eta\right]}{m}\right\|_2\right)^2\right] = \frac{\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right]}{m} \,.$$

Then we apply a Chebyshev Bound to get for any $\tau > 0$

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}\left[\eta\right]}{m}\right\|_2 > \tau\right] = \mathbb{P}\left[\left(\left\|\frac{\eta - \mathbb{E}\left[\eta\right]}{m}\right\|_2\right)^2 > \tau^2\right] \leq \frac{\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right]}{m \cdot \tau^2} \tag{31}$$

For a $\epsilon > 0$ we choose any m with

$$m \geq \frac{\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right]}{\tau^2 \epsilon}$$

and get

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}\left[\eta\right]}{m}\right\|_2 > \tau\right] \leq \epsilon \,. \tag{32}$$

Thus, we have

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}\left[\eta\right]}{m}\right\|_2 \leq \tau\right] = 1 - \mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}\left[\eta\right]}{m}\right\|_2 > \tau\right] \geq 1 - \epsilon \,. \tag{33}$$

□

For the minterm family where $\mathcal{F} = \delta$ the noise tensor is a rescaled and centered multinomial. In that case, the bound of The. 81 can be simplified by

$$\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right] = 1 - \left\langle \mu\left[L\right]^2 \right\rangle\left[\varnothing\right] \,.$$

11.5 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 Talagrand (2014). While we showed bounds based on the sub-Gaussian norm, more general sub-exponential bounds could be used, see Wainwright (2019).

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 Goeßmann (2021).

12 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.

12.1 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 14 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 63 (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*

$$x_W[L, O_{[n]}] : [d] \times \left(\bigtimes_{l\in[n]} [r] \right) \to [2] . \tag{34}$$

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 $I^{-1}(o_0), \ldots, I^{-1}(o_{n-1})$.

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 $e_r[O]$, where we add an auxiliary object I_r 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.

12.1.1 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 6) is a special case of first-order logics, namely when demanding $n = 0$. Consistent with Def. 63 we have a propositional logic world by

$$x_W : [d] \to [2] ,$$

which we have in Chapter 6 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

$$e_{x_W}\left[X_{[d]}\right] = \bigotimes_{k\in[d]} e_{x_W[L=k]}\left[X_k\right] .$$

12.1.2 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 15 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 Russell and Norvig (2021)), 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.

12.1.3 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 13. 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.

12.1.4 Semantics of formulas

Following the development of Chapter 6, 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. 75) 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.

12.1.5 Two levels of tensor representation

In comparison with propositional logics, first-order logic bears two levels of natural tensor representations. In the first level, which we call the structured level, each world (see Def. 63) 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. 16). We call this the factored level, since it arises in general in the discussion of factored representations. As argued above, the assumption of database semantics is central to exploit the tensor structure of the substitution 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 substitution level.

12.2 Formulas in a fixed first-order logic world

Following the argumentation above, we in this section restrict to the exploitation of tensors in the structured level, namely a fixed world represented as a tensor $x_W[L, O_{[n]}]$, see Def. 63. 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.

12.2.1 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 6.

Definition 64 (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*

$$q|_{x_W}[O_q] : \bigtimes_{l\in[O_q]} [r] \to \{0,1\} .$$

Each coordinate represents thereby the boolean, whether the substitution of the variables in the formula is satisfied given a world x_W, that is

$$q|_{x_W}[O_q = o_q] = \begin{cases} 1 & \text{if the substitution of } q \text{ with the variables } O_q \text{ replaced by the objects } I(o_l) \text{ is satisfied on the world } x_W. \\ 0 & \text{else} \end{cases} .$$

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 factored level of tensor interpretation, namely representation of all possible worlds.

12.2.2 Atomic Formulas

Atomic formulas in first-order logic are predicates, which are applied on terms.We restrict in this chapter to function-free logic, therefore terms are either constants or variables. If all arguments of a predicate are assigned by free variables, the corresponding grounding tensor is stored in the slices to the first axis of x_W and we have

$$g_k|_{x_W} = \left\langle x_W[L, O_{[n]}], e_k\left[L\right]\right\rangle \left[O_{[n]}\right] . \tag{35}$$

In contrast, when a constant object I_o is assigned to an argument of a predicate, the grounding tensor reduced to a slice of the grounding with exclusively free variables. We capture such slicings by contractions with one-hot encodings of the corresponding constant.

We formalize this approach by atom creating tensors, which contraction with the world tensor results in the grounding of the corresponding atomic formula.

Definition 65. *Let there be an atomic formula q, which is constructed using the l-th predicate and has constants assigned on the arguments $\mathcal{U}^C \subset [n]$ and free variables to the arguments $\mathcal{U}^V = [n]/\mathcal{U}^C$. Let the constant map $C : \mathcal{U}^C \subset [n] \to [r]$ map to the specific objects represented by the constant and $V : \mathcal{U}^V \subset [n] \to \mathcal{V}$ to free variables labeled by a set $\mathcal{V}$. Then the atom creating tensor to q is*

$$\psi_q\left[O_{V(\mathcal{U}^V)}\right] = e_l\left[L\right] \otimes \left(\bigotimes_{l\in\mathcal{U}^C} e_{C(l)}\left[O_l\right]\right) \otimes \left(\bigotimes_{l\in\mathcal{U}^V} \delta\left[O_{V(l)}, O_l\right]\right) .$$

The ground of the atom is then the contraction of the atom creating tensor with the world tensor, that is

$$q|_{x_W}\left[O_{V(\mathcal{U}^V)}\right] = \left\langle x_W[L, O_{[n]}], \psi_q\left[O_{\mathcal{V}}\right]\right\rangle \left[O_{V(\mathcal{U}^V)}\right] .$$

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. 12.3.2.

12.2.3 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 82. *For any connective $\circ$ and formulas q_1 and q_2 we have*

$$(q_1 \circ q_2)|_{x_W}\left[O_{q_1} \cup O_{q_2}\right] = \left\langle \rho^{q_1|_{x_W}}\left[Y_{q_1}, O_{q_1}\right], \rho^{q_2|_{x_W}}\left[Y_{q_2}, O_{q_2}\right], \rho^{\circ}\left[Y_{q_1\circ q_2}, Y_{q_1}, Y_{q_2}\right], e_1\left[Y_{q_1\circ q_2}\right]\right\rangle \left[O_{[n]}\right] . \tag{36}$$

Proof. This directly follows from The. 99. □

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.

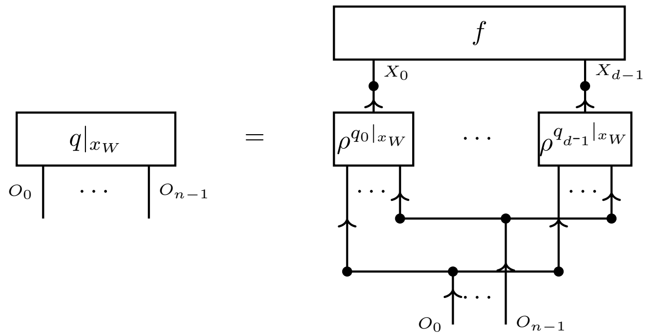
When interpreting the head variables of relational encoded atomic formulas as the atoms of a propositional theory, we find a propositional formula f associated with any decomposable first order logic formula.

Definition 66. *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:



12.2.4 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 67. *Given a grounding tensor*

$$q|_{x_W}\left[O_0,\dots,O_{n-1}\right]$$

the existential and universal quantification with respect to the first variable are the tensors

$$(\exists_{o_0} q)|_{x_W}\left[O_1,\dots,O_{n-1}\right] \quad \textit{and} \quad (\forall_{o_0} q)|_{x_W}\left[O_1,\dots,O_{n-1}\right]$$

with coordinates as follows. For an assignment $o_1,\dots,o$ to the non-quantified variables we have

$$(\exists_{o_0} q)|_{x_W}\left[O_1=o_1,\dots,O_{n-1}=o_{n-1}\right]=1$$

if and only if there is an assignment $o_0\in[r_0]$ such that

$$q|_{x_W}\left[O_0=o_0,O_1=o_1,\dots,O_{n-1}=o_{n-1}\right]=1\,.$$

Conversely, we have for the universal quantification that

$$(\forall_{o_0} q)|_{x_W}\left[O_1=o_1,\dots,O_{n-1}=o_{n-1}\right]=1$$

if and only if for any assignment $o_0\in[r_0]$ we have

$$q|_{x_W}\left[O_0=o_0,O_1=o_1,\dots,O_{n-1}=o_{n-1}\right]=1\,.$$

Let us now show, that existential and universal quantification are coordinatewise transforms (see Def. 73) 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

$$\mathbb{I}_{>\cdot}\mathbb{R}\rightarrow\{0,1\}\quad,\quad \mathbb{I}_{>z}\left(x\right)=\begin{cases}1 & \text{if} \quad x>z\\ 0 & else\end{cases}\,.$$

Theorem 83. *For any formula q with variables $O_{[n]}$ we have*

$$(\exists o_0 q)|_{x_W}\left[O_1,\dots,O_{n-1}\right]=\mathbb{I}_{>0}\left(\left\langle q|_{x_W}\right\rangle\left[O_1,\dots,O_{n-1}\right]\right)\left[O_1,\dots,O_{n-1}\right]$$

and

$$(\forall o_0 q)|_{x_W}\left[O_1,\dots,O_{n-1}\right]=\mathbb{I}_{>r-1}\left(\left\langle q|_{x_W}\right\rangle\left[O_1,\dots,O_{n-1}\right]\right)\left[O_1,\dots,O_{n-1}\right]$$

Proof. We proof the claimed equalities to arbitrary slices of the remaining variables, which amount to arbitrary substitutions of the formulas. For any indices $o_1\in[r_1],\dots,o_{n-1}\in[r_{n-1}]$ we notice, that

$$\begin{aligned}\left\langle q|_{x_W}\right\rangle\left[O_1=o_1,\dots,O_{n-1}=o_{n-1}\right] &= \sum_{o_0\in[r_0]} q|_{x_W}\left[O_0=o_0,\dots,O_{n-1}=o_{n-1}\right]\\ &= \left|o_0\in[r_0]\,:\,q|_{x_W}\left[O_0=o_0,\dots,O_{n-1}=o_{n-1}\right]=1\right|\,.\end{aligned}$$

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

$$(\exists_{o_0} q)|_{x_W} [O_1 = o_1, \dots, O_{n-1} = o_{n-1}] = \mathbb{I}_{>0} \left(\langle q|_{x_W} \rangle [O_1, \dots, O_{n-1}] \right) [O_1 = o_1, \dots, O_{n-1} = o_{n-1}] .$$

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

$$(\forall o_0 q)|_{x_W} [O_1 = o_1, \dots, O_{n-1} = o_{n-1}] = \mathbb{I}_{>r-1} \left(\langle q|_{x_W} \rangle [O_1, \dots, O_{n-1}] \right) [O_1 = o_1, \dots, O_{n-1} = o_{n-1}] .$$

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 (2011) and Sect. 12.3).

As will be discussed in Chapter 14, 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. 100). 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.

To summarize, let us assume a formula is in its prenex normal form, that is a collection of quantifiers are acting on a qantifier free part. We can represent its grounding tensor by

- Instantiations of the tom groundings with the assigned variables, as contractions of the relational encoding of the world tensor with atom selecting tensors.
- Propositional formula acting on the head variables of the predicate instantiations, representing the connectives combining the formula.
- Quantifiers as a composition of contractions closing the quantified variable and coordinatewise transforms with the respective greater-than indicators.

12.2.5 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 15) satisfies

$$\ell_0 \left(q|_{x_W} \right) << r^{|O_q|} .$$

In this case, since most coordinates vanish, the basis CP decomposition (see Sect. 15.1.2) enables a representation of the grounding with significantly lower storage demand, see The. 108. 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. 29.

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.

12.2.6 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. 15.1.2) of $p|_{x_W}$. We can understand these solution mappings as stored in the leg-matrices $V^{q,l}$ (see Figure **??**).

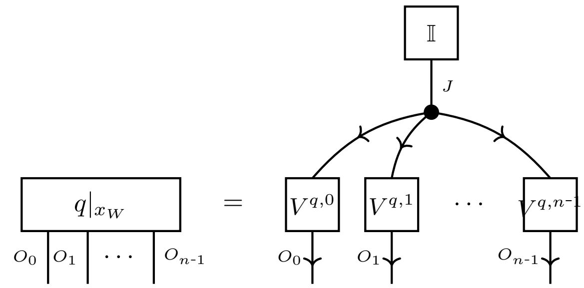


Figure 29: Basis CP Decomposition of the grounding of q, following the scheme of The. 108. Instead of direct storage of the grounding tensor $q|_{x_W}$, the non-zero coordinates are enumerated by a variable J and the corresponding coordinates stored in leg-matrices $V^{q,l}$.

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.

Definition 68 (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*

$$\mathrm{JOIN}\,(q_0,\ldots,q_{p-1})|_{x_W}\left[\bigcup_{l\in[p]} O_{\mathcal{V}^l}\right] = \langle q_l|_{x_W}[O_{\mathcal{V}^l}] \,:\, l\in[p]\rangle\left[\bigcup_{l\in[p]} O_{\mathcal{V}^l}\right]\,.$$

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. 73) 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 7). 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.

12.3 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 (2011).

12.3.1 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

$$\mathrm{KG}|_{x_W}[L,O_0,O_1] : [d]\times[r]\times[r] \to \{0,1\}\,.$$

12.3.2 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 $\mathrm{rdftype}$ 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

$$\mathrm{RDF}|_{x_W} : [r] \times [r] \times [r] \to \{0, 1\}$$

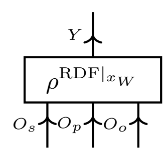
where

$$\mathrm{RDF}|_{x_W}[O_s = o_s, O_p = o_p, O_o = o_o] = \begin{cases} \mathrm{KG}|_{x_W}[L = o_s, O_0 = o_o, O_1 = 0] & \text{if} \quad o_p = I^{-1}(\mathrm{rdftype}) \\ \mathrm{KG}|_{x_W}[L = o_p, O_0 = o_s, O_1 = o_o] & \text{if} \quad o_p = I^{-1}(g_k) \quad \text{for some} \quad k \\ 0 & \text{else} \end{cases} .$$

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 $\mathrm{rdftype}$ 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 $\mathrm{rdftype}$ 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 $\mathrm{RDF}|_{x_W}[O_s = o_s, O_p = o_p, O_o = o_o] = 1$ A recent implementation of a triple store exploiting these intuitions is TENTRIS, see Bigerl et al. (2020). In this work, such decompositions are generalized into more generic CP formats, see Chapter 15. Approximations of grounding tensors by decompositions leads to embeddings of the individuals such as Tucker, ComplEx and RESCAL (see Nickel et al. (2016)).

For our purposes of evaluating logical formulas such as SPARQL queries we use the relational encoding of the groundings, which are depicted by



12.3.3 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 $\mathrm{WHERE}\cdot$ 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.

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 build a corresponding atom creating tensor (see Def. 65). The triple pattern is then evaluated by contraction of the atom creating 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

$$\psi_{\langle Z, \mathrm{rdftype}, g_k \rangle}[O_s, O_p, O_o, Z] = \delta[O_s, Z] \otimes e_{I^{-1}(\mathrm{rdftype})}[O_p] \otimes e_{I^{-1}(g_k)}[O_o] .$$

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

$$\psi_{\langle Z_0, g_k, Z_1 \rangle}[O_s, O_p, O_o, Z_0, Z_1] = \delta[O_s, Z_0] \otimes e_{I^{-1}(g_k)}[O_p] \otimes \delta[O_o, Z_1] .$$

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. 75).

Examples of triple patterns, drawn in Figure 30 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 30a.

$$\psi_{\langle Z, \mathrm{rdftype}, g_k \rangle}[O_s, O_p, O_o, Z] = \delta[O_s, Z] \otimes e_{I^{-1}(\mathrm{rdftype})}[O_p] \otimes e_{I^{-1}(C)}[O_o]$$

  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 30b. 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.

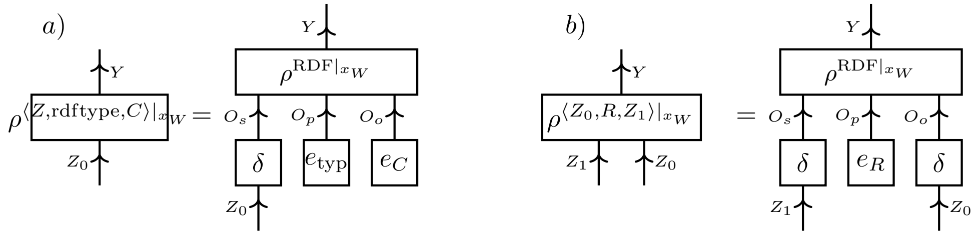


Figure 30: Triple patterns of SPARQL as tensor networks. a) Example of unary triple pattern $\langle Z, \mathrm{rdftype}, C\rangle$ specifying whether an individual I_{o_1} is a member of class C. b) Example of a binary triple pattern $\langle Z_0, R, Z_1\rangle$ specifying whether individuals I_{o_1} and I_{o_2} have a relation R. By $e_{\mathrm{typ}}, e_C, e_R$ we denote the one-hot encodings of the enumeration of the resources $rdf: type, C$ and R.

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$\{f_1, f_2\}$	$f_1 \vee f_2$	$\rho^{\vee}\left[Y_{f_1\vee f_2}, Y_{f_1}, Y_{f_2}\right]$
FILTER NOT EXISTS$\{f\}$	$\neg f$	$\rho^{\neg}\left[Y_{\neg f}, Y_f\right]$

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. 12.2.3. 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. 14.6 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 $\neg$.

12.4 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 Getoor and Taskar (2019).

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.

12.4.1 Hybrid First-Order Logic Networks

Following Richardson and Domingos (2006) 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 9 we directly allow for boolean base measures and call the distributions Hybrid First-Order Logic Networks.

Definition 69 (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 9, 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 $\sum_{l\in[p]}|\mathcal{U}|^{\left|O_{q_l}\right|}$. This polynomial in the cardinality of the domain set increase poses significant computational challenges, see Richardson and Domingos (2006). We will in the next sections explore an alternative way to apply the theory of Chapter 9 and Chapter 10, namely based on importance formulas.

12.4.2 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 $\{o_{[n]}^j\,:\,j\in[m]\}$ of indices to the important tuples we build a set encoding (see Def. 75)

$$\underline{p}=\sum_{j\in[m]}\left(\bigotimes_{l\in[n]}e_{o_l^j}\left[O_l\right]\right)\,.$$

We interpret the tensor $\underline{p}$ 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 $\underline{p}$ by designing a base measure

$$\nu^{\underline{p}}\left[X_W\right] = \begin{cases} 1 & \text{if} \quad p|_{x_W}\left[O_p\right] = \underline{p} \\ 0 & \text{else} \end{cases} .$$

The base measure restricts the HFLN to be those worlds, where $p|_{x_W}$ is coincides with the fixed tensor $\underline{p}$. Intuitively, $p|_{x_W}$ represents certain evidence about a first-order logic world, whereas other formulas are uncertain.

Assumption 1. *Given a base measure $\nu^{\underline{p}}\left[X_W\right]$, we assume that there is an importance formula $p\left[O_{[n]}\right]$ such that*

$$\nu^{\underline{p}}\left[X_W\right] = \begin{cases} 1 & \textit{if} \quad p|_{x_W}\left[O_p\right] = \underline{p} \\ 0 & \textit{else} \end{cases} .$$

12.4.3 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

$$q_l\left[O_{q_l}\right] = \left(p\left[O_{[n]}\right] \Rightarrow h_l\left[O_{q_l}\right]\right) \tag{37}$$

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.

Towards connecting with propositional logics, we further make the assumption, that we can decompose the formula h_l in what we will call extraction formulas.

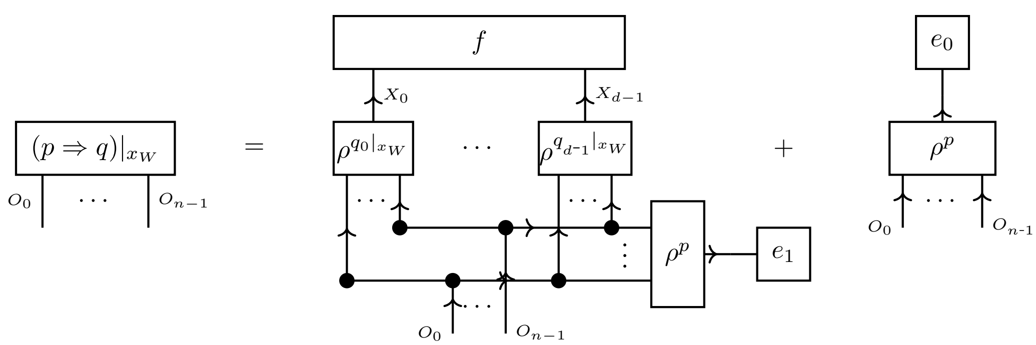
Assumption 2. *We assume that there exist formulas $\{q_k\left[O_{[n]}\right] \,:\, k \in [d]\}$, which we refer to as atom extraction formulas, and an importance formula $p\left[O_{[n]}\right]$ such that the following holds. To each first-order logic formula q_l there is another first-order logic formula $h_l\left[O_{q_l}\right]$ and a propositional formula $f_l\left[X_{[d]}\right]$ such that*

$$q_l\left[O_{q_l}\right] = \left(p\left[O_{[n]}\right] \Rightarrow h_l\left[O_{q_l}\right]\right)$$

and

$$h_l\left[O_{q_l}\right] = \left\langle \{f_l\left[X_{[d]}\right]\} \cup \{\rho^{q_k}\left[X_k, O_{[n]}\right] \,:\, k \in [d]\} \right\rangle \left[O_{f_l}\right] .$$

We depict the assumption, that any formula is of the form (37) 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 $\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}}}$, the probability of a world x_W with $p|_{x_W} = \underline{p}$ is

$$\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}}}\left[X_W = x_W\right] = \frac{1}{\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}}\right)} \exp\left[\sum_{l\in[p]} \theta\left[L = l\right] \left\langle (p \Rightarrow h)|_{x_W} \right\rangle \left[\varnothing\right]\right]$$

where the partition function is

$$\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}\right) = \sum_{x_W \,:\, p|_{x_W}\left[O_{[n]}\right] = \underline{p}[O_p]} \exp\left[\sum_{l\in[p]} \theta\left[L=l\right] \langle (p \Rightarrow h)|_{x_W}\rangle\left[\varnothing\right]\right] .$$

Let us now decompose the statistics into constant and varying terms. We have

$$\langle (p \Rightarrow h)|_{x_W}\rangle\left[\varnothing\right] = \langle p \wedge h|_{x_W}\rangle\left[\varnothing\right] + \langle \neg p|_{x_W}\rangle\left[\varnothing\right] ,$$

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

$$\langle p \wedge h|_{x_W}\rangle\left[\varnothing\right] = \sum_{j\in[m]} h|_{x_W}\left[O_{[n]} = o^j_{[n]}\right] .$$

Using these insights we decompose a normalized log likelihood as

$$\frac{1}{m}\ln\left[\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}}}\left[X_W = x_W\right]\right] = \frac{1}{m}\sum_{j\in[m]}\sum_{l\in[p]} \theta\left[L=l\right] h|_{x_W}\left[o_{[n]} = o^j_{[n]}\right] - \frac{1}{m}\ln\left[\frac{\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}\right)}{\exp\left[\langle\theta\rangle\left[\varnothing\right]\cdot\langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right]}\right] \tag{38}$$

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 (38) coincides with the first term of the likelihood

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F},\theta,\mathbb{I})}\right] = \frac{1}{m}\sum_{j\in[m]}\sum_{l\in[p]} \theta\left[L=l\right] f_l\left[D(j)\right] - \ln\left[\mathcal{Z}\left(\mathcal{F},\theta\right)\right]$$

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.

12.4.4 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 3. *Let $\nu^{\underline{p}}\left[X_W\right]$ be a base measure of worlds such that the vectors*

$$\left(q_0|_{\tilde{x_W}}\left[O_0 = o^j_0, \dots, O_{n-1} = o^j_{n-1}\right], \dots, q_{d-1}|_{\tilde{x_W}}\left[O_0 = o^j_0, \dots, O_{n-1} = o^j_{n-1}\right]\right) \tag{39}$$

for $j \in [m]$ are independent and identical distributed by the normation of a boolean base measure ν, when drawing X_W by $\nu^{\underline{p}}\left[X_W\right]$.

When these assumption holds, we now show that the probability of a first-order logic world coincides with the likelihood of a propositional logic dataset.

Theorem 84. *Let there be a set of formulas $\mathcal{Q}$ and a base measure $\nu^{\underline{p}}\left[X_W\right]$ such that Assumption 1, Assumption 2 and Assumption 3 hold. We then have for the likelihood of any by $\nu^{\underline{p}}\left[X_W\right]$ supported world x_W that*

$$\frac{1}{m}\ln\left[\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}}}\left[X_W = x_W\right]\right] = \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F},\theta}\right]$$

where $\mathcal{F}$ is the set of propositional equivalents to $\mathcal{Q}$ (see Assumption 2) and D the data map with evaluation at $j \in [m]$ by the enumerated non-vanishing coordinates of $\underline{p}\left[O_p\right]$

$$D(j) = \left(q_0|_{x_W}\left[O_0 = o^j_0, \dots, O_{n-1} = o^j_{n-1}\right], \dots, q_{d-1}|_{x_W}\left[O_0 = o^j_0, \dots, O_{n-1} = o^j_{n-1}\right]\right) .$$

To show the theorem, we show first in the following lemma the factorization of the partition function of the HFLN.

Lemma 15. *Given the assumptions of The. 84, we have*

$$\frac{\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}\right)}{\exp\left[\langle\theta\rangle\left[\varnothing\right] \cdot \langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right]} = \left(\mathcal{Z}\left(\mathcal{F}, \theta, \nu\right)\right)^m \,.$$

Proof. We have

$$\begin{aligned}
\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}\right) &= \mathbb{E}_{x_W \sim \nu^{\underline{p}}[X_W]}\left[\exp\left[\sum_{q\in\mathcal{Q}} \theta_q \left\langle (p \Rightarrow h)|_{x_W}\right\rangle\left[\varnothing\right]\right]\right] \\
&= \exp\left[\langle\theta\rangle\left[\varnothing\right] \cdot \langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right] \cdot \mathbb{E}_{x_W \sim \nu^{\underline{p}}[X_W]}\left[\exp\left[\sum_{q\in\mathcal{Q}} \theta_q \sum_{j\in[m]} h|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]\right]\right] \\
&= \exp\left[\langle\theta\rangle\left[\varnothing\right] \cdot \langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right] \cdot \mathbb{E}_{x_W \sim \nu^{\underline{p}}[X_W]}\left[\prod_{j\in[m]} \exp\left[\sum_{q\in\mathcal{Q}} \theta_q \cdot h|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]\right]\right]
\end{aligned}$$

Since the substitutions of the atom formulas at the respective object tuples are independent, also the variables

$$\exp\left[\theta_q \cdot h|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]\right]$$

for $j \in [m]$ are independent. We therefore get

$$\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}\right) = \exp\left[\langle\theta\rangle\left[\varnothing\right] \cdot \langle\neg p|_{x_W}\rangle\left[\varnothing\right]\right] \cdot \prod_{j\in[m]} \mathbb{E}_{x_W \sim \nu^{\underline{p}}[X_W]}\left[\exp\left[\sum_{q\in\mathcal{Q}} \theta_q \cdot h|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]\right]\right] \tag{40}$$

Each $h|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]$ depends by Assumption 2 only on the random tuple $\{q_k[O_{[n]} = o^j_{[n]}] \,:\, k \in [d]\}$. We build the expectation over all possible values $x_{[d]}$ of this tuple at any $j \in [m]$ and get

$$\begin{aligned}
&\mathbb{E}_{x_W \sim \nu^{\underline{p}}[X_W]}\left[\exp\left[\sum_{l\in[p]} \theta\left[L = l\right] \cdot h_l|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]\right]\right] \\
&\quad = \sum_{x_{[d]} \in \times_{k\in[d]}[2]} \mathbb{P}_{\forall k\in[d]\,:\, q_k[O_{[n]} = o^j_{[n]}] = x_k}\left[x_W \sim \nu^{\underline{p}}\left[X_W\right]\right] \cdot \exp\left[\sum_{l\in[p]} \theta\left[L = l\right] \cdot f_l\left[X_{[d]} = x_{[d]}\right]\right] \\
&= \sum_{x_{[d]} \in \times_{k\in[d]}[2]} \nu[X_{[d]} = x_{[d]}] \cdot \exp\left[\sum_{l\in[p]} \theta\left[L = l\right] \cdot f_l\left[X_{[d]} = x_{[d]}\right]\right] \\
&= \mathcal{Z}\left(\mathcal{F}, \theta, \nu\right) \,.
\end{aligned}$$

We arrive at the claim, when combining this equation with (40). ☐

With this lemma, we are now show The. 84.

Proof of The. 84. We have for the logarithm of the probability of a world x_W given the distribution $\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}}$, that

$$\ln\left[\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}}\left[X_W = x_W\right]\right] = \sum_{l\in[p]} \theta\left[L = l\right] \langle q_l|_{x_W}\rangle\left[\varnothing\right] - \ln\left[\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}\right)\right]$$

The first term obeys with Assumption 2

$$\sum_{l\in[p]} \theta\left[L = l\right] \langle q_l|_{x_W}\rangle\left[\varnothing\right] = \langle\theta\rangle\left[\varnothing\right] \cdot \langle\neg p|_{x_W}\rangle\left[\varnothing\right] + \sum_{l\in[p]} \sum_{j\in[m]} \theta\left[L = l\right] \cdot h_l\left[O_{[n]} = o^j_{[n]}\right] = \langle\theta\rangle\left[\varnothing\right] \cdot \langle\neg p|_{x_W}\rangle\left[\varnothing\right] + \sum_{l\in[p]} \sum_{j\in[m]} \theta\left[L =\right.$$

With Lem. 15 we have under the given assumptions for the second term

$$\ln\left[\mathcal{Z}\left(\mathcal{Q}|_{\mathcal{U}}, \theta, \nu^{\underline{p}}\right)\right] = m \cdot \ln\left[\mathcal{Z}\left(\mathcal{F}, \theta, \nu\right)\right] + \langle\theta\rangle\left[\varnothing\right] \cdot \langle\neg p|_{x_W}\rangle\left[\varnothing\right] .$$

Combining both, we have

$$\frac{1}{m}\ln\left[\mathbb{P}^{\mathcal{Q}|_{\mathcal{U}},\theta,\nu^{\underline{p}}}\left[X_W = x_W\right]\right] = \frac{1}{m}\sum_{j\in[m]}\sum_{l\in[p]}\theta\left[L=l\right]\cdot f_l\left[X_{[d]} = x^j_{[d]}\right] - \ln\left[\mathcal{Z}\left(\mathcal{F},\theta,\nu\right)\right]$$

which coincides with $\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F},\theta}\right]$. □

Let us now investigate, in which cases the Assumption 3 of independent data can be matched.

Lemma 16. *Let p and $q_0, \ldots, q_{d-1}$ be quantor and constant free and let the index tuples of the support of $\underline{p}$ be pairwise disjoint. Then the vectors (39) 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. □

There are situations, where Assumption 3 is violated. For example, when two object tuples are not disjoint, then some formulas might always coincide on both datapoints, which would violate independence.

In further situations the atom base ν are not the uniform $\mathbb{I}$:

- 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, in this case still boolean
- extraction formulas contradicting each other, more general not independent from each other

Let us notice, that non-boolean base measures could be treated in a same manner, but several developments in this work, such as cross-entropy decompositions in Chapter 5 would receive further terms.

Remark 22 (Approximation by Independent Samples)**.** *As observed above, we do not have independent samples in general. As a consequence, we cannot apply Lem. 15 to decompose the partition function term of the log-probability into factors to each solution map of p. In this case, it might be still benefitial to use the reduction to the likelihood of a HLN, but needs to understand it as a approximation to the true world probability.*

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.

12.5 Sample extraction from first-order logic worlds

The decomposition of the likelihood suggests the following approach to generate samples from groundings:

- Define a query formula p, which we decompose in the basis CP decomposition and interpret each slice as the one-hot encoding of the datapoint.
- Define for $k \in [d]$ queries q_k generating the the atoms X_k: Predicates along with assignment of variables / constants to its positions.
- Contract the groundings of each formula q_k with the grounding of p to build a data core

12.5.1 Representation by Tensor Networks

We model the extraction process as a relation between a tuple of individuals and the extracted world in the factored system of atoms X_k.

Definition 70. *Given a first-order logic world x_W, an importance formula p and extraction formulas q_k for $k \in [d]$, we define the extraction relation*

$$\mathcal{R} \subset \left(\bigtimes_{l\in[n]}[r]\right) \otimes \left(\bigtimes_{k\in[d]}[2]\right)$$

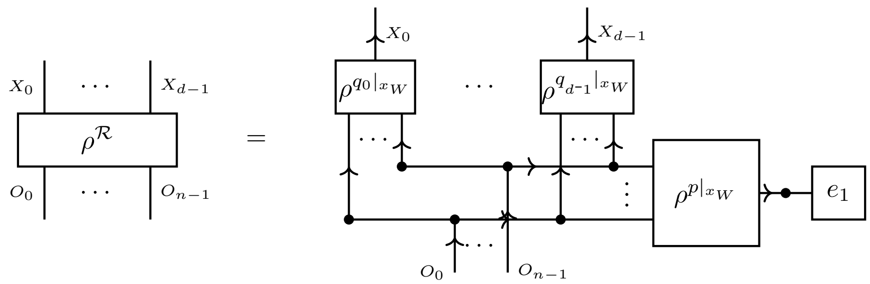
by

$$\mathcal{R} = \left\{(o_{[n]}, x_{[d]}) \,:\, p|_{x_W}\left[O_{[n]} = o_{[n]}\right] = 1\,,\, \forall k \in [d] \,:\, x_k = q_k\left[O_{[n]} = o_{[n]}\right]\right\}.$$

The encoding of an extraction relation is

$$\rho^{\mathcal{R}}\left[O_{[n]}, X_{[d]}\right] \subset \left(\bigotimes_{l\in[n]} \mathbb{R}^{r}\right) \otimes \left(\bigotimes_{k\in[d]} \mathbb{R}^{2}\right)$$

and drawn in a contraction diagram by



Here the contraction of ρ^{p} with the truth vector e_1 represents the matching condition posed by p when extracting pairs of individuals.

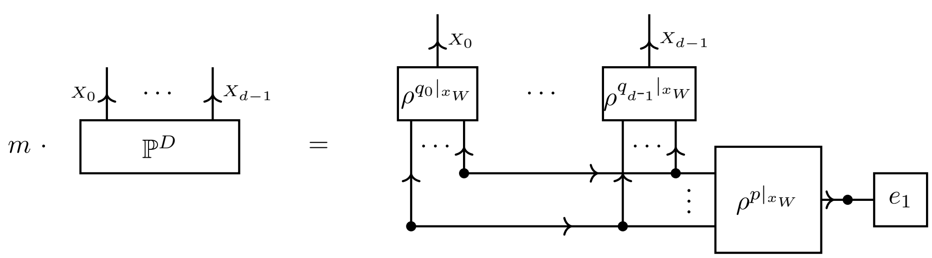
The empirical distribution is then the normalized contraction leaving only the legs to the extracted atomic formulas open, that is

$$\mathbb{P}^{D} = \frac{\left\langle \rho^{\mathcal{R}} \right\rangle \left[X_{[d]}\right]}{\left\langle \rho^{\mathcal{R}} \right\rangle [\varnothing]} \,.$$

Here the number of extracted data is the denominator

$$m = \left\langle \rho^{\mathcal{R}} \right\rangle [\varnothing] = \left\langle \rho^{p}\left[Y_p, O_{[n]}\right], e_1\left[Y_p\right] \right\rangle [\varnothing] \,.$$

We depict this by



12.5.2 Basis CP Decomposition of extracted data

To connect with the empirical distribution introduced in Sect. 4.8 we now show how the empirical distribution extracted from the interpretations of the formulas $p, q_0, \ldots, q_{d-1}$ on a first-order logic world x_W can be represented by tensor networks.

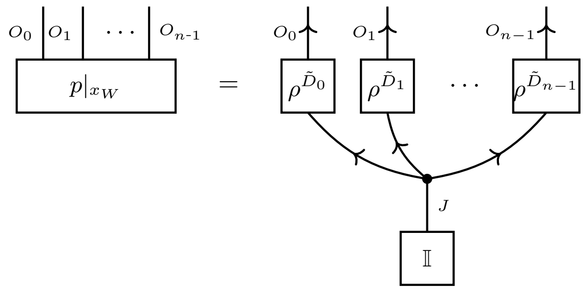
First of all, we decompose the importance formula into a basis CP format (see Chapter 15), that is a decomposition

$$p|_{x_W}\left[O_{[n]}\right] = \left\langle \left\{ V^{l}\left[O_l, J\right] \,:\, l \in [n] \right\} \right\rangle \left[O_{[n]}\right]$$

such that all $V^{l}\left[O_l, J\right]$ are directed and boolean tensors. Here an auxiliary variables J taking values in $[m]$ is introduced, which we call the data variable, which enumerates the non-vanishing coordinates of $p|_{x_W}$. With this decomposition, we can understand the decomposition of $p|_{x_W}\left[O_{[n]}\right]$ as a relational encoding of an term selection map $\tilde{D}$ with coordinate maps defined such that

$$\rho^{\tilde{D}_l}\left[O_l, J\right] = V^{l}\left[O_l, J\right] \,.$$

We depict this decomposition by:



Based on these construction, we now provide a tensor network decomposition of the extracted empirical distribution.

Theorem 85. *Given a first-order logic world x_W, an importance formula p and extraction formulas q_k for $k \in [d]$, we have*

$$\rho^{\mathcal{R}}\left[O_{[n]}, X_{[d]}\right] = \left\langle \{\rho^{q_k|_{x_W}}\left[X_k, O_{[n]}\right] \,:\, k \in [d]\} \cup \{\rho^{\tilde{D}_l}\left[O_l, J\right] \,:\, l \in [n]\} \right\rangle \left[O_{[n]}, X_{[d]}\right]$$

and thus

$$\mathbb{P}^D\left[X_{[d]}\right] = \frac{1}{m} \left\langle \{\rho^{q_k|_{x_W}}\left[X_k, O_{[n]}\right] \,:\, k \in [d]\} \cup \{\rho^{\tilde{D}_l}\left[O_l, J\right] \,:\, l \in [n]\} \right\rangle \left[X_{[d]}\right] .$$

Proof. To show the first claim, let us choose arbitrary state tuples $o_{[n]}$ and $x_{[d]}$. We then have

$$\begin{aligned}
&\left\langle \{\rho^{q_k|_{x_W}}\left[X_k, O_{[n]}\right] \,:\, k \in [d]\} \cup \{\rho^{\tilde{D}_l}\left[O_l, J\right] \,:\, l \in [n]\} \right\rangle \left[O_{[n]} = o_{[n]}, X_{[d]} = x_{[d]}\right] \\
&= \left\langle \{\rho^{q_k|_{x_W}}\left[X_k = x_k, O_{[n]} = o_{[n]}\right] \,:\, k \in [d]\} \cup \{\rho^{\tilde{D}_l}\left[O_l = o_l, J\right] \,:\, l \in [n]\} \right\rangle [\varnothing] .
\end{aligned}$$

This contraction evaluates to 1, if and only if for all $k \in [d]$ we have $\rho^{q_k|_{x_W}}\left[X_k, O_{[n]}\right] = 1$ and

$$\left\langle \{\rho^{\tilde{D}_l}\left[O_l = o_l, J\right] \,:\, l \in [n]\} \right\rangle [\varnothing] = 1 .$$

The first condition is equal to $x_k = q_k\left[O_{[n]} = o_{[n]}\right]$ for all $k \in [d]$ and the second to $p|_{x_W}\left[O_{[n]} = o_{[n]}\right] = 1$. Comparing with the definition of the extraction relation (see Def. 70), we notice that these conditions are equal to $(o_{[n]}, x_{[d]}) \in \mathcal{R}$ and therefore to

$$\rho^{\mathcal{R}}\left[O_{[n]} = o_{[n]}, X_{[d]} = x_{[d]}\right] .$$

The first claim follows, since $\rho^{\mathcal{R}}$ is boolean, as is the contraction of the cores $\rho^{q_k|_{x_W}}$ with the cores $\rho^{\tilde{D}_l}$, which leaves the outgoing variables $X_{[d]}$ open. The second claim follows from the first using that $\mathbb{P}^D\left[X_{[d]}\right] = \frac{1}{m}\left\langle \rho^{\mathcal{R}} \right\rangle \left[X_{[d]}\right]$. □

To connect with the representation of empirical distributions based on data cores (see Sect. 4.8), we now form data cores by contractions with the grounding of extraction formulas with the cores $\rho^{\tilde{D}_l}$ (see Figure 31),

$$\rho^{D_k}[X_k, J] = \left\langle \{\rho^{q_k|_{x_W}}\left[X_k, O_{[n]}\right]\} \cup \{V^l\left[O_l, J\right] \,:\, l \in [n]\} \right\rangle [X_k, J] .$$

The empirical distribution is then a tensor network of these tensors, as we show next.

Theorem 86. *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

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

Proof. By The. **??** we have

$$\rho^{\mathcal{R}}\left[O_{[n]}, X_{[d]}\right] = \left\langle \{\rho^{q_k|_{x_W}}\left[X_k, O_{[n]}\right] \,:\, k \in [d]\} \cup \{\rho^{\tilde{D}_l}\left[O_l, J\right] \,:\, l \in [n]\} \right\rangle \left[O_{[n]}, X_{[d]}\right] .$$

Since $\rho^{\tilde{D}_l}[O_l, J]$ are directed and boolean, they can be copied and separately contracted with each $q_k|_{x_W}$, without changing the contraction. We arrive at

$$\begin{aligned}
&\rho^{\mathcal{R}}\left[O_{[n]}, X_{[d]}\right] \\
&= \left\langle \left\{ \left\langle \{\rho^{q_k|_{x_W}}\left[X_k, O_{[n]}\right]\} \cup \{\rho^{\tilde{D}_l}\left[O_l, J\right] \,:\, l \in [n]\} \right\rangle \left[X_k, J\right] \,:\, k \in [d] \right\} \right\rangle \left[O_{[n]}, X_{[d]}\right] \\
&= \left\langle \{\rho^{D_k}[J, X_k] \,:\, k \in [d]\} \right\rangle \left[X_{[d]}\right] ,
\end{aligned}$$

which established the claim. □

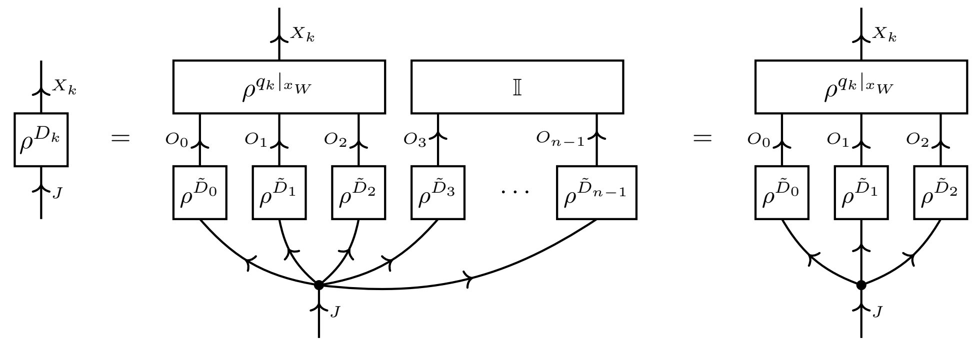


Figure 31: Generation of a data core for the variable X_k given an extraction formula q_k and an importance formula, which grounding is decomposed into a basis CP format with leg vectors $\rho^{\tilde{D}_l}[O_l, J]$. Term variables, which are appearing in the importance formula, but not in the extraction formula q_k can be treated trivally by contraction with the trivial tensor (here $O_4, \ldots, O_{n-1}$).

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. 7.3.2), treated as a categorical variable, with respect to the constant in the extraction query. The advantages are that we can avoid the ρ-formalism and directly model the categorical distributions.

This also enables a batchwise computation of multiple SPARQL queries, which differ only in one constant.

12.6 Generation of first-order logic worlds

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.

In the previous section we have described a way to extract an effective empirical distribution for the likelihood of a first-order logic world given a HFLN. We now want to investigate methods to reproduce an empirical distribution based on a constructed first-order logic world.

Definition 71 (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 $q_{[d]} = \{q_k \,:\, k \in [d]\}$ reproduces $\mathbb{P}^D$, when*

$$\mathbb{P}^D = \left\langle \{p|_{x_W}\} \cup \{\rho^{q_k|_{x_W} \,:\, k\in[d]}\} \right\rangle \left[X_{[d]}|\varnothing\right] .$$

Note that for distribution $\mathbb{P}$ to be reproducable, it needs to have rational coordinates. If any only if all coordinates are rational, we find a $m \in \mathbb{N}$ such that $\mathrm{im}\,(m \cdot \mathbb{P}) \subset \mathbb{N}$. We can then interpret m as the number of samples, and construct a sample selector map by understanding each coordinate of $m \cdot \mathbb{P}$ as the number of appearances of the respective world in the samples.

We show different schemes and give examples on Knowledge Graphs, where we provide examples for importance and extraction formulas by SPARQL queries.

12.6.1 Samples by single objects

In the first reproduction scheme we construct datapoints by dedicated objects, which represent a sample, that is we choose a domain $\mathcal{U} = [m]$.

Theorem 87. *Let there be an empirical distribution $\mathbb{P}^D$ to a sample selector map D (see Def. 32), we construct a world $x_W[L,O]$ with d unary predicates by*

$$x_W[L,O] = \sum_{k\in[d]} \sum_{j\in[m]\,:D_k(j)=1} e_k\left[L\right] \otimes e_j\left[O\right] \, .$$

We further choose a trivial importance query, that is

$$p|_{x_W}\left[O\right] = \mathbb{I}\left[O\right] \, ,$$

and extraction queries coinciding with the unary predicates, that is for $k \in [d]$

$$q_k = g_k \, .$$

Then, the triple $(x_W, p, q_{[d]})$ reproduces $\mathbb{P}^D$.

Proof. By The. 86 it is enough to show, that the data cores constructed from the data extraction process coincide with those of $\mathbb{P}^D$. We enumerate to this end the non-vanishing coordinates of $p|_{x_W}$ by the data variable J taking values $j \in [m]$, as

$$p|_{x_W}\left[O=j\right] = 1$$

and choose

$$\tilde{D} = \delta \, .$$

For arbitrary $k \in [d]$ and $j \in [m]$ we now have

$$\rho^{D_k}[X_k, J=j] = \left\langle \rho^{q_k|_{x_W}}\left[X_k, O\right], V^0\left[O, J=j\right] \right\rangle [X_k, J] = \left\langle \rho^{q_k|_{x_W}}\left[X_k, O\right], e_{\tilde{D}(j)}\left[O\right] \right\rangle [X_k, J=j] = e_{D_k(j)}\left[X_k\right] \, .$$

This coincides with the slice of the data core of the CP representation of empirical distributions used in The. 15. Since the slice and the core was arbitrary, the tensor network representations in The. 15 and The. 86 are equal and thus the triple $(x_W, p, q_{[d]})$ reproduces $\mathbb{P}^D$. □

We now give by the next theorem an example of a Knowledge Graph with SPARQL queries reproducing and arbitrary empirical distribution.

Theorem 88. *Let $\mathbb{P}^D$ be an empirical distribution to the sample selector D. We construct a Knowledge Graph of the resources $\mathcal{U} = \{s_j \,:\, j \in [m]\} \cup \{C\} \cup \{C_k \,:\, k \in [d]\}$, where s_j represent samples and C_k unary predicates, by*

$$\mathrm{RDF}|_{x_W} = \sum_{j\in[m]} e_{I_{s_j}} O_s \otimes e_{I_{\mathrm{rdftype}}} O_p \otimes e_{I_C} O_o + \sum_{j\in[m]} \sum_{k\in[d]\,:\,D_k(j)=1} e_{I_{s_j}} O_s \otimes e_{I_{\mathrm{rdftype}}} O_p \otimes e_{I_{C_k}} O_o \, .$$

We further define an importance formula by the SPARQL *query*

p= SELECT { ?x } WHERE { ?x $\mathrm{rdf:type}$ *C .}*

and for each $k \in [d]$ an extraction formula by the query

q_k = SELECT { ?x } WHERE { ?x $\mathrm{rdf:type}$ *C_k .} .*

Then the triple $(\mathrm{KG}|_{x_W}, p, q_{[d]})$ reproduces $\mathbb{P}^D$.

Proof. We show the theorem analogously to The. 87, with the slide difference in the importance formula. We have for the grounding of p on $\mathrm{KG}|_{x_W}$ that

$$p|_{x_W}\left[O\right] = \sum_{j\in[m]} e_{I_{s_j}} O$$

and enumerate the non-vanishing coordinates by J.

For each extraction formula we have

$$q_k|_{x_W}[O] = \sum_{j\in[m]\,:\,D_k(j)=1} e_{I_{s_j}} O\,.$$

It follows that the data cores used in The. 86 are

$$\rho^{D_k}[X_k, j] = e_0[X_k] \otimes \left(\sum_{j\in[m]\,:\,D_k(j)=0} e_j[J] \right) + e_1[X_k] \otimes \left(\sum_{j\in[m]\,:\,D_k(j)=1} e_j[J] \right)$$

and they thus coincide with those in the decomposition in The. 15. The claim follows therefore with the same argumentation as in the proof of The. 87. □

Let us provide some more insights on the construction of the reproducing Knowledge Graph in The. **??**. By the insertions to the one-hot encodings $e_{I_{s_j}} O_s \otimes e_{I_{\mathrm{rdftype}}} O_p \otimes e_{I_C} O_o$ we mark each sample representing resource by a class and ensure its appearance as a $\mathrm{owl}:\mathrm{NamedIndividual}$ in the graph. The insertions $e_{I_{s_j}} O_s \otimes e_{I_{\mathrm{rdftype}}} O_p \otimes e_{I_{C_k}} O_o$ on the other side encode the sample selecting map, by inserting exactly the assertions corresponding with the respective sample. In this simple Knowledge Graph, Description Logic is expressive enough to represent any formula q composed of the formulas $q_0, \dots, q_{d-1}$.

12.6.2 Samples by pairs of objects

We now instantiate multiple objects for each datapoint, one for each variable of the importance formula, i.e. $\mathcal{U} = [m] \times [n]$ Label individuals $s_{j,l}$ by data index and variable index.

Lemma 17. *Let there a data map D, queries $p, q_{[d]}$ and a first-order logic world containing objects $s_{j,l}$ for $j \in [m]$ and $l \in [n]$ If*

$$p|_{x_W} = \sum_{j\in[m]} \bigotimes_{l\in[n]} e_{I_{s_{j,l}}}[O_l]$$

and for any $k \in [d]$

$$q_k|_{x_W} = \sum_{j:D_k(j)=1} \bigotimes_{O_l \in O_{q_k}} e_{I_{s_{j,l}}}[O_l]\ .$$

Then the tuple $(\mathrm{KG}|_{x_W}, p, \{q_0, \dots, q_{d-1}\})$ reproduces $\mathbb{P}^D$.

Proof. We notice, that the grounding of the importance formula is in a basis CP format, since by assumption

$$p|_{x_W} = \sum_{j\in[m]} \bigotimes_{l\in[n]} e_{I_{s_{j,l}}}[O_l]\ .$$

We choose J to enumerate the non-vanishing entries and get a term selecting map

$$\tilde{D}_l(j) = I_{s_{j,l}}\ .$$

From this we have

$$\left\langle \{\rho^{q_k|_{x_W}}[X_k, O_{q_k}]\} \cup \{\rho^{\tilde{D}_l}[O_l, J] \,:\, l \in [n]\} \right\rangle [X_k, J] = \rho^{D_k}[X_k, J]$$

and the claim follows with the same argumentation as in the proof of The. 87. □

12.7 Discussion

Statistical Models are called Probabilistic Relational Models. Extensions are models that also handle structural uncertainty, i.e. distributions of worlds with varying $\mathcal{U}$.

In the emerging area of network science Barabási (2016); Giovanni Russo Vito Latora (2017), statistical models for random graphs are investigated. Statistical Models of first-order logic go beyond the typical single edge type perspective of network science.

Remark 23 (Alternative Representation of empirical distributions)**.** *So far, we have motivated the representation of empirical distributions based on basis CP decompositions based on data maps. In this section, based on the extraction queries, we have observed that empirical distributions might have more efficient representation formats. 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.*

Part III

Contraction Calculus

13 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 14), we call this scheme of storing and retrieving information the coordinate calculus.

13.1 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 18. *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*

$$\left\langle e_{x_{[d]}}\left[X_{[d]}\right], e_{\tilde{x}_{[d]}}\left[X_{[d]}\right]\right\rangle[\varnothing] = \delta_{x_{[d]},\tilde{x}_{[d]}} := \begin{cases} 1 & \textit{if} \quad x_{[d]} = \tilde{x}_{[d]} \\ 0 & \textit{else} \end{cases} .$$

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

$$\left\langle e_{x_{[d]}}\left[X_{[d]}\right], e_{\tilde{x}_{[d]}}\left[X_{[d]}\right]\right\rangle[\varnothing] = \prod_{k\in[d]} \left\langle e_{x_k}\left[X_k\right], e_{\tilde{x}_k}\left[X_k\right]\right\rangle[\varnothing] = \prod_{k\in[d]} \delta_{x_k,\tilde{x}_k} = \delta_{x_{[d]},\tilde{x}_{[d]}} .$$

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 72. *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 89 (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. 18 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. $\square$

Coordinate calculus is the representation of real-valued functions as tensors, from which its evaluations can be retrieved by the scheme of The. 89. This is in contrast to the basis calculus scheme to be discussed (see The. 98), 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 90. *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*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_A, X_B = x_B] = \left\langle \{ \langle T^e \rangle \left[X_{e/B}, X_{e\cap B} = x_{e\cap B}\right] \,:\, e \in \mathcal{E} \} \right\rangle [X_A] \, .$$

Proof. By definition of contractions we have for any x_A

$$\begin{aligned}
\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_A = x_A, X_B = x_B] &= \sum_{x_{\mathcal{V}/(A\cup B)} \in \bigtimes_{v\in\mathcal{V}/(A\cup B)}[m_v]} \prod_{e\in\mathcal{E}} T^e \left[X_{e/B} = x_{e/B}, X_{e\cap B} = x_{e\cap B}\right] \\
&= \left\langle \{ \langle T^e \rangle [\varnothing] \, X_{e/(A\cup B)}, X_{e\cap A} = x_{e\cap A}, X_{e\cap B} = x_{e\cap B} \,:\, e \in \mathcal{E} \} \right\rangle [\varnothing] \\
&= \left\langle \{ \langle T^e \rangle [\varnothing] \, X_{e/B}, X_{e\cap B} = x_{e\cap B} \,:\, e \in \mathcal{E} \} \right\rangle [X_A = x_A]
\end{aligned}$$

and the claim follows. $\square$

If we retrieve a single coordinate of a tensor, we have the situation $A = \varnothing$, $B = \mathcal{V}$. In that case, Theorem 90 shows, that the coordinate is the product of the coordinates of the cores.

13.2 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 73. *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 14, see The. 100.

In the following lemma, we state that coordinatewise transforms can be restricted to slices of tensors, when Although this is an obvious fact, this property can tremendously reduce the computational demand of contractions with coordinatewise transforms of tensors.

Lemma 19. *For any function $h : \mathbb{R} \to \mathbb{R}$, any tensor $T\left[X_{[d]}\right]$ and index x_A, where $A \subset [d]$, we have*

$$h\left(T\left[X_{[d]}\right]\right)\left[X_{[d]/A}, X_A = x_A\right] = h\left(X_{[d]/A}, X_A = x_A\right)\left[X_{[d]/A}\right] .$$

Proof. For any state $x_{[d]/A}$ we have that

$$h\left(T\left[X_{[d]}\right]\right)\left[X_{[d]/A} = x_{[d]/A}, X_A = x_A\right] = h\left(T\left[X_{[d]} = x_{[d]}\right]\right) = h\left(X_{[d]/A}, X_A = x_A\right)\left[X_{[d]/A} = x_{[d]/A}\right] .$$

□

Example 17 (Hadamard products as coordinatewise transforms)**.** *Hadamard products of tensors (see Example 3) are a special way of coordinate calculus, where the transform is the product and thus*

$$\cdot\left(T^0, \ldots, T^{p-1}\right)\left[X_{[d]}\right] = \left\langle\left\{T^l\left[X_{[d]}\right] : l \in [p]\right\}\right\rangle\left[X_{[d]}\right] .$$

These hadamard products are applied in the effective computation of conjunctions, as we will discuss in more detail in Sect. 14.6.

Example 18 (Exponentiation of energies)**.** *In Def. 28 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*

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

Both the nominator and the denominator involve a coordinatewise transform of the energy tensor $E^{(\phi,\theta,\nu)}[X_{[d]}]$ by the exponentiation. The. 13 provided a transform-free contraction expression by relational encodings, which is the central tool of basis calculus (see Chapter 14).

Let us note, that Lem. 19 enables the energy-based answering of conditional queries, as has been shown in The. 16.

13.3 Directed Tensors

Directionality as defined in Def. 12 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. 22, and refered to Markov Networks (see Def. **??**) satisfying these by Bayesian Networks (see Def. 27). To support our findings therein, we now discuss in more detail the connection between directed hypergraphs and directed tensors.

Definition 74 (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 19 (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

$$\delta^{[d],m}\left[[d], m\right] = \begin{cases} 1 & \textit{if} \quad x_0 = \ldots = x_{d-1} \\ 0 & \textit{else} \end{cases} . \tag{41}$$

The contractions with respect to subsets $\tilde{\mathcal{V}} \subset [d]$ are

$$\left\langle \delta^{[d],m} \right\rangle [X_{\tilde{\mathcal{V}}}] = \begin{cases} m & \textit{if} \quad \tilde{\mathcal{V}} = \varnothing \\ \mathbb{I}[X_{\tilde{\mathcal{V}}}] & \textit{if} \quad \left|\tilde{\mathcal{V}}\right| = 1 \\ \delta^{\tilde{\mathcal{V}},m}\left[\tilde{\mathcal{V}}, m\right] & \textit{else} \end{cases} . \tag{42}$$

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 20. *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 $\Delta^{\mathcal{G}} = \big\{\{v\} \cup \{v^e : e \ni v\} : v \in \mathcal{V}\big\}$ 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

$$\begin{aligned} \left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] &= \left\langle \{T^{\tilde{e}}[X_{\{v^e : v \in e\}}] : e \in \mathcal{E}\} \cup \{\delta^{\{v\} \cup \{v^e : e \ni v\}}[X_{\{v^e : e \ni v\}}, X_v = x_v] : v \in \mathcal{V}\} \right\rangle [\varnothing] \\ &= \left\langle \{T^{\tilde{e}}[X_{\{v^e : v \in e\}} = x_{\{v : v \in e\}}] : e \in \mathcal{E}\} \right\rangle [\varnothing] \\ &= \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] , \end{aligned}$$

which establishes the claim. □

13.3.1 Normation

Normed tensors (see Def. 13) are directed and directed tensors invariant under normation wrt their incoming and outgoing variable, as we show next.

Theorem 91. *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. 12, $\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\mathcal{V}^{\text{out}} | \mathcal{V}^{\text{in}}\right]$ is thus directed. □

The normation operation coincides in cases of non-negative tensors with the conditioning of a Markov Network representing a probability distribution.

13.3.2 Normation Equations

Normation equations capture certain properties of normations of tensors. We first show that any normable tensor is the contraction of its normation and an accompanying contraction, which generalizes the Bayes The. 4 towards more generic normable tensors.

Theorem 92 (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

$$\langle T\rangle\left[X_{\mathcal{V}^{\text{in}}}=x_{\mathcal{V}^{\text{in}}}|X_{\mathcal{V}^{\text{out}}}=x_{\mathcal{V}^{\text{out}}}\right]=\frac{\langle T\rangle\left[X_{\mathcal{V}^{\text{in}}}=x_{\mathcal{V}^{\text{in}}},X_{\mathcal{V}^{\text{out}}}=x_{\mathcal{V}^{\text{out}}}\right]}{\langle T\rangle\left[X_{\mathcal{V}^{\text{in}}}=x_{\mathcal{V}^{\text{in}}}\right]}$$

and therefor

$$\langle T\rangle\left[X_{\mathcal{V}^{\text{in}}}=x_{\mathcal{V}^{\text{in}}},X_{\mathcal{V}^{\text{out}}}=x_{\mathcal{V}^{\text{out}}}\right]=\langle T\rangle\left[X_{\mathcal{V}^{\text{in}}}=x_{\mathcal{V}^{\text{in}}}|X_{\mathcal{V}^{\text{out}}}=x_{\mathcal{V}^{\text{out}}}\right]\cdot\langle T\rangle\left[X_{\mathcal{V}^{\text{in}}}=x_{\mathcal{V}^{\text{in}}}\right]$$

Since the equation holds for arbitrary indices, the claim is established. □

Based on this property, we now show a generic decomposition scheme of tensors, which generalizes the chain rule of The. 5.

Theorem 93 (Generic Chain Rule). *For any Tensor $T[X_{\mathcal{V}}]$ and any total order $\prec$ on the nodes $\mathcal{V}$ we have*

$$T[X_{\mathcal{V}}]=\left\langle\left\{\langle T\rangle\left[X_v|X_{\{\tilde{v}\,:\,\tilde{v}\prec v,\tilde{v}\neq v\}}\right]\,:\,v\in\mathcal{V}\right\}\right\rangle[X_{\mathcal{V}}]\ ,$$

provided that the normations exist.

Proof. We apply The. 92 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. □

13.3.3 Contraction of Directed Tensors

Let us now investigate, which contractions inherit the directionality of the tensors.

Theorem 94. *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. 20 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.

$n\rightarrow n-1$**:** 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 121 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

$$\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle[X_{\mathcal{V}^{\text{in}}}]=\left\langle\mathcal{T}^{(\mathcal{V},\mathcal{E}/\{e\})}\}\right\rangle[X_{\mathcal{V}^{\text{in}}}]=\mathbb{I}[X_{\mathcal{V}^{\text{in}}}]\ .$$

□

13.4 Proof of Hammersley-Clifford Theorem

Let us now proof the Hammersley-Clifford theorem formulated in Chapter 4 as The. 10. Different to the original statement (see Clifford and Hammersley (1971)), we here proof the analogous statement for hypergraphs, where we have to demand the property of clique-capturing defined in Def. 25. We start with showing the following Lemmata to be exploited in the proof.

Lemma 21. *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

$$\begin{aligned}
&\ln \left[\prod_{\tilde{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} \Bigg] \\
&= \sum_{\bar{\mathcal{V}} \subset \mathcal{V}} \ln \left[\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right] \right] \left(\sum_{\tilde{\mathcal{V}} \subset \mathcal{V} \, : \, \bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} (-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|} \right) \\
&= \sum_{\bar{\mathcal{V}} \subset \mathcal{V}} \ln \left[\langle T \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right] \right] \left(\sum_{i \in [|\mathcal{V}| - |\bar{\mathcal{V}}|]} (-1)^i \binom{|\mathcal{V}| - |\bar{\mathcal{V}}|}{i} \right)
\end{aligned}$$

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 22. *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*

$$\langle T \rangle \left[X_{A,B} | X_{\mathcal{V}/\{A,B\}}\right] = \left\langle \langle T \rangle \left[X_A | X_{\mathcal{V}/\{A,B\}}\right], \langle T \rangle \left[X_B | X_{\mathcal{V}/\{A,B\}}\right] \right\rangle [X_{\tilde{\mathcal{V}}}]$$

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

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} (Z_{\bar{\mathcal{V}}})^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} = \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}/\{A,B\}} \left(\frac{Z_{\bar{\mathcal{V}}} \cdot Z_{\bar{\mathcal{V}} \cup \{A,B\}}}{Z_{\bar{\mathcal{V}} \cup \{A\}} \cdot Z_{\bar{\mathcal{V}} \cup \{B\}}}\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} . \tag{43}$$

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 (43) 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 (43) is trivial, which establishes the claim. □

We are finally ready to proof the Hammersley-Clifford The. 10 based on the Lemmata above.

Proof of The. 10. By Lem. 21 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. 22 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

$$\begin{aligned}
\mathbb{P}\left[X_{\mathcal{V}}\right] &= \langle \{T^e\left[X_e\right] \,:\, e \in \mathcal{E}\} \rangle \left[X_{\mathcal{V}}\right] \\
&= \langle \{T^e\left[X_e\right] \,:\, e \in \mathcal{E}\} \rangle \left[X_v | \varnothing\right] .
\end{aligned}$$

We have thus constructed a Markov Network with trivial partition function, which contraction coincides with the probability distribution. □

13.5 Differentiation of Contraction

The structured mean field approaches discussed in Chapter 5 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 23. *For any tensor network $\mathcal{T}^{\mathcal{G}}$ with positive T^e we have*

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

Proof. By multilinearity of tensor network contractions we have

$$\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right]=\left\langle\left\{\delta\left[Y_e,X_e\right]\right\}\cup\left\{T^{\tilde{e}}\left[X_{\tilde{e}}\right]\,:\,\tilde{e}\neq e\right\}\right\rangle\left[Y_e,X_{\mathcal{V}}\right]$$

and thus

$$\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]=\left\langle\left\{\delta\left[Y_e,X_e\right]\right\}\cup\left\{T^{\tilde{e}}\left[X_{\tilde{e}}\right]\,:\,\tilde{e}\neq e\right\}\right\rangle\left[Y_e\right].$$

Using both we get

$$\begin{aligned}\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]&=\frac{\partial}{\partial T^e\left[Y_e\right]}\frac{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right]}{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}\\&=\frac{\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right]}{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}-\frac{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right]\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}{\left(\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]\right)^2}\\&=\frac{\left\langle\left\{\delta\left[Y_e,X_e\right]\right\}\cup\left\{T^{\tilde{e}}\left[X_{\tilde{e}}\right]\,:\,\tilde{e}\neq e\right\}\right\rangle\left[Y_e,X_{\mathcal{V}}\right]}{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}\\&\quad-\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]\cdot\frac{\left\langle\left\{\delta\left[Y_e,X_e\right]\right\}\cup\left\{T^{\tilde{e}}\left[X_{\tilde{e}}\right]\,:\,\tilde{e}\neq e\right\}\right\rangle\left[Y_e\right]}{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}\end{aligned}\tag{44}$$

For the first term we get with a normation equation (see Theorem 92) that

$$\begin{aligned}\frac{\left\langle\left\{\delta\left[Y_e,X_e\right]\right\}\cup\left\{T^{\tilde{e}}\left[X_{\tilde{e}}\right]\,:\,\tilde{e}\neq e\right\}\right\rangle\left[Y_e,X_{\mathcal{V}}\right]}{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}&=\frac{\left\langle\left\{\delta\left[Y_e,X_e\right]\right\}\cup\left\{T^{\tilde{e}}\left[X_{\tilde{e}}\right]\,:\,\tilde{e}\in\mathcal{E}\right\}\right\rangle\left[Y_e,X_{\mathcal{V}}\right]}{T^e\left[X_e\right]\cdot\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[\varnothing\right]}\\&=\frac{\left\langle\delta\left[Y_e,X_e\right],\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]\right\rangle\left[Y_e,X_{\mathcal{V}}\right]}{T^e\left[X_e\right]}\\&=\frac{\left\langle\delta\left[Y_e,X_e\right],\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_e|\varnothing\right],\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}/e}|X_e\right]\right\rangle\left[Y_e,X_{\mathcal{V}}\right]}{T^e\left[X_e\right]}.\end{aligned}$$

Analogously, we have

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

With (44), 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 24. *For any function $f(T^e)[X_{\mathcal{V}}]$ we have*

$$
\begin{aligned}
&\frac{\partial}{\partial T^e\left[Y_e\right]}\left\langle\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], f(T^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] \\
&\quad=\frac{\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_e=x_e|\varnothing\right]}{T^e\left[X_e=x_e\right]}\Big(\left\langle\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}/e}|X_e=x_e\right], f(T^e)[X_{\mathcal{V}},Y_e]\right\rangle[\varnothing] \\
&\qquad\qquad-\left\langle\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], f(T^e)[X_{\mathcal{V}}]\right\rangle[\varnothing]\Big) \\
&\quad+\left\langle\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]\frac{\partial f(T^e)[X_{\mathcal{V}}]}{\partial T^e\left[Y_e\right]}\right\rangle[\varnothing]
\end{aligned}
$$

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. 23 on the first term. □

13.6 Discussion

Representations of linear maps is the typical application of tensors, reason for refering to tensor networks as multilinear algebra.

14 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. 11) or directed (see Def. 12). 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 32.

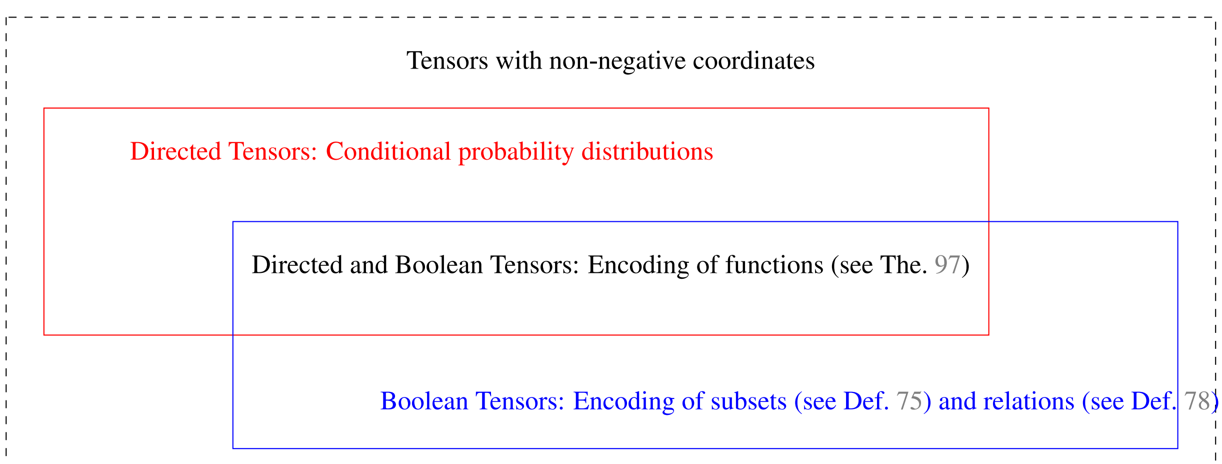


Figure 32: Sketch of the tensors with non-negative coordinates. We investigate in this chapter tensors, which are directed and boolean.

14.1 Encoding of Subsets

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 75 (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

$$e_{\mathcal{V}}\left[O = o\right] = \begin{cases} 1 & \mathit{if} I(o) \in \mathcal{V} \\ 0 & \mathit{else} \end{cases} .$$

In a one-hot basis decomposition we have

$$e_{\mathcal{V}}\left[O\right] \coloneqq \sum_{o \in [|\mathcal{U}|] \,:\, I(o) \in \mathcal{V}} e_o\left[O\right] .$$

The inclusion of subsets is represented by the partial ordering of tensors. Let us first define this property for arbitrary tensors.

Definition 76 (Partial ordering of tensors)**.** *We say that two tensors $f\left[X_{[d]}\right]$ and $h\left[X_{[d]}\right]$ attached with the same variables are partially ordered, denoted by*

$$f \prec h ,$$

if for all $x_{[d]} \in \bigtimes_{k \in [d]} [m_k]$

$$f\left[X_{[d]} = x_{[d]}\right] \leq h\left[x_{[d]}\right] .$$

For boolean tensors, the partially ordering is equal to a subset relation of the coordinates with value 1, as we show next.

Theorem 95. *Let $\mathcal{U}$ be an arbitrary set enumerated by the variable O and index interpretation function I. For two subsets $\mathcal{U}^0, \mathcal{U}^1$ of $\mathcal{U}$ we have*

$$\mathcal{U}^0 \subset \mathcal{U}^1$$

if and only if

$$e_{\mathcal{U}^0}\left[O\right] \prec e_{\mathcal{U}^1}\left[O\right] .$$

Proof. We have $\mathcal{U}^0 \subset \mathcal{U}^1$ if and only if

$$\forall o \in [|\mathcal{U}|] \big(I(o) \in \mathcal{U}^0\big) \Rightarrow \big(I(o) \in \mathcal{U}^1\big) ,$$

which is equal to

$$\forall o \in [|\mathcal{U}|] \big(e_{\mathcal{U}^0}\left[O = o\right] = 1\big) \Rightarrow \big(e_{\mathcal{U}^0}\left[O = o\right] = 1\big) .$$

Since subset encodings are boolean tensors, this is equivalent to

$$e_{\mathcal{U}^0}\left[O\right] \prec e_{\mathcal{U}^1}\left[O\right] .$$

□

14.1.1 Binary Relations

Since relations are subsets of cartesian products between two sets, their encoding is a straightforward generalization of Def. 75.

Definition 77 (Relation Encoding)**.** *A relation between two finite sets $\mathcal{U}^{\mathrm{in}}$ and $\mathcal{U}^{\mathrm{out}}$ is a subset of their cartesian product*

$$\mathcal{R} \subset \mathcal{U}^{\mathrm{in}} \times \mathcal{U}^{\mathrm{out}} .$$

Given an enumeration of $\mathcal{U}^{\mathrm{in}}$ and $\mathcal{U}^{\mathrm{out}}$ 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 $e_{\mathcal{R}}[O_{\mathrm{in}}, O_{\mathrm{out}}]$ with the coordinates

$$e_{\mathcal{R}}\left[O_{\mathrm{in}} = o_{\mathrm{in}}, O_{\mathrm{out}} = o_{\mathrm{out}}\right] = \begin{cases} 1 & \textit{if } (I_{\mathrm{in}}(o_{\mathrm{in}}), I_{\mathrm{out}}(o_{\mathrm{out}})) \in \mathcal{R} \\ 0 & \textit{else} \end{cases} .$$

The relation encoding has a decomposition into one-hot encodings as

$$e_{\mathcal{R}}\left[O_{\mathrm{in}}, O_{\mathrm{out}}\right] = \sum_{o_{\mathrm{in}}, o_{\mathrm{out}} \,:\, (I_{\mathrm{in}}(o_{\mathrm{in}}), I_{\mathrm{out}}(o_{\mathrm{out}})) \in \mathcal{R}} e_{o_{\mathrm{in}}}\left[O_{\mathrm{in}}\right] \otimes e_{o_{\mathrm{out}}}\left[O_{\mathrm{out}}\right] .$$

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.

14.1.2 Higher order relations

We can extend this contraction to relations of higher order, and arrive at encoding schemes usable for relational databases.

Definition 78. *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

$$e_{\mathcal{R}}\left[O_{[d]} = o_{[d]}\right] = \begin{cases} 1 & \textit{if} \quad (I_0(o_0), \ldots, I_{d-1}(o_{d-1})) \in \mathcal{R} \\ 0 & \textit{else} \end{cases} .$$

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. 40 by their relational encoding.

Theorem 96. *The encoding of any d-ary relation*

$$\mathcal{R} = \{x^i_{[d]} \,:\, i \in [n]\} \subset \bigtimes_{k \in [d]} \{\mathrm{False}, \mathrm{True}\}$$

where the objects in $\{\mathrm{False}, \mathrm{True}\}$ are enumerated by X_k with the standard index interpretation function (see Sect. 6.1)

$$I(\mathrm{True}) = 1 \quad \textit{and} \quad I(\mathrm{False}) = 0 ,$$

coincides with the propositional formula

$$f\left[X_{[d]}\right] = \bigvee_{i \in [n]} Z^{\wedge}_{x^i_{[d]}}, .$$

Proof. By definition, the encoding $e_{\mathcal{R}}$ is decomposed as

$$e_{\mathcal{R}}\left[X_{[d]}\right] = \sum_{i \in [n]} e_{x^i_{[d]}}\left[X_{[d]}\right] .$$

By The. 36 this is equal to

$$T\left[X_{[d]}\right] = \left(\bigvee_{x_{[d]} \,:\, T\left[X_{[d]} = x_{[d]}\right] = 1} Z^{\wedge}_{\{k : x_k = 0\}, \{k : x_k = 0\}} \right) \left[X_{[d]}\right] = \bigvee_{i \in [n]} Z^{\wedge}_{x^i_{[d]}}, .$$

□

Example 20 (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

$$\prod_{k\in[d]} |\mathcal{U}^k|$$

and therefore growing exponentially with the number of variables. Relations are however often sparse, in the sense that

$$|\mathcal{R}| << \prod_{k\in[d]} |\mathcal{U}^k| \,.$$

It is therefore often benefitially to choose sparse encoding schemes, for example by restricted CP formats (see Chapter 15) to represent $e_{\mathcal{R}}$.

14.2 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.

14.2.1 Relational Encoding of Functions

Definition 79 (Relational Encoding of Maps)**.** *Any map*

$$f : \mathcal{U}^{\mathrm{in}} \to \mathcal{U}^{\mathrm{out}}$$

can be represented by a relation

$$\mathcal{R}^f := \left\{(x, f(x)\,:\, x\in\mathcal{U}^{\mathrm{in}})\right\} \subset \mathcal{U}^{\mathrm{in}} \times \mathcal{U}^{\mathrm{out}} \,.$$

Given a enumeration of the sets by O_{in} and O_{out} we define the relational encoding of f as the tensor

$$\rho^f\left[O_{\mathrm{in}}, O_{\mathrm{out}}\right] = e_{\mathcal{R}^f}\left[O_{\mathrm{in}}, O_{\mathrm{out}}\right] \,.$$

Remark 24 (Reduction to images)**.** *When f maps into a set of infinite cardinality, we restrict $\mathcal{U}^{\mathrm{out}}$ 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. $\mathcal{U}^{\mathrm{out}} = \mathbb{R}$. While the variable O_f can in general be of the same cardinality as the domain set $\mathcal{U}^{\mathrm{in}}$, 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 97. *Let $\mathcal{U}^{\mathrm{in}}$,$\mathcal{U}^{\mathrm{out}}$ be sets and $\mathcal{R} \subset \mathcal{U}^{\mathrm{in}} \times \mathcal{U}^{\mathrm{out}}$ a relation. If and only if there exists a map $f : \mathcal{U}^{\mathrm{in}} \to \mathcal{U}^{\mathrm{out}}$ 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. "$\Rightarrow$": When f is a function, we have for any $o_{\mathrm{in}} \in [r_{\mathrm{in}}]$

$$\sum_{o_{\mathrm{out}}\in[r_{\mathrm{out}}]} \rho^f\left[O_{\mathrm{in}} = o_{\mathrm{in}}, O_{\mathrm{out}} = o_{\mathrm{out}}\right] = \rho^f\left[O_{\mathrm{in}} = o_{\mathrm{in}}, O_{\mathrm{out}} = I_{\mathrm{out}}^{-1}(f(I_{\mathrm{in}}(o_{\mathrm{in}})))\right] = 1 \,.$$

Thus, $\rho^f\left[O_{\mathrm{out}}, O_{\mathrm{in}}\right]$ is a directed tensor with variables O_{in} incoming and O_{out} outgoing.

"$\Leftarrow$": Conversely let there be a relation $\mathcal{R}$, such that $\rho^{\mathcal{R}}$ is directed. To this end, we observe that for any $o_{\mathrm{in}} \in [r_{\mathrm{in}}]$ the tensor

$$e_{\mathcal{R}}\left[O_{\mathrm{in}} = o_{\mathrm{in}}, O_{\mathrm{out}}\right]$$

is a boolean tensor with coordinate sum one and therefore a basis vector. It follows that the function $f : \mathcal{U}^{\mathrm{in}} \to \mathcal{U}^{\mathrm{out}}$ defined for $x \in \mathcal{U}^{\mathrm{in}}$ as

$$f(x) = I_{\mathrm{out}}(e^{-1}(e_{\mathcal{R}}\left[O_{\mathrm{in}} = I_{\mathrm{in}}(x), O_{\mathrm{out}}\right]))$$

is well-defined. We then have by construction

$$\begin{aligned}\rho^f[O_{\text{out}}, O_{\text{in}}] &= \sum_{o_{\text{in}}\in[r_{\text{in}}]} e_{f(o_{\text{in}})}[O_{\text{out}}] \otimes e_{o_{\text{in}}}[O_{\text{in}}] \\ &= \sum_{o_{\text{in}}\in[r_{\text{in}}]} e_{\mathcal{R}}[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}] \otimes e_{o_{\text{in}}}[O_{\text{in}}] \\ &= e_{\mathcal{R}}[O_{\text{out}}, O_{\text{in}}]\end{aligned}$$

and therefore by Def. 79 $\mathcal{R} = \mathcal{R}^f$. □

We are specially interested in sets of states of a factored system, which amounts to the case in Def. 14. 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. 79 has an obvious generalization to this case with multiple enumeration variables.

14.2.2 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 98 (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*

$$e_{I_{\text{out}}^{-1}(f(u))}[O_{\text{out}}] = \left\langle \rho^f[O_{\text{out}}, O_{\text{in}}], e_{I_{\text{in}}(u)}[O_{\text{in}}] \right\rangle [O_{\text{out}}] .$$

Thus, we can retrieve the function evaluation by the inverse one-hot mapping as

$$f(u) = e^{-1}(\left\langle \rho^f[O_{\text{out}}, O_{\text{in}}], e_{I_{\text{in}}(u)}[O_{\text{in}}] \right\rangle [O_{\text{out}}]) .$$

Proof. From the representation

$$\rho^f[O_{\text{out}}, O_{\text{in}}] = \sum_{o_{\text{in}}\in[r_{\text{in}}]} e_{(I_{\text{out}}^{-1}\circ f\circ I_{\text{in}})o_{\text{in}}}[O_{\text{in}}] \otimes e_{o_{\text{in}}}[O_{\text{in}}]$$

and the orthonormality of the one-hot encodings of the input enumeration we get

$$\left\langle \rho^f[O_{\text{out}}, O_{\text{in}}], e_{I_{\text{in}}(u)}[O_{\text{in}}] \right\rangle [O_{\text{out}}] = e_{I_{\text{out}}^{-1}(f(u))}[O_{\text{out}}] .$$

□

In comparison with the Coordinate Calculus scheme (see The. 89), 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.

14.3 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.

14.3.1 Composition of function

We have already used (see The. 35), 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 99 (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

$$\begin{aligned}
\left\langle \rho^f, \rho^g \right\rangle\left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] &= \sum_{x_{\mathcal{V}^1}\in\times_{v\in\mathcal{V}^1}[m_v]} \sum_{x_{\mathcal{V}^2}\in\times_{v\in\mathcal{V}^2}[m_v]} \Big\langle \left(e_{g(x_{\mathcal{V}^2})}\left[X_{\mathcal{V}^3}\right] \otimes e_{x_{\mathcal{V}^2}}\left[X_{\mathcal{V}^2}\right]\right), \\
&\qquad \left(e_{f(x_{\mathcal{V}^1})}\left[X_{\mathcal{V}^2}\right] \otimes e_{x_{\mathcal{V}^1}}\left[X_{\mathcal{V}^1}\right]\right) \Big\rangle\left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] \\
&= \sum_{x_{\mathcal{V}^1}\in\times_{v\in\mathcal{V}^1}[m_v]} \delta_{x_{\mathcal{V}^2}, x_{\mathcal{V}^1}} \cdot e_{g(x_{\mathcal{V}^2})}\left[X_{\mathcal{V}^3}\right] \otimes e_{x_{\mathcal{V}^1}}\left[X_{\mathcal{V}^1}\right] \\
&= \sum_{x_{\mathcal{V}^1}\in\times_{v\in\mathcal{V}^1}[m_v]} e_{(g\circ f)(x_{\mathcal{V}^1})}\left[X_{\mathcal{V}^3}\right] \otimes e_{x_{\mathcal{V}^1}}\left[X_{\mathcal{V}^1}\right] \\
&= \rho^{g\circ f}\left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] ,
\end{aligned}$$

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. 99 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. 35 to efficiently represent propositional formulas, for which syntactical expressions are given.

14.3.2 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. 13.2) 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 100. *The coordinatewise transform of any tensor T (see Def. 73) by a real function h is the contraction (see Figure 33)*

$$h(T)[X_{[d]}] = \left\langle \rho^T \left[O_T, X_{[d]}\right], h|_{\mathrm{im}(T)} \left[O_T\right] \right\rangle \left[X_{[d]}\right] \, .$$

Proof. By the basis calculus The. 98 we have for any state $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$, that

$$\begin{aligned} \left\langle \rho^T \left[O_T, X_{[d]}\right], h|_{\mathrm{im}(T)} \left[O_T\right] \right\rangle \left[X_{[d]} = x_{[d]}\right] &= \left\langle \rho^T \left[O_T, X_{[d]} = x_{[d]}\right], h|_{\mathrm{im}(T)} \left[O_T\right] \right\rangle [\varnothing] \\ &= \left\langle e_{I_{T\left[X_{[d]}=x_{[d]}\right]}} \left[O_T\right], h|_{\mathrm{im}(T)} \left[O_T\right] \right\rangle [\varnothing] \\ &= h(T)[X_{[d]} = x_{[d]}] \, . \end{aligned}$$

Since both tensors coincide on all coordinates, they are equal. □

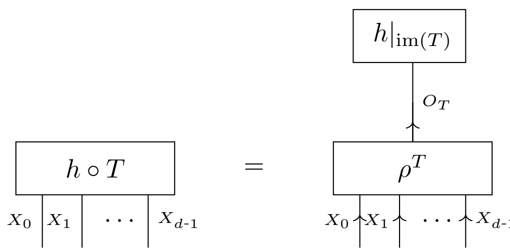


Figure 33: Representation of the composition of a tensor T with a real function h.

Corollary 10. *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. 100 using $h = \mathrm{Id}$ and by noticing that

$$T\left[X_{[d]}\right] = \mathrm{Id}(T)[X_{[d]}] \, .$$

□

Corollary 11. *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. 100 using $h = \mathbb{I}$ and by noticing that

$$\mathbb{I}\left[X_{[d]}\right] = \mathbb{I}(T)[X_{[d]}] \, .$$

□

14.3.3 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 101. *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. 101 in The. 116 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 27), and for exponential families, which statistics have images, which are included in cartesian products of the images to each coordinate (see Example 28).

14.4 Selection Encodings

Selection encodings as introduced in Def. 15 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.

14.4.1 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 $\times_{k\in[d]}[m_k]$ 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 $\times_{k\in[d]}[m_k]$ are an intermediate representation.

Definition 80. *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*

$$\beta^f [O_1, O_2] \in \mathbb{R}^{r_1} \otimes \mathbb{R}^{r_2}$$

defined for $o_1 \in [r_1]$ and $o_2 \in [r_2]$ by

$$\beta^F [O_1 = o_1, O_2 = o_2] = \left\langle F^{I_1(o_1)}, I_2(o_2) \right\rangle [\varnothing] .$$

Basis encodings for compositions of linear functions can be computed via contractions of the respective basis encodings, as we show next.

Theorem 102. *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*

$$\beta^{F^2 \circ F^1} [O_1, O_3] = \left\langle \beta^{F^2} [O_2, O_3] , \beta^{F^1} [O_1, O_2] \right\rangle [O_1, O_3] .$$

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

$$\left\langle F^2 \circ F^1(I_1(o_1)), I_3(o_3) \right\rangle [=] \left\langle F^2 \left(\sum_{o_2 \in [r_2]} \beta^{F^1}\left[O_1 = o_1, O_2 = o_2\right] \cdot I_2(o_2) \right), I_3(o_3) \right\rangle [=] \sum_{o_2 \in [r_2]} \left\langle F^2 \left(\beta^{F^1}\left[O_1 = o_1, O_2 = o_2\right] \cdot I \right.\right.$$

Therefore, both tensors are equivalent. □

For basis encodings we thus have a similar composition theorem as for relational encodings of arbitrary functions (see The. 99). What is more, one can understand each relational encodings as a basis encoding of a linear function. Along this line, the composition theorem The. 102 as the principle of linear algebra, which underlies The. 99. A typical interpretation of The. 102 is matrix multiplication, where matrices understood since matrices are basis encodings of linear maps.

Example 21 (Relational encodings as basis encodings)**.** *Let us justify that we refered to the contractions of relational encodings by basis calculus, by describing that relational encodings are a special case of basis encodings. To that end, we understand the sets $\mathcal{U}^{\text{in}} = [r_{\text{in}}]$ and $\mathcal{U}^{\text{out}} = [r_{\text{out}}]$ as labels of a basis in $\mathbb{R}^{r_{\text{in}}}$ and $\mathbb{R}^{r_{\text{out}}}$. Then, given a relation $\mathcal{R} \subset \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{out}}$, we define a linear map $F : \mathbb{R}^{r_{\text{in}}} \to \mathbb{R}^{r_{\text{out}}}$ through the action on the $i \in [r_{\text{in}}]$-th basis element as*

$$F(e_i) = \sum_{j \in [r_{\text{out}}] \,:\, (i,j) \in \mathcal{R}} e_j \, .$$

Comparing the coefficients of the basis encoding of F and the relational encoding $\mathcal{R}$ we get

$$\beta^{F}\left[O_{\text{out}} = o_{\text{out}}, O_{\text{in}} = o_{\text{in}}\right] = e_{\mathcal{R}}\left[O_{\text{out}} = o_{\text{out}}, O_{\text{in}} = o_{\text{in}}\right] \, .$$

14.4.2 Selection encodings as basis encodings

Selection encodings (see Def. 15) are related to basis encodings of linear maps as we show in the next theorem.

Theorem 103. *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

$$F^f(e_{x_{[d]}}) \coloneqq f(x_{[d]})$$

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

$$\gamma^f\left[X_{[d]} = x_{[d]}, L_0, \dots, L_{n-1}\right] = f(x_{[d]})[L_{[n]}] = F^f(e_{x_{[d]}})[L_{[n]}] \, .$$

We further have

$$F^f(e_{x_{[d]}})[L_{[n]}] = \sum_{l_{[n]}} \left\langle F^f(e_{x_{[d]}})[L_{[n]} = l_{[n]}], e_{l_{[n]}}\left[L_{[n]}\right] \right\rangle [\varnothing] \cdot e_{l_{[n]}}\left[L_{[n]}\right] = \sum_{l_{[n]}} \beta^{F^f}\left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]}\right] \cdot e_{l_{[n]}}\left[L_{[n]}\right] = \beta^{F^f}\left[X\right.$$

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. 15 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 104. *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*

$$f : \bigtimes_{v\in A}[m_v] \rightarrow \bigotimes_{v\in B}\mathbb{R}^{m_v}$$

defined for $x_A \in \bigtimes_{v\in A}[m_v]$ as

$$f(x_A) \coloneqq T[X_A = x_A, X_B]$$

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

$$\gamma^f [X_A, X_B = x_B] = \sum_{x_A\in\bigtimes_{v\in A}[m_v]} e_{x_A}[X_A] \otimes f(x_A)[X_B = x_B] = \sum_{x_A\in\bigtimes_{v\in A}[m_v]} e_{x_A}[X_A] \otimes T[X_A = x_A, X_B = x_B] = T[X_A, X_B$$

and the equivalence follows. □

Example 22 (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. 104. We can construct $\mathcal{H}^T$ by first building the selection encoding to $\mathcal{H}$ and then applying the construction of The. 104 with $A = L$ and $B = X_{[d]}$.*

We use selection encodings to represent weighted sums of functions, based on the next theorem.

Theorem 105 (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 $\theta[L]$ be a tensor. Then*

$$\left(\sum_{l\in[p]}\theta[L=l]\cdot\phi_l\right)[X_{[d]}] : \bigtimes_{k\in[d]}[m_k] \rightarrow \mathbb{R}$$

is represented as

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

Proof. The representation holds, since for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have

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

□

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 105 as basis decompositions of linear maps.

14.5 Value subspaces of functions

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 81. *Given any function $f:\mathcal{U}^{\mathrm{in}}\to\mathcal{U}^{\mathrm{out}}$*

$$V^{f}=\left\{\left\langle\rho^{f}\left[O_{\mathrm{out}},O_{\mathrm{in}}\right],W\left[O_{\mathrm{out}}\right]\right\rangle\left[O_{\mathrm{out}},O_{\mathrm{in}}\right]\,:\,W\left[O_{\mathrm{out}}\right]\in\mathbb{R}^{r_{\mathrm{out}}}\right\}$$

We always have $\mathbb{I}\left[X_{[d]}\right]\in V^{f}$, when choosing $W=\mathbb{I}\left[Y_{f}\right]$.

A value subspace of its function

$$V^{f}=\mathrm{span}\left(\mathbb{I}^{f=y}\left[O_{\mathrm{in}}\right]\,:\,y\in\mathrm{im}\left(f\right)\right)$$

We further define contractions of subspaces.

Definition 82. *Given two subspaces V^1,V^2 of tensors with variables $X_{\mathcal{V}^1}$, $X_{\mathcal{V}^2}$, their contraction is*

$$\left\langle V^1,V^2\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]=\left\{\left\langle T^1\left[\mathcal{V}^1\right],T^2\left[\mathcal{V}^2\right]\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\,:\,T^1\left[\mathcal{V}^1\right]\in V^1,\,T^2\left[\mathcal{V}^1\right]\in V^2\right\}\,.$$

Importantly, the contraction of two subspaces is in general not a subspace.

14.5.1 Cartesian products of functions

Since always $\mathbb{I}\left[X_{[d]}\right]\in V^{f}$, we have

$$\left\langle V^1\right\rangle\left[\subset\right]\left\langle V^1,V^f\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]$$

We define the cartesian product of functions on the same input set $\mathcal{U}^{\mathrm{in}}$ and output sets $\mathcal{U}^{1,\mathrm{out}},\mathcal{U}^{2,\mathrm{out}}$ as the function

$$(f,g):\mathcal{U}^{\mathrm{in}}\to\mathcal{U}^{1,\mathrm{out}}\times\mathcal{U}^{2,\mathrm{out}}$$

with

$$(f,g)(z)=\left(f(z),g(z)\right).$$

For its subspace we have

$$V^{(f,g)}=\mathrm{span}\left(\left\langle V^{f},V^{g}\right\rangle\left[X_{[d]}\right]\right)\,.$$

We now apply this developments in the characterization of distributions realizable by an exponential family.

Example 23 (Subspace of statistics)**.** *The realizable distributions defined in Def. 30 given a statistic function are then*

$$\Lambda^{\phi,\mathcal{G}^{\mathrm{EL}}}=\left\langle V^{\phi_l}\,:\,l\in[p]\right\rangle\left[X_{[d]}\right]\,.$$

The set of realizable distributions therefore increases with each added coordinate of the statistic.

Conversely, taking the ϕ as the function to be represented, we have

$$\Lambda^{\phi,\mathcal{G}^{\max}}=V^{\phi}=\mathrm{span}\left(\Lambda^{\phi,\mathcal{G}^{\mathrm{EL}}}\right)\,.$$

14.5.2 Composition of functions

Composition of functions are then understood as contractions of their corresponding subspaces. The subspace of a composition satisfied an inclusion relation, which is the opposite of those for cartesian products, that is

$$V^{f\circ g}\subset V^{(f,g)}\,. \tag{45}$$

Example 24 (Propositional Formulas)**.** *Each formula defines by its relational encoding the subspace of $\bigotimes_{k\in[d]}\mathbb{R}^2$*

$$V^{f}=\mathrm{span}\left(\neg f\left[X_{[d]}\right],f\left[X_{[d]}\right]\right)\,.$$

For composition of formulas f,h with a connective $\circ$ acting on their images we have

$$V^{f\circ h}\subset V^{(f,h)}\,.$$

A connective $\circ$ can thus be understood as a selection of a two-dimensional subspace in the four-dimensional subspace of the cartesian product of the connected formulas. The contraction of atomic subspaces further span the space

$$\bigotimes_{k\in[d]}\mathbb{R}^2=\mathrm{span}\left(\left\langle V^{X_k}\,:\,k\in[d]\right\rangle\left[X_{[d]}\right]\right)\,.$$

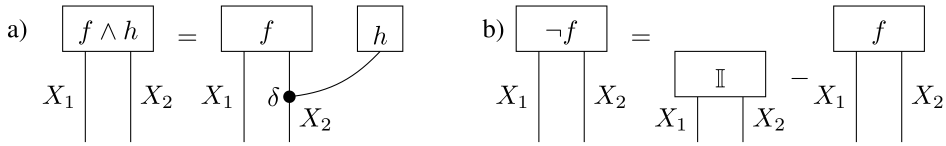


Figure 34: Decomposition schemes by effective calculus, using coordinatewise transforms of tensors (see Def. 73). a) Conjunction performed by coordinatewise multiplications, b) Negations performed by coordinatewise substraction from one.

14.6 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. 73). 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 Tsilionis et al. (2024) in batchwise logical inference.

Theorem 106. *For any formulas f, h we have*

$$\left\langle \rho^{\wedge}\left[Y_{f\wedge h}, X_f, X_h\right], e_1\left[Y_{f\wedge h}\right]\right\rangle \left[X_f, X_h\right] = e_1\left[X_f\right] \otimes e_1\left[X_h\right] .$$

In particular, it holds that (see Figure 34a)

$$(f \wedge h)[X_{[d]}] = \langle f, h\rangle \left[X_{[d]}\right] .$$

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

$$\begin{aligned}(f \wedge h)[X_{[d]}] &= \left\langle \rho^{f}\left[X_f, X_{[d]}\right], \rho^{h}\left[X_h, X_{[d]}\right], \rho^{\wedge}\left[Y_{f\wedge h}, X_f, X_h\right], e_1\left[Y_{f\wedge h}\right]\right\rangle \left[X_{[d]}\right] \\ &= \left\langle \rho^{f}\left[X_f, X_{[d]}\right], \rho^{h}\left[X_h, X_{[d]}\right], \left(e_1\left[X_f\right] \otimes e_1\left[X_h\right]\right)\right\rangle \left[X_{[d]}\right] \\ &= \langle f, h\rangle \left[X_{[d]}\right] .\end{aligned}$$

□

A similar decomposition holds for negations, as we show next.

Theorem 107. *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 34b)

$$(\neg f)[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - f\left[X_{[d]}\right] .$$

Proof. Using that for two dimensional variables we have $\mathbb{I}\left[X\right] = e_0\left[X\right] + e_1\left[X\right]$. □

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 15. In Figure 35 we provide an example of this decomposition.

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.

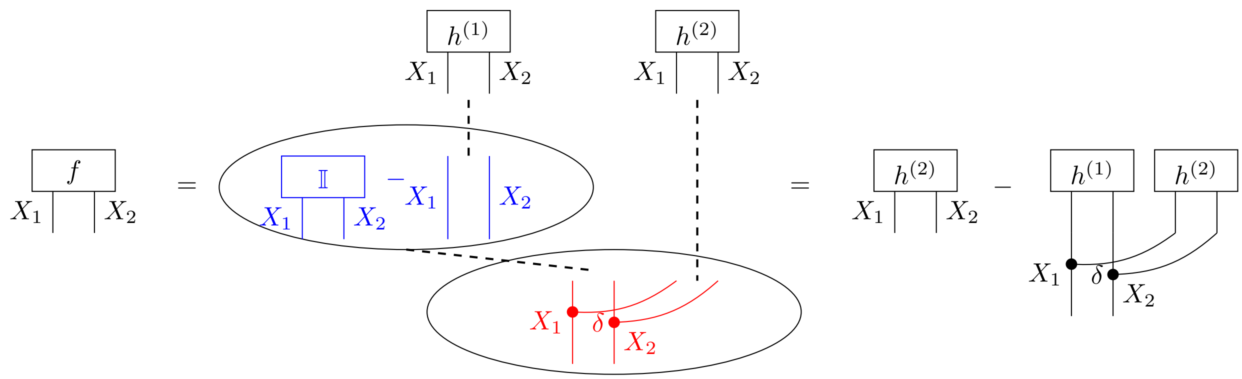


Figure 35: Example of a decomposition by effective calculus of a formula $f(X_1, X_2) = \neg h^{(1)}(X_1, X_2) \wedge h^{(2)}(X_1, X_2)$ into a sum of contractions.

14.7 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 98, 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.

15 Sparse Calculus

We in this chapter develop sparse tensor representation formats based on constrained CP formats. Our motivation for these formats result from the connection to encoding mechanisms, which we have applied in Part I and Part II, and to sparse optimization formats.

15.1 CP Decomposition

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 matrix $M[X_0, X_1]$, we enumerate its singular values by I taking values in $[n]$ and store them in a vector $\sigma[I]$. With the corresponding singular vectors by $V^0[X_0, I]$ and $V^1[X_1, I]$, the singular value decomposition of M is

$$M[X_0, X_1] = \sum_{i\in[n]} \sigma[I = i] \cdot V^0[X_0, I = i] \otimes V^1[X_1, I = i] \ .$$

Here the smallest n such that this decomposition exists, is the matrix rank $\mathrm{rank}(M)$. In contraction notation we abbreviate this to

$$M[X_0, X_1] = \left\langle \sigma[I], V^0[X_0, I], V^1[X_1, I] \right\rangle [X_0, X_1] \ .$$

Given a tensor of higher order, a generalization of this decomposition is a tensor product over multiple vectors, as we define next.

Definition 83. *A* CP *decomposition of size n of a tensor $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ is a collections of a scalar core $\sigma[I]$ and leg cores $V^k[I, X_k]$ for $k \in [d]$, where I is an enumeration variable taking values in $[n]$, such that*

$$T\left[X_{[d]}\right] = \left\langle \{\sigma[I]\} \cup \{V^k[X_k, I] \,:\, k \in [d]\} \right\rangle \left[X_{[d]}\right] \ .$$

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.*
- *boolean, when for each k the core V^k is boolean.*
- *basis, where we demand both properties, that is for each $k \in [d]$ and $i \in [n]$*
$$V^k [X_k, I = i] \in \{e_{[x_k]} [X_k] \, x_k \in [m_k]\} .$$
- *basis+, when for each $k \in [d]$ and $i \in [n]$*
$$V^k [X_k, I = i] \in \{e_{[x_k]} [X_k] \, x_k \in [m_k]\} \cup \{\mathbb{I} [X_k]\} .$$

We denote by $\mathrm{rank}(T)$*, respectively* $\mathrm{rank}^{\mathrm{bin}}(T)$, $\mathrm{rank}^{\mathrm{bas}}(T)$ *and* $\mathrm{rank}^{\mathrm{bas+}}(T)$ *the minimal cardinality such that T has a* CP *Decomposition, respectively with directed cores, boolean cores, basis cores and basis+ cores.*

All ranks have a naive bound by the space dimension, which is obvious from the coordinate decomposition (see Chapter 13)

$$T \left[X_{[d]}\right] = \sum_{x_{[d]} \in \times_{k\in[d]}[m_k]} T \left[X_{[d]} = x_{[d]}\right] \cdot \bigotimes_{k\in[d]} e_k [X_k] \ .$$

If we construct i as an enumeration of the coordinates in $\times_{k\in[d]}[m_k]$, that is $n = \prod_{k\in[d]} m_k$, this is a CP decomposition, which is basis and therefore also directed, boolean and basis+.

CP decomposition as a tensor network format come with some drawbacks. The set of tensors with a fixed rank are not closed Beylkin and Mohlenkamp (2005) and approximation problems are often ill posed de Silva and Lim (2008). Since as a consequence their numerical treatment comes with many problems Espig et al. (2012), alternative formats have gained popularity. Common formats are the TUCKER-format originally introduced in Hitchcock (1927), and often refered to as higher-order singular value decomposition, and the more recently developed TT and HT decomposition formats (see Chapter 1). Given a HT the best approximation of a tensor always exists (Theorem 11.58 in Hackbusch (2012)).

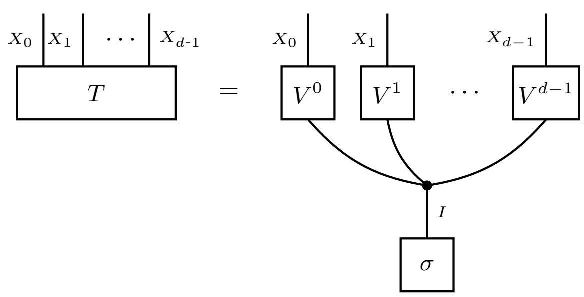


Figure 36: Tensor Network diagram of a generic CP decomposition (see Def. 83)

15.1.1 Directed Leg Cores

The contraint of directionality of the leg cores does not influence decomposablitiy of a tensor, as we show next.

Lemma 25. *For any tensor $T\left[X_{[d]}\right]$ we have*

$$\mathrm{rank}(T) = \mathrm{rank}^{\mathrm{dir}}(T) \ .$$

Proof. Let there be a CP decomposition of T by

$$T \left[X_{[d]}\right] = \left\langle \{\sigma[I]\} \cup \{V^k [X_k, I] \, : \, k \in [d]\} \right\rangle \left[X_{[d]}\right] \ .$$

We then transform the scalar core to another core $\tilde{\sigma}[I]$ with coordinates to $i \in [n]$ by

$$\tilde{\sigma}[I = i] = \sigma[I = i] \cdot \prod_{k\in[d]} \left\langle V^k [X_k, I = i] \right\rangle [\varnothing] \ .$$

It follows for any $i \in [n]$, that

$$\sigma[I=i] \cdot \bigotimes_{k\in[d]} V^k\left[X_k, I=i\right] = \tilde{\sigma}[I=i] \cdot \bigotimes_{k\in[d]} \left\langle V^k\left[X_k, I=i\right]\right\rangle\left[X_k|\varnothing\right]$$

and thus

$$T\left[X_{[d]}\right] = \left\langle \{\tilde{\sigma}[I] \cup \{\left\langle V^k\left[X_k, I\right]\right\rangle\left[X_k, I|\varnothing\right] \,:\, k \in [d]\}\right\rangle\left[X_{[d]}\right] \,.$$

We have thus constructed a directed CP decomposition of same size n to an arbitrary CP decomposition and conclude that $\mathrm{rank}\left(T\right) = \mathrm{rank}^{\mathrm{dir}}\left(T\right)$. □

15.1.2 Basis CP decompositions and the ℓ_0-norm

The slices of directed and boolean tensors with respect to incoming variables are basis tensors. We have thus called CP decomposition with the restiction of directed an boolean leg vectors basis CP decomposition. Based on this intuition, we can interpret basis CP decomposition by mappings to non-vanishing coordinates of the decomposed tensor. To start, let us define the number of nonzero coordinates of tensors by the ℓ_0-norm.

Definition 84. *The ℓ_0-norm counts the nonzero coordinates of a tensor by*

$$\ell_0\left(T\right) = \#\left\{x_0, \ldots, x_{d-1} \,:\, T_{x_0,\ldots,x_{d-1}} \neq 0\right\}.$$

The ℓ_0-norm is not a norm, but at each tensor the limit of ℓ_p-norms (which are norms for $p \geq 1$) for $p \to 0$.

The ℓ_0 norm is the number of non-vanishing coordinates of a tensor. 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 14, where we characterized boolean and directed cores by the encoding of associated functions. Based on this idea, we can proof, that any tensor has a directed and boolean CP decomposition with rand $\ell_0\left(T\right)$.

Theorem 108. *For any tensor $T\left[X_{[d]}\right]$ we have*

$$\mathrm{rank}^{\mathrm{bas}}\left(T\right) = \ell_0\left(T\right)\,.$$

Proof. Let us first show, that $\mathrm{rank}^{\mathrm{bas}}\left(T\right) \leq \ell_0\left(T\right)$. We find a map

$$D : [\ell_0\left(T\right)] \to \bigtimes_{k\in[d]} [m_k]$$

which image is the set of non-vanishing coordinates of $T\left[X_{[d]}\right]$. Denoting its image coordinate maps by D_k we have

$$T\left[X_{[d]}\right] = \sum_{j\in[m]} \sigma[D(()j)] \left(\bigotimes_{k\in[d]} e_{D_k(j)}\left[X_k\right]\right)\,.$$

This is a basis CP decomposition of size $\ell_0\left(T\right)$ and we thus have $\mathrm{rank}^{\mathrm{bas}}\left(T\right) \leq \ell_0\left(T\right)$.

Conversely, let us show $\mathrm{rank}^{\mathrm{bas}}\left(T\right) \geq \ell_0\left(T\right)$. Any basis CP decomposition of T with size r would has at most r coordinates different from zero and thus $\ell_0\left(T\right) \leq r$. Thus, there cannot be a CP decomposition with a dimension $r \leq \ell_0\left(T\right)$. □

The next theorem relates the basis CP decomposition with encodings of d-ary relations (see Def. 78).

Theorem 109. *Any boolean tensor $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ is the encoding of a d-ary relation $\mathcal{R} \subset \bigtimes_{k\in[d]}[m_k]$ with cardinality*

$$|\mathcal{R}| = \mathrm{rank}^{\mathrm{bas}}\left(T\right)\,.$$

Proof. We fine a basis CP decomposition of $T\left[X_{[d]}\right]$ with $n = \mathrm{rank}^{\mathrm{bas}}\left(T\right)$. Since $T\left[X_{[d]}\right]$ is boolean, and since each i labels a disjoint non-vanishing coordinate (see proof of The. 108), the decomposition has a trivial scalar core $\sigma[I] = \mathbb{I}\left[I\right]$. It follows, that

$$T\left[X_{[d]}\right] = \sum_{i\in[n]} \left(\bigotimes_{k\in[d]} V^k\left[X_k, I=i\right]\right)$$

Since the CP decomposition is basis, the slice $V^k [X_k, I = i]$ is for any $k \in [d]$ and $i \in [n]$ a basis vector. We then define

$$x_k^i = e^{-1}(V^k [X_k, I = i])$$

and notice, that for the relation

$$\mathcal{R} = \{x_{[d]}^i \,:\, i \in [n]\} \subset \bigtimes_{k \in [d]} [m_k]$$

we have

$$e_{\mathcal{R}} \left[X_{[d]}\right] = \sum_{i \in [n]} \left(\bigotimes_{k \in [d]} V^k [X_k, I = i] \right) .$$

This coincides with the above CP decomposition of $T \left[X_{[d]}\right]$ and the claim is established. □

Remark 25 (Matrix Storage of basis CP decompositions)**.** *The storage demand of any* CP *decomposition is at most linear in the size 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* $e_\cdot$*, 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 define a leg selecting variable* L *taking values in* $[d+1]$ *and store a basis* CP *decomposition of size* n *by the matrix*

$$M[I, L] \in \mathbb{R}^{m \times (d+1)}$$

defined for $i \in [n]$ *and* $k \in [d]$ *by*

$$M[I = i, L = k] = \begin{cases} \sigma[I = i] & \textit{if} \quad k = d \\ e^{-1}(V^k [X_k, I = i]) & \textit{else} \end{cases} .$$

This is a common trick to store relational databases.

15.1.3 Basis+ CP decompositions and polynomials

The basis+ CP decompositions are closely related to monomial decompositions of a tensor, which we will define next.

Definition 85. *A monomial decomposition of a tensor* $T \left[X_{[d]}\right] \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ *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*

$$T \left[X_{[d]}\right] = \sum_{(\lambda, A, x_A) \in \mathcal{M}} \lambda \cdot \langle e_{x_A} [X_A] \rangle \left[X_{[d]}\right] . \tag{46}$$

For any tensor $T \left[X_{[d]}\right] \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ *we define its polynomial sparsity of order* r *as*

$$\operatorname{rank}^r (T) = \min \left\{ |\mathcal{M}| \,:\, T \left[X_{[d]}\right] = \sum_{(\lambda, A, x_A) \in \mathcal{M}} \lambda \cdot \langle e_{x_A} [X_A] \rangle \left[X_{[d]}\right] \,,\, \forall_{(\lambda, A, x_A) \in \mathcal{M}} \, |A| \leq r \,. \right\}$$

We refer to the terms in a decomposition (46) in Def. 85 as monomials of boolean features, which are enumerated by pairs (k, x_k) and indicate whether the variable X_k is in state $x_k \in [m_k]$. Each such boolean features is represented by the indicator

$$\mathbb{I}_{X_k = x_k} [X_k] = e_{x_k} [X_k] .$$

The monomial of multiple such boolean features indicates, whether all variables labelled by a set A are in the state X_A. We have

$$\mathbb{I}_{\forall k \in A \colon X_k = x_k} [X_A] = e_{x_A} [X_A] = \bigotimes_{k \in A} e_{x_k} [X_k] .$$

The states of the variables labeled by $k \in [d]/A$ are not specified in the monomial and the indicators are trivially extended to

$$\langle e_{x_A} [X_A] \rangle \left[X_{[d]}\right] = e_{x_A} [X_A] \otimes \mathbb{I} \left[X_{[d]/A}\right] .$$

Since we are working with boolean features, there is no need to consider higher-order powers of individual features, since for any $n \in \mathbb{N}$, $n \geq 1$ and any boolean value $z \in \{0,1\}$ we have $z^n = z$.

For some monomial orders $r < d$ there are tensors $T\left[X_{[d]}\right]$, which do not have a monomial decomposition of order r. In that case the minimum is over an empty set and we define $\mathrm{rank}^r(T) = \infty$. We characterize in the next theorem the set of tensors with monomial decompositions of order r.

Theorem 110. *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. The set of tensors admitting a monomial decomposition of order r is closed under addition and scalar multiplication. Specifically, the sum of two such tensors retains a monomial decomposition, formed by concatenating their respective decompositions. Scalar multiplication can be performed by a rescaling of each scalar λ and therefore preserves the decomposition structure. Hence, these tensors form a linear subspace.

To bound the dimension of this subspace, we consider tensors of the form $\langle e_{x_A} \rangle \left[X_{[d]}\right]$. The number of such tensors is given by

$$\sum_{s \in [r]} m^s \binom{d}{s}.$$

Since any tensor with a monomial decomposition is a weighted sum of those, this provides an upper bound on the dimension.

We notice, that the set of slices is in general not linear independent, and therefore forms a frame instead of a linear basis Casazza et al. (2013). The number of elements in the frame is therefore in general a loose upper bound on the dimension. □

The. 110 states, that the tensors admitting 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 $\mathrm{rank}^r(T)$ furthermore coincides with the basis+ CP rank $\mathrm{rank}^{\mathrm{bas+}}(T)$.

Theorem 111. *For any tensor $T\left[X_{[d]}\right] \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ we have*

$$\mathrm{rank}^d(T) = \mathrm{rank}^{\mathrm{bas+}}(T).$$

In case of two-dimensional legs, that is $m_k = 2$ for all $k \in [d]$, we also have

$$\mathrm{rank}^{\mathrm{bin}}(T) = \mathrm{rank}^d(T).$$

Proof. To proof the first claim, we construct a basis+ CP decomposition given a monomial decomposition and vice versa. To show $\mathrm{rank}^d(T) \geq \mathrm{rank}^{\mathrm{bas+}}(T)$, let there be an arbitrary 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

$$\begin{aligned}
T\left[X_{[d]}\right] &= \sum_{i\in[n]} \lambda^i \cdot \left\langle e_{x_A^i} \right\rangle \left[X_{[d]}\right] \\
&= \sum_{i\in[n]} \left(\sigma[I=i] \cdot \bigotimes_{k\in[d]} V^k\left[I=i, X_k\right] \right) \\
&= \left\langle \{\sigma[I]\} \cup \{V^k\left[I, X_k\right] \,:\, k\in[d]\} \right\rangle \left[X_{[d]}\right] \,.
\end{aligned}$$

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

$$\mathrm{rank}^d\left(T\right) \geq \mathrm{rank}^{\mathrm{bas+}}\left(T\right) \,.$$

To show $\mathrm{rank}^d\left(T\right) \leq \mathrm{rank}^{\mathrm{bas+}}\left(T\right)$, let there now be a basis+ CP decomposition of an arbitrary $T\left[X_{[d]}\right]$. We define for each $i\in[n]$

$$A^i = \{k\in[d] : V^k\left[I=i, X_k\right] \neq \mathbb{I}\left[X_k\right]\} \quad \text{and} \quad x_A^i = \{e^{-1}(V^k\left[I=i, X_k\right]) \,:\, k\in A\}$$

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

$$\mathcal{M} = \{(\sigma[I=i], A^i, x^i_{A^i}) \,:\, i\in[n]\} \,.$$

It follows from this that

$$\mathrm{rank}^d\left(T\right) \leq \mathrm{rank}^{\mathrm{bas+}}\left(T\right)$$

and the first claim is shown.

The second claim follows from the observation, that the set of non-vanishing boolean vectors coincides with the set of one-hot encodings extended by the trivial vector. Thus, a CP decomposition with non-vanishing slices is boolean if and only if it is basis+. This establishes, that both ranks are equal, since a CP decomposition of minimal rank cannot contain non-vanishing slices. ☐

Remark 26 (Sparse representation of propositional formulas)**.** *When all leg dimensions of a boolean tensor T are* 2*, we can further interpret T as a logical formula. We can use the boolean* CP *decomposition of any tensor $\tilde{T}$ with $\mathbb{I}_{\neq 0}\left(\tilde{T}\right) = T$ as a CNF of T. Finding the sparsest CNF thus amounts to finding the $\tilde{T}$ with minimal $\mathrm{rank}^d\left(\tilde{T}\right)$ such that $\mathbb{I}_{\neq 0}\left(\tilde{T}\right) = T$.*

Remark 27 (Matrix Storage of basis+ CP decompositions)**.** *We can adapt the storage format of Remark 25 from basis to basis+* CP *decompositions. To this end, let there be a basis+* CP *decomposition of a tensor with scalar core $\sigma[I]$ and leg cores $\{V^k\left[X_k, I\right] \,:\, k\in[d]\}$. We use a value $z\in\mathbb{R}/\mathrm{im}\left(\sigma\right)$ distinguished from the coordinates of the scalar core and define a matrix*

$$M^z\left[I, L\right]$$

where L takes values in $[d]$, coordinatewise as

$$M^z\left[I=i, L=k\right] = \begin{cases} \sigma[I=i] & \textit{if} \quad k=d \\ z & \textit{if} \quad V^k\left[X_k, I=i\right] = \mathbb{I}\left[X_k\right] \\ e^{-1}(V^k\left[X_k, I=i\right]) & \textit{else} \end{cases} \,.$$

15.2 Constructive Bounds on CP Ranks

After having defined different CP decompositions, let us investigate bounds on their ranks, which proofs rely on explicit core constructions.

15.2.1 Cascade of ranks

We start by showing a cascade of bounds of CP ranks, when demanding different leg restrictions as in Def. 83.

Theorem 112. *For any tensor $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we have*

$$\mathrm{rank}\left(T\right) = \mathrm{rank}^{\mathrm{dir}}\left(T\right) \leq \mathrm{rank}^{\mathrm{bin}}\left(T\right) \leq \mathrm{rank}^{\mathrm{bas+}}\left(T\right) \leq \mathrm{rank}^{\mathrm{bas}}\left(T\right) \,.$$

Proof. The equality $\mathrm{rank}\left(T\right) = \mathrm{rank}^{\mathrm{dir}}\left(T\right)$ has been established in Lem. 25. The further inequalities follow by consecutive subset relations of the set of allowed leg slices in the respective CP decompositions. These imply, that any basis CP decomposition of a tensor $T\left[X_{[d]}\right]$ is also a basis+ CP decomposition, further that any basis+ CP decomposition is also a boolean CP decomposition and that any boolean CP decomposition is trivially an unrestricted CP decomposition. Thus, the ranks are minima of enlarging sets and the claimed rank cascade is established. ☐

Let us notice, that the stated bounds are not tight in general. To give an example, let us consider the tensor $T\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$, for which we have

$$\mathrm{rank}^{\mathrm{bas}}\left(T\right) = \ell_0\left(T\right) = \prod_{k\in[d]} m_k \,.$$

Since in the other restricted CP formats we can choose trivial slices to the leg cores, we have

$$\mathrm{rank}^{\mathrm{bas+}}\left(T\right) = 1 = \mathrm{rank}^{\mathrm{bin}}\left(T\right) = \mathrm{rank}^{\mathrm{dir}}\left(T\right) = \mathrm{rank}\left(T\right) \,.$$

The trivial tensor serves thus as an example, where the demand of the the storage format in Remark 25 has an exponential overhead compared to the storage format in Remark 27.

15.2.2 Operations on CP decompositions

When using CP decompositons of tensors in practicle applications, such as those investigated in Part I and Part II, we have to perform numerical manipulations in the form of summations, contractions and normations of the represented tensors. Let us here investigate, how these operations influence the decomposition.

Summation We start with the sum of tensors in a CP decomposition, which can be captured by a concatenation of the slices.

Theorem 113. *For any collections of tensors $\{T^l\left[X_{\mathcal{V}}\right] : l \in [p]\}$ with identical variables and scalars $\lambda^l \in \mathbb{R}$ for $l \in [p]$ we have*

$$\mathrm{rank}\left(\sum_{l\in[p]} \lambda^l \cdot T^l\right) \leq \sum_{l\in[p]} \mathrm{rank}\left(T^l\right) \,.$$

The bound still holds, when we replace on both sides $\mathrm{rank}\left(\cdot\right)$ by $\mathrm{rank}^{\mathrm{bin}}\left(\cdot\right)$, by $\mathrm{rank}^{\mathrm{bas}}\left(\cdot\right)$ or by $\mathrm{rank}^{\mathrm{bas+}}\left(\cdot\right)$.

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. ☐

Let us notice, that the upper bound is loose in many applications. For example, if two slice tuples of two decomposed tensors agree on x_A, A, then their sum can be performed by a sum of the corresponding scalar.

Contraction We continue to show rank bounds for arbitrary contractions by the product of the ranks of contracted tensors.

Theorem 114. *For any tensor network $\mathcal{T}^{\mathcal{G}}\left[X_{\mathcal{V}}\right]$ on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we have for any subset $\tilde{\mathcal{V}} \subset \mathcal{V}$*

$$\mathrm{rank}\left(\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\tilde{\mathcal{V}}}\right]\right) \leq \prod_{e\in\mathcal{E}\,:\,\tilde{\mathcal{V}}\cap e \neq \varnothing} \mathrm{rank}\left(T^e\right) \,.$$

The bound still holds, when we replace on both sides $\mathrm{rank}\left(\cdot\right)$ by $\mathrm{rank}^{\mathrm{bin}}\left(\cdot\right)$, by $\mathrm{rank}^{\mathrm{bas}}\left(\cdot\right)$ or by $\mathrm{rank}^{\mathrm{bas+}}\left(\cdot\right)$.

Remarkably, in The. 114 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 26. *For any tensors $T^1[X_{\mathcal{V}^1}]$ and $T^2[X_{\mathcal{V}^2}]$ and any set of variables $\tilde{\mathcal{V}} \subset \mathcal{V}^1 \cup \mathcal{V}^2$ we have*

$$\operatorname{rank}\left(\left\langle T^1, T^2 \right\rangle [X_{\tilde{\mathcal{V}}}]\right) \leq \operatorname{rank}\left(T^1\right) \cdot \operatorname{rank}\left(T^2\right) .$$

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. Let there be CP decompositions of $T^1[X_{\mathcal{V}^1}]$ and $T^2[X_{\mathcal{V}^2}]$ by

$$T^1[X_{\mathcal{V}^1}] = \left\langle \{\sigma^1[I_1]\} \cup \{V^{1,k}[X_k, I_1] : k \in \mathcal{V}^1\} \right\rangle [X_{\mathcal{V}^1}]$$

and

$$T^2[X_{\mathcal{V}^1}] = \left\langle \{\sigma^2[I_2]\} \cup \{V^{2,l}[X_l, I_2] : l \in \mathcal{V}^2\} \right\rangle [X_{\mathcal{V}^2}] .$$

By linearity of contractions we have

$$\begin{aligned}
\left\langle T^1, T^2 \right\rangle [X_{\tilde{\mathcal{V}}}] &= \sum_{i_1 \in n_1} \sum_{i_2 \in n_2} \sigma^1[I_1 = i_1] \cdot \sigma^2[I_2 = i_2] \\
&\quad \cdot \left\langle \{V^{1,k}[X_k, I_1 = i_1] : k \in \mathcal{V}^1\} \cup \{V^{2,l}[X_l, I_2 = i_2] : l \in \mathcal{V}^2\} \right\rangle [X_{\tilde{\mathcal{V}}}] \\
&= \sum_{i_1 \in n_1} \sum_{i_2 \in n_2} \sigma^1[I_1 = i_1] \cdot \sigma^2[I_2 = i_2] \cdot \\
&\quad \left\langle \{V^{1,k}[X_k, I_1 = i_1] : k \in \mathcal{V}^1 / \tilde{\mathcal{V}}\} \cup \{V^{2,l}[X_l, I_2 = i_2] : l \in \mathcal{V}^2 / \tilde{\mathcal{V}}\} \right\rangle [\varnothing] \cdot \\
&\quad \bigotimes_{k \in \tilde{\mathcal{V}}} V^k[X_k, I_1 = i_1, I_2 = i_2] ,
\end{aligned}$$

where we denote

$$V^k[X_k, I_1 = i_1, I_2 = i_2] = \begin{cases} V^{1,k}[X_k, I_1 = i_1] & \text{if} \quad k \notin \mathcal{V}^2 \\ V^{2,k}[X_k, I_2 = i_2] & \text{if} \quad k \notin \mathcal{V}^1 \\ \left\langle V^{1,k}[X_k, I_1 = i_1], V^{2,k}[X_k, I_2 = i_2] \right\rangle [X_k] & \text{else} \end{cases} .$$

Note, that since $k \in \tilde{\mathcal{V}} \subset \mathcal{V}^1 \cup \mathcal{V}^2$, these slices are well-defined. We build a new decomposition variable I enumerating the summands to indices $[n_1] \times [n_2]$ and have thus found a CP decomposition of $\left\langle T^1, T^2 \right\rangle [X_{\tilde{\mathcal{V}}}]$ of size $n = n_1 \cdot n_2$. This shows the claim in the case of $\operatorname{rank}(\cdot)$.

When the CP decompositions of T^1 and T^2 are boolean, basis or basis+, then the property is preserved in the constructed CP decomposition, since the constructed slices $V^k[X_k, I_1 = i_1, I_2 = i_2]$ are either copies of the leg cores or their contractions and the respective property is preserved in both cases. Thus, the constructive rank bounds hold also for $\operatorname{rank}^{\text{bin}}(\cdot)$, $\operatorname{rank}^{\text{bas}}(\cdot)$ and $\operatorname{rank}^{\text{bas+}}(\cdot)$. $\square$

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. 26 as we show next.

Lemma 27. *For any two tensors $T^1[X_{\mathcal{V}^1}]$, $T^2[X_{\mathcal{V}^2}]$ and any set $\tilde{\mathcal{V}}$ with $\tilde{\mathcal{V}} \cap \mathcal{V}^2 = \varnothing$ we have*

$$\operatorname{rank}\left(\left\langle \{T^1, T^2\} \right\rangle [X_{\tilde{\mathcal{V}}}]\right) \leq \operatorname{rank}\left(T^1\right) .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$*, by* $\operatorname{rank}^{\text{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas+}}(\cdot)$*.*

Proof. As in the proof of Lem. 26 we assume a CP decomposition of $T^1[X_{\mathcal{V}^1}]$ and $T^2[X_{\mathcal{V}^2}]$ and use the linearity of contractions to get

$$\begin{aligned}
\left\langle T^1, T^2 \right\rangle [X_{\tilde{\mathcal{V}}}] &= \sum_{i_1 \in n_1} \sum_{i_2 \in n_2} \sigma^1[I_1 = i_1] \cdot \sigma^2[I_2 = i_2] \cdot \\
&\quad \left\langle \{V^{1,k}[X_k, I_1 = i_1] : k \in \mathcal{V}^1 / \tilde{\mathcal{V}}\} \cup \{V^{2,l}[X_l, I_2 = i_2] : l \in \mathcal{V}^2 / \tilde{\mathcal{V}}\} \right\rangle [\varnothing] \cdot \\
&\quad \bigotimes_{k \in \tilde{\mathcal{V}}} V^{1,k}[X_k, I_1 = i_1] ,
\end{aligned}$$

where we used that $\tilde{\mathcal{V}} \cup \mathcal{V}^2 = \varnothing$. By rearranging the sum of i_2, we have a CP decomposition with decomposition variable I_1 and slices

$$\sigma[I_1 = i_1] = \sum_{i_2 \in n_2} \sigma^1[I_1 = i_1] \cdot \sigma^2[I_2 = i_2] \cdot \\ \left\langle \{V^{1,k}[X_k, I_1 = i_1] \,:\, k \in \mathcal{V}^1/\tilde{\mathcal{V}}\} \cup \{V^{2,l}[X_l, I_2 = i_2] \,:\, l \in \mathcal{V}^2/\tilde{\mathcal{V}}\} \right\rangle [\varnothing] \,.$$

This shows the rank bound for $\mathrm{rank}(\cdot)$. The properties of the CP decomposition are trivially inherited by the constructed decomposition, since the leg cores of the decomposition of $T^1[X_{\mathcal{V}^1}]$ are chosen. Thus, the rank bounds hold also for any other rank in the claim. $\square$

Proof of The. 114. We partition the edges into the set $\mathcal{E}^1 = \{e \in \mathcal{E} \,:\, e \cup \tilde{\mathcal{V}} \neq \varnothing\}$ and $\mathcal{E}^2 = \{e \in \mathcal{E} \,:\, e \cup \tilde{\mathcal{V}} = \varnothing\}$. We then have

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \left\langle \{T^e[X_e] \,:\, e \in \mathcal{E}^1\} \right\rangle \left[X_{\bigcup_{e \in \mathcal{E}^1} e}\right], \left\langle \{T^e[X_e] \,:\, e \in \mathcal{E}^2\} \right\rangle \left[X_{\bigcup_{e \in \mathcal{E}^2} e}\right] \right\rangle [X_{\tilde{\mathcal{V}}}] \tag{47}$$

By an iterative application of Lem. 26 when including the cores to $e \in \mathcal{E}^1$ after each other to the contraction, we get the bound

$$\mathrm{rank}\left(\left\langle \{T^e[X_e] \,:\, e \in \mathcal{E}^1\} \right\rangle \left[X_{\bigcup_{e \in \mathcal{E}^1} e}\right] \right) \leq \prod_{e \in \mathcal{E}^1} \mathrm{rank}(T^e) \,.$$

With the decomposition (47) and Lem. 27 we then arrive at the claim

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] \leq \prod_{e \in \mathcal{E}^1} \mathrm{rank}(T^e) \,.$$

Since the applied lemmata hold also for the restricted CP ranks in the claim, the derived bound is also for those valid. $\square$

Example 25 (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 26 (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.*

15.3 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 115. *For any function*

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

between factored systems we have

$$\mathrm{rank}^{\mathrm{bas}}\left(\rho^f\right) = \prod_{k \in [d]} m_k \,.$$

Proof. The bound follows from The. 108, using that $\ell_0\left(\rho^f\right) = \prod_{k \in [d]} m_k$. $\square$

Let us further provide a construction scheme to find a basis CP decomposition of ρ^f of size $\prod_{k \in [d]} m_k$. We notice that

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

We build for $k \in [d]$ decomposition variables $I_{[d]}$ with $n_k = m_k$ and define leg cores

$$V^k \left[X_k, I_{[d]}\right] = \delta \left[X_k, I_k\right]$$

and for $\tilde{k} \in [r]$ and $i_{[d]}$

$$V^{\tilde{k}} \left[Y_{\tilde{k}}, I_{[d]} = i_{[d]}\right] = e_{f_{\tilde{k}}[X_{[d]}=i_{[d]}]} Y_{\tilde{k}} \,.$$

We then have with a trivial scalar core

$$\rho^f \left[Y_{[p]}, X_{[d]}\right] = \left\langle \{\mathbb{I}\left[I_{[d]}\right]\} \cup \{V^k \left[X_k, I_{[d]}\right] \,:\, k \in [d]\} \cup \{V^{\tilde{k}} \left[Y_{\tilde{k}}, I_{[d]}\right] \,:\, \tilde{k} \in [r]\} \right\rangle \left[Y_{[p]}, X_{[d]}\right] \,.$$

This is a basis CP decomposition of size $\prod_{k\in[d]} m_k$.

In combination with The. 112, The. 115 also provides bounds on all other CP ranks defined in Def. 83. This is obvious, since basis leg slices are the most restrictive properties compared with boolean, directed or basis+.

We restate The. 101 as a basis CP decomposition bound.

Theorem 116. *Let f and be a function between factored systems*

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

and f_k as in The. 101. Then $\rho^f \left[X_{[d]}, X\right]$ has a basis CP *decomposition with decomposition index X, trivial slices $\mathbb{I}\left[X\right]$ leg vectors $\rho^{f_k}\left[X_k, X\right]$, that is*

$$\rho^f \left[X, X_{[d]}\right] = \left\langle \{\mathbb{I}\left[X\right]\} \cup \{\rho^{f_k}\left[X_k, X\right] \,:\, k \in [d]\right\rangle \left[X_{[d]}\right]$$

Proof. The claimed decomposition directly follows from The. 101, since the trivial scalar core $\sigma[X] = \mathbb{I}\left[X\right]$ does not influence the contraction and can be omitted. ☐

Basis CP decompositions can be constructed by understanding the variable O_{in} of the relational encoding of a function $f : \mathcal{U}^{\mathrm{in}} \to \mathcal{U}^{\mathrm{out}}$ as the slice selection variable.

Example 27 (Empirical distributions, see The. 15)**.** *Let there be a data map*

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

We can use The. 116 to find a tensor network representation fo ρ^D as

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

This representation is a basis CP *decomposition, when adding trivial scalar core. This provides also a basis* CP *decomposition for the empirical distribution, since normation can be done by setting a slice core to $\frac{1}{m}\mathbb{I}\left[J\right]$.*

Example 28. *Exponential families The statistic has a* CP *decomposition with rank by the cardinality of states, that is*

$$\rho^\phi \left[Y_{[p]}, X_{[d]}\right] = \left\langle \{\rho^{\phi_l}\left[Y_l, X_{[d]}\right] \,:\, l \in [p]\}\right\rangle \left[Y_{[p]}, X_{[d]}\right] \,.$$

While The. 115 and The. 116 provide CP rank bounds based on the domain factored system, we can also show in the next theorem a bound using the structure of the image.

Theorem 117. *Let $f : \mathcal{U}^{\mathrm{in}} \to \mathcal{U}^{\mathrm{out}}$ be an arbitrary function and let us consider for each $y \in \mathrm{im}\,(f)$ the indicator*

$$\mathbb{I}^{f=y}\left[O_{\mathrm{in}}\right] = \begin{cases} 1 & \textit{if} \quad f(O_{\mathrm{in}} = o_{\mathrm{in}}) = y \\ 0 & \textit{else} \,. \end{cases}$$

The basis+ rank of the relational encoding of f then obeys the bound

$$\mathrm{rank}\left(\rho^f\right) \leq \sum_{y\in\mathrm{im}(f)} \mathrm{rank}\left(\mathbb{I}^{f=y}\right) \,.$$

The bound still holds, when we replace on both sides $\mathrm{rank}\,(\cdot)$ *by* $\mathrm{rank}^{\mathrm{bin}}\,(\cdot)$*, by* $\mathrm{rank}^{\mathrm{bas}}\,(\cdot)$ *or by* $\mathrm{rank}^{\mathrm{bas+}}\,(\cdot)$*.*

Proof. We have

$$\rho^f\left[O_{\text{out}}, O_{\text{in}}\right] = \sum_{y\in\text{im}(f)} e_{I(y)}\left[O_{\text{out}}\right] \otimes \mathbb{I}^{f=y}\left[O_{\text{in}}\right] .$$

For any $y \in \text{im}(f)$ it is obvious that

$$\text{rank}\left(e_{I(y)}\left[O_{\text{out}}\right] \otimes \mathbb{I}^{f=y}\left[O_{\text{in}}\right]\right) = \text{rank}\left(\mathbb{I}^{f=y}\left[O_{\text{in}}\right]\right) ,$$

which also holds true for the other bounds in the claim. We then use the summation bound of The. 113 to get

$$\begin{aligned}\text{rank}\left(\rho^f\left[O_{\text{out}}, O_{\text{in}}\right]\right) &\leq \sum_{y\in\text{im}(f)} \text{rank}\left(e_{I(y)}\left[O_{\text{out}}\right] \otimes \mathbb{I}^{f=y}\left[O_{\text{in}}\right]\right)\\ &\leq \sum_{y\in\text{im}(f)} \text{rank}\left(\mathbb{I}^{f=y}\left[O_{\text{in}}\right]\right) .\end{aligned}$$

Again, the bound still hold for the other ranks in the claim. □

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. 115, since summing the number of non zero coordinators of the indicators is the cardinality of the domain.

Example 29 (Propositional formulas)**.** *Let us now illustrate how the above representation scheme can be leveraged for the sparse representation of propositional formulas. For an arbitrary propositional formula f we have $\text{im}(f) \subset \{0,1\}$ and the indicators*

$$\mathbb{I}^{f=1}\left[X_{[d]}\right] = f\left[X_{[d]}\right] \quad \textit{and} \quad \mathbb{I}^{f=0}\left[X_{[d]}\right] = \neg f\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] - f\left[X_{[d]}\right] .$$

For the conjunction $\wedge[X_0, X_1] = X_0 \wedge X_1$ we have

$$\rho^{\wedge}\left[X_0, X_1\right] = e_1\left[Y_{\wedge}\right] \otimes e_{1,1}\left[X_0, X_1\right] + e_0\left[Y_{\wedge}\right] \otimes \left(\mathbb{I}\left[X_0, X_1\right] - e_{1,1}\left[X_0, X_1\right]\right)$$

and thus

$$\text{rank}^{\text{bas+}}\left(\rho^{\wedge}\right) \leq 3$$

while $\text{rank}^{\text{bas}}\left(\rho^{\wedge}\right) = 4$.

We can even generalize this observation to d-ary conjunctions $\wedge\left[X_{[d]}\right] = X_0 \wedge \ldots \wedge X_{d-1}$ (see Remark 3)

$$\wedge[X_{[d]}] = \bigotimes_{k\in[d]} e_1\left[X_k\right] \quad \textit{and} \quad \neg\wedge[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - \bigotimes_{k\in[d]} e_1\left[X_k\right]$$

and thus

$$\rho^{\wedge}\left[X_{[d]}\right] = e_1\left[Y_{\wedge}\right] \otimes \left(\bigotimes_{k\in[d]} e_1\left[X_k\right]\right) + e_0\left[Y_{\wedge}\right] \otimes \left(\bigotimes_{k\in[d]} e_1\left[X_k\right]\right)$$

Thus, while the basis CP *rank is* $\text{rank}^{\text{bas}}\left(\rho^{\wedge}\right) = 2^d$*, the basis+ rank is bounded by* 3*, independently of* d*.*

15.4 Optimization of sparse tensors

Let us now study the problem of searching for the maximal coordinate in a tensor represented by a monomial decomposition. Given a tensor $T\left[X_{[d]}\right]$ we state this as the problem:

$$\text{argmax}_{x_{[d]}\in\times_{k\in[d]}[m_k]} T\left[X_{[d]} = x_{[d]}\right] \tag{$\mathrm{P}^{\max}_T$}$$

Problem $\mathrm{P}^{\max}_T$ can be reformulated as optimization over the standard simplex

$$\mathcal{M}_{\wedge} = \text{conv}\left(e_{x_{[d]}}\left[X_{[d]}\right] \,:\, x_{[d]} \in \bigtimes_{k\in[d]} [m_k]\right)$$

as

$$\text{argmax}_{\mu\left[L_{[d]}\right]\in\mathcal{M}_{\wedge}} \langle \mu, T\rangle\left[\varnothing\right] .$$

Example 30 (Mode search in exponential families)**.** *Given a statistic ϕ, a canonical parameter θ and a boolean base measure ν, the mode search problem for the member $\mathbb{P}^{\phi,\theta,\nu}$ of the exponential family $\Gamma^{\phi,\nu}$ is*

$$\max_{x_{[d]}\in\times_{k\in[d]}[2]\,:\,\nu\left[X_{[d]}=x_{[d]}\right]=1}\left\langle\gamma^{\phi}\left[X_{[d]}=x_{[d]},L\right],\theta\left[L\right]\right\rangle\left[\varnothing\right]=\max_{\mu\in\mathcal{M}_{\phi,\nu}}\left\langle\mu\left[L\right],\theta\left[L\right]\right\rangle\left[\varnothing\right].$$

Such mode search problems have appeared as generic MAP queries (see Chapter 5). In Chapter 10 we have discussed them for the specific cases of hybrid logic networks and grafting proposal distributions.

15.4.1 Unconstrained Binary Optimization

For leg dimensions $m_k = 2$, Problem $\mathrm{P}_T^{\max}$ is known as the unconstrained binary optimization. Problem $\mathrm{P}_T^{\max}$ is a Higher-Order Unconstrained Binary Optimization (HUBO), when $T\left[X_{[d]}\right]$ has a when T has a monomial decomposition (see Def. 85) with $\left|A^i\right| \leq r$ for all $i\in[n]$, that is when $\mathrm{rank}^r\left(T\right) < \infty$.

Definition 86. *Let $T\left[X_{[d]}\right]$ be a tensor with a monomial decomposition $\{(\lambda^i, A^i, x^i_{A^i}) \,:\, i\in[n]\}$, where $\max_{i\in[n]}\left|A^i\right| = r$. Se then call Problem $\mathrm{P}_T^{\max}$ a r-Order Unconstrained Binary Optimization (HUBO), which we denote as*

$$\mathrm{argmax}_{x_{[d]}\in\times_{k\in[d]}[2]}\sum_{i\in[n]}\lambda^i\left\langle e_{x^i_{A^i}}\left[X_{A^i}\right]\right\rangle\left[X_{[d]}=x_{[d]}\right]. \qquad (\mathrm{P}_T^{\mathrm{HUBO}})$$

Remark 28 (Leg dimensions larger than 2)**.** *We demanded leg dimensions $m_k = 2$ to have boolean 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. 7.3.2).*

The sparsity $\mathrm{rank}^r\left(T\right)$ is the minimal number of monomials, for which a weighted sum is equal to T. Thus we interpret Problem $\mathrm{P}_T^{\mathrm{HUBO}}$ as searching for the maximum in a polynomial consistent of $\mathrm{rank}^r\left(T\right)$ monomial terms.

Problem $\mathrm{P}_T^{\mathrm{HUBO}}$ is called Quadratic Unconstrained Binary Optimization problems, if $r = 2$. We can transform certain Higher-Order Unconstrained Binary Optimization (HUBO) problems into Quadratic Unconstrained Binary Optimization (QUBO) problems by introducing auxiliary variables. An example of such an transform is provided by the next lemma.

Lemma 28. *For any $x_0,\ldots,x_{d-1}\in[2]$ 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. □

15.4.2 Integer Linear Programming

Let us now show how optimization problems can be represented as linear programming problems. To this end, we understand each index tuple $x_{[d]}\in\bigtimes_{k\in[d]}[m_k]$ as a vector $v_{x_{[d]}}\left[L\right]\in\mathbb{R}^d$ with coordinates

$$v_{x_{[d]}}\left[L=k\right]=x_k\,.$$

Definition 87. *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*

$$\mathrm{argmax}_{x_{[d]}\in \times_{k\in[d]}[m_k]} \left\langle c[L], v_{x_{[d]}}[L]\right\rangle[\varnothing] \quad \textit{subject to} \quad \left\langle M[J,L], v_{x_{[d]}}[L]\right\rangle[\varnothing] \prec b[J]\,, \tag{$\mathrm{P}^{\mathrm{ILP}}_{c,M,b}$}$$

where by $\prec$ we denote partial ordering of tensors (see Def. 76).

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 $c^{\mathcal{M}}[L]$, where L takes values in $[d+n]$, as

$$c^{\mathcal{M}}[L=l] = \begin{cases} \lambda^{l-d} & \text{if} \quad l > d \\ 0 & \text{else} \end{cases}. \tag{48}$$

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

$$m_i = \sum_{\tilde{i}=0}^{i} \left(\left|A^{\tilde{i}}\right| + 1\right)$$

and further enumerate the set A^i by a function $I : [\left|A^i\right|] \to A^i$.

We then construct a matrix $M^{\mathcal{M}}[J,L]$, were for $l \in [d+n]$, $i \in [n]$ and $j \in [\left|A^i\right|]$ we have

$$M^{\mathcal{M}}[J = m_i + j, L = l] = \begin{cases} 1 - 2\cdot x^i_{I(j)} & \text{if} \quad j < \left|A^i\right|,\ j = l \text{ and } l = I(j) \\ 1 & \text{if} \quad j < \left|A^i\right| \text{ and } l = d+i \\ -x^i_{I(j)} & \text{if } j = \left|A^i\right| \text{ and } l = I(j) \\ -1 & \text{if } j = \left|A^i\right| \text{ and } l = d+i \\ 0 & \text{else} \end{cases}. \tag{49}$$

Similarly, we define $b^{\mathcal{M}}[J]$ as the vector which nonvanishing coordinates are for $i \in [n]$ at

$$b^{\mathcal{M}}[J = m_i + j] = \begin{cases} 1 - x^i_{I(j)} & \text{if } j < \left|A^i\right| \\ -1 + \left|\{k \in A^i \,:\, x^i_k = 1\}\right| & \text{if } j = \left|A^i\right| \end{cases}. \tag{50}$$

Informally, we pose for each tuple (λ, A, x_A) $|A|+1$ 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 118. *Given a monomial decomposition $\mathcal{M} = \{(\lambda^i, A^i, x^i_{A^i}) \,:\, i \in [n]\}$ of a tensor T, let $y^{ILP,\mathcal{M}}$ be a solution of the integer linear program defined by the matrix and vectors in equations (48), (49) and (50). Then we have*

$$y^{ILP,\mathcal{M}}|_{[d]} \in \mathrm{argmax}_{x_{[d]}\in\times_{k\in[d]}[2]} \quad T\left[X_{[d]} = x_{[d]}\right].$$

where by $y^{ILP,\mathcal{M}}|_{[d]}$ we denote the restriction of the index tuple $y^{ILP,\mathcal{M}}$ to the first d.

Proof. We show that the linear constraints by

$$\left\langle M^{\mathcal{M}}[J,L], v_{y_{[d+n]}}[L]\right\rangle[\varnothing] \prec b^{\mathcal{M}}[J]$$

are satisfied for a vector $y_{[d+n]} = (x_{[d]}, z_{[n]})$, if and only if for all $i \in [n]$ the product constraints

$$z_i = \left(\prod_{k \in A^i \,,\, x^i_k = 0} (1 - x_k \right) \cdot \left(\prod_{k \in A^i \,,\, x^i_k = 1} x_k \right) \tag{51}$$

hold. We will see, that the linear constraints where J takes indices in $m_i + [|A^i|]$ are equivalent to the upper bound on z_i and the constraint to $J = m_i + |A^i|$ is equivalent to an lower bound on z_i. To show the upper bound, we notice that for any $j \in [|A^i|]$ the constraint $J = m_i + j$ is

$$z_i \leq \begin{cases} x_{I(j)} & \text{if} \quad x^i_{I(j)} = 1 \\ (1 - x_{I(j)}) & \text{if} \quad x^i_{I(j)} = 0 \end{cases} .$$

Thus, whenever a factor on the right side of (51) is 0, we have $z_i = 0$ if the respective constraint is satisfied. We conclude, that

$$z_i \leq \left(\prod_{k \in A^i \,,\, x^i_k = 0} (1 - x_k) \right) \cdot \left(\prod_{k \in A^i \,,\, x^i_k = 1} x_k \right) .$$

To show the lower bound, we have the constraint to $J = m_i + |A^i|$ by

$$z_i \geq 1 - \left(\sum_{k \in A^i \,,\, x^i_k = 0} x_k \right) + \left(\sum_{k \in A^i \,,\, x^i_k = 1} (x_k - 1) \right) .$$

The right side of this inequality is 1, if and only if all factors on the right side of (51) are 1, and less or equal to 0 else. Thus, whenever this constraint is satisfied, we have

$$z_i \geq \left(\prod_{k \in A^i \,,\, x^i_k = 0} (1 - x_k) \right) \cdot \left(\prod_{k \in A^i \,,\, x^i_k = 1} x_k \right) .$$

In summary, the equation (51) holds, if and only if the constraints where J takes indices in $m_i + [|A^i| + 1]$ are satisfied.

This characterization of the constraints implies, that for any $x_{[d]} \in \bigtimes_{k \in [d]} [m_k]$ there is exactly one feasible index $y_{[d+n]}$ with $(y_{[d+n]})|_{[d]} = x_{[d]}$, and the objective takes for this index the value

$$\begin{aligned}
\left\langle c[L], v_{y_{[d+n]}}[L] \right\rangle [\varnothing] &= \sum_{i \in [n]} \lambda^i \cdot z_i \\
&= \sum_{i \in [n]} \lambda^i \cdot \left(\prod_{k \in A^i \,,\, x^i_k = 0} (1 - x^i_k) \right) \cdot \left(\prod_{k \in A^i \,,\, x^i_k = 1} x^i_k \right) \\
&= T\left[X_{[d]} = x_{[d]}\right] .
\end{aligned}$$

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. $\square$

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

$$\ell_0\left(M^{\mathcal{M}}[J, L]\right) \leq \sum_{i \in [n]} 3 \cdot |A^i| + 1 .$$

15.5 Discussion and Outlook

The slice selection network described here have been taylored to the CP format.

We can extend the slice selection network to parametrize more general tensor network formats than the CP format. Here the graph structure of the format itself can be optimized, by the usage of variable selectors. The activation selectors, generalizing the connective selection, are then choices of specific cores in the format. Thus, each neuron represents an hypercore $T^e[X_e]$, where $e \subset \mathcal{V}$ is selected by variable selectors and the values of the tensor T^e by activation selectors.

16 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.

16.1 Selection tensor networks for CP decompositions

In this section, we show that the set of tensors respresentable in specific CP formats coincides with the expressivity of tailored selection architectures (see Chapter 8). We first define a basis+ CP selecting tensor and then show its decomposition into a formula selecting neural network.

Definition 88. *Given a set of categorical variables $X_{[d]}$ with $m = \max_{k\in[d]} m_k$, a* CP *selecting tensor of maximal cardinality r is the tensor*

$$\mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{0,0}, \dots, L_{r-1,0}, L_{0,1}, \dots, L_{r-1,1}\right]$$

with dimensions

$$p_{s,0} = m+1, p_{s,1} = d \quad \textit{for} \quad s \in [r]$$

and coordinates

$$\begin{aligned}&\mathcal{H}_{\wedge,d,r}\left[X_{[d]} = x_{[d]}, L_{0,0} = l_{0,0}, \dots, L_{r-1,0} = l_{r-1,0}, L_{0,1} = l_{0,1}, \dots, L_{r-1,1} = l_{r-1,1}\right] \\ &= \begin{cases} 1 & \textit{if} \quad \forall k \in [d], s \in [r] : \big(l_{s,1} = k \wedge l_{s,0} \neq m\big) \Rightarrow (l_{s,0} = x_k) \\ 0 & \textit{else} \end{cases} .\end{aligned}$$

Intuitively, the selection variables $L_{s,1}$ of $\mathcal{H}_{\wedge,d,r}$ select a variable out of $[d]$ to be included in A and the selection variables $L_{s,0}$ select a corresponding state to that variable. As in Chapter **??** we refer to variables $L_{s,1}$ as variable selectors and $L_{s,0}$ as state selectors. When $L_{s,0} = m$, the slice is left trivial, that is the selected variable is effectively not included in A. This then allows to also represent slices where $|A| < r$. We in the following prove this more formally.

Theorem 119. *Let the non-vanishing indices of $\theta\left[L_{[r]\times[m+1]}\right]$ denote by $\{l^i_{[r]\times[m+1]} : i \in [n]\}$. Let further $\mathcal{M} \subset [n]$ be the set of agreeing selection indices, that is*

$$\mathcal{M} = \{i : i \in [n], \forall s, \tilde{s} \in [r] : \big(l^i_{s,1} = l^i_{\tilde{s},1} \wedge l^i_{s,1} \neq m \wedge l^i_{\tilde{s},1} \neq m\big) \Rightarrow \big(l^i_{s,1} = l^i_{\tilde{s},1}\big)\}$$

Then

$$\begin{aligned}&\left\langle \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle \left[X_{[d]}\right] \\ &= \sum_{i\in\mathcal{M}} \theta\left[L_{[r]\times[m+1]} = l^i_{[r]\times[m+1]}\right] \left\langle \{e_{l_{s,0}}\left[X_{l_{s,1}}\right] : s \in [r], l_{s,0} \neq m\}\right\rangle \left[X_{[d]}\right]\end{aligned}$$

Further we have

$$\mathrm{rank}^{\mathrm{bas+}}\left(\left\langle \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle \left[X_{[d]}\right]\right) \leq \ell_0\left(\theta\right) .$$

Proof. Let us notice, that whenever $i \notin \mathcal{M}$ then

$$\mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{[r]\times[m+1]} = l_{[r]\times[m+1]}\right] = 0\left[X_{[d]}\right] ,$$

since the condition for non-vanishing coordinates

$$\forall k \in [d], s \in [r] : \big(l_{s,1} = k \wedge l_{s,0} \neq m\big) \Rightarrow (l_{s,0} = x_k)$$

in Def. 88 cannot be satisfied for any $x_{[d]}$. For $i \notin \mathcal{M}$ we have

$$\mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{[r]\times[m+1]} = l_{[r]\times[m+1]}\right] = \left\langle \{e_{l_{s,0}}\left[X_{l_{s,1}}\right] : s \in [r], l_{s,0} \neq m\}\right\rangle \left[X_{[d]}\right] .$$

We now use these insights and linearity of contraction to get

$$\begin{aligned}&\left\langle \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle \left[X_{[d]}\right] \\ &= \sum_{i\in[n]} \theta\left[L_{[r]\times[m+1]} = l^i_{[r]\times[m+1]}\right] \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{[r]\times[m+1]} = l_{[r]\times[m+1]}\right] \\ &= \sum_{i\in\mathcal{M}} \theta\left[L_{[r]\times[m+1]} = l^i_{[r]\times[m+1]}\right] \left\langle \{e_{l_{s,0}}\left[X_{l_{s,1}}\right] : s \in [r], l_{s,0} \neq m\}\right\rangle \left[X_{[d]}\right]\end{aligned}$$

This shows the decomposition claim. We notice, that this is a basis+ CP decomposition of size $\mathcal{M}$ and thus

$$\mathrm{rank}^{\mathrm{bas+}}\left(\left\langle \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle\left[X_{[d]}\right]\right) \leq |\mathcal{M}| \leq \ell_0\left(\theta\right) .$$

□

Towards a practicle usage of this representation scheme for basis+ CP decompositions, let us show that the CP selecting tensor coincides with a formula selecting network.

Lemma 29. *The* CP *selection tensor coincides with a formula selecting neural network with neurons (see Figure 16.1):*

- *unary state selecting neurons enumerated by s, selecting one of the $X_{[d]}$ with the variable $L_{s,1}$ and selecting a state, extended by a possible choice of trivial legs $\mathbb{I}$*
- *r-ary output neuron fixed to the $\wedge$ connective.*

Proof. This can be easily checked on each coordinate. □

If $m_k = 2$ for $k \in [d]$, the state selector chooses between the connectives $\{\neg, \mathrm{Id}, \mathrm{True}\}$. We can in that case understand each slice by a logical term.

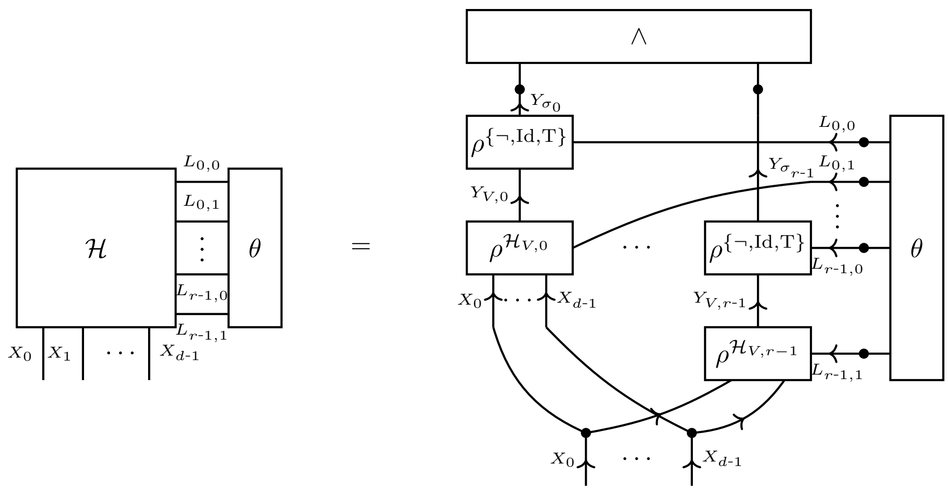


Figure 37: Representation of a basis+ Tensor by the contraction of a parameter tensor θ with a CP selecting architecture $\mathcal{H}$, which has a decomposition as a formula selecting neural network (see Lem. 29). The nonzero coordinates of θ represent slices of the CP decomposition.

16.1.1 Applications

One application is as a parametrization scheme in the approximation of a tensor by a slice-sparse tensor, see Chapter 16. 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 29 (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 $e_0, e_1, \mathbb{I}[1]$. 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.*

16.2 Approximation of Energy tensors

The Hilbert-Schmidt norm of a tensor is the contraction of the coordinatewise transform with the square function

$$\|T\left[X_{[d]}\right]\|_2 = \sqrt{\left\langle\left(T\left[X_{[d]}\right]\right)2\right\rangle\left[\varnothing\right]}\,.$$

Approximation involving a selection architecture $\mathcal{H}$ is the problem

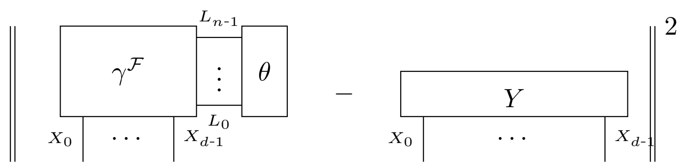
$$\mathrm{argmin}_{\theta\in\Gamma^{\mathcal{G}}} \|E - \left\langle \gamma^{\mathcal{H}}, \theta \right\rangle \left[X_{[d]}\right] \|^2 .$$

Direct approximation is then the choice of the minterm statistic

$$\mathrm{argmin}_{\theta\in\Gamma^{\mathcal{G}}} \|E\left[X_{[d]}\right] - \theta \left[X_{[d]}\right] \|^2 .$$

In a tensor network diagram we depict this as

$\mathrm{argmin}_{\theta\in\Gamma^{\mathcal{G}}}$



Example 31 (Approximate based on a slice sparsity selecting architecture)**.** *Use a term selecting neural network (conjunction neuron on d unary neurons selecting a variable and* $\mathrm{Id}, \neg, \mathrm{True}$ *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.*

16.3 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.

16.3.1 Weighted Squares Loss Trick

Lemma 30. *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 120 (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*

$$\mathrm{argmax}_{T\in\Gamma} \left\langle I, T\right\rangle [\varnothing] = \mathrm{argmin}_{T\in\Gamma} \left\langle I, (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2\right\rangle [\varnothing] \tag{52}$$

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

$$\left\langle I, T\right\rangle [\varnothing] = \left\langle I, \mathbb{I}\left[X_{[d]}\right]\right\rangle [\varnothing] - \left\langle I, (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2\right\rangle [\varnothing]$$

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. 120 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. 120 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 32 (Proposal distribution maxima)**.** *The Problem* **??** *of finding the maximal coordinate can thus be turned into*

$$\mathrm{argmax}_{l_{[n]}} \left\langle (\mathbb{P}^D - \tilde{\mathbb{P}}), \mathcal{H} \right\rangle \left[L_{[n]} = l_{[n]}\right] = \mathrm{argmin}_{l_{[n]}} \left\langle (\mathbb{P}^D - \tilde{\mathbb{P}}), \left(\left\langle \mathcal{H}, e_{l_{[n]}}\left[L_{[n]}\right]\right\rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2 \right\rangle [\varnothing] \,.$$

16.3.2 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 33 (Fitting a boolean tensor by a formula tensor)**.** *Given a tensor T, we want to find a formula $f \in \mathcal{F}$ such that it approximates 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

$$\mathrm{argmax}_{f\in\mathcal{F}} \left\langle f \Leftrightarrow T \right\rangle [\varnothing] \,.$$

We can use the squares risk trick and get an equivalent problem

$$\mathrm{argmin}_{f\in\mathcal{F}} \| \left\langle f \Leftrightarrow T \right\rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right] \|^2 \,.$$

We have since T and f are boolean

$$\| \left\langle f \Leftrightarrow T \right\rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right] \|^2 = \| \left\langle f \right\rangle \left[X_{[d]}\right] - T\left[X_{[d]}\right] \|^2$$

Now, when representing $\mathcal{F}$ in a formula selecting architecture we have

$$\mathrm{argmin}_{\theta\in\Gamma_1} \| \left\langle \mathcal{H}, \theta \right\rangle \left[X_{[d]}\right] - T\left[X_{[d]}\right] \|^2 \,.$$

where Γ_1 is the set of basis tensors.

When we extend Γ_1 to a set including the trivial tensor $\mathbb{I}[L]$, when the formulas f are pairwise disjoint and $T\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$, then the solution would be $\mathbb{I}[L]$.

16.4 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).

16.4.1 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.

16.5 Regularization and Compressed Sensing

When regularizing the least squares problem by enforcing the sparsity of θ, we arrive at the compressed sensing problem

$$\mathrm{argmin}_{\theta[L]} \ell_0(\theta) \quad \text{subject to} \quad \left\| \left\langle \gamma^\phi, \theta \right\rangle \left[X_{[d]}\right] - E\left[X_{[d]}\right] \right\|_2 \leq \eta \tag{53}$$

Here, the sensing matrix is the selection tensor.

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

$$T^e[X_e] \leftarrow \mathrm{argmin}_{T^e[X_e]} \left\langle I, (\langle \mathcal{H}, \theta\rangle \left[X_{[d]}\right] - Y[X_{[d]}])^2 \right\rangle [\varnothing]$$

  end for
end while

Example 34 (Formula fitting to an example)**.** *Choosing the best formula fitting data (see Example 33) is the problem*

$$\mathrm{argmin}_{\theta[L]\,:\,\ell_0(\theta)=1} \left\| \left\langle I, \gamma^{\phi}, \theta \right\rangle \left[X_{[d]}\right] - Y \right\|_2 \tag{54}$$

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 Foucart and Rauhut (2013).

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 Goeßmann (2021)) are not applicable.

Example 35 (Sensing matrix for propositional Formulas)**.** *Let there be a set $\mathcal{F}$ of formulas, then we have*

$$\left\langle \gamma^{\mathcal{F}}\left[X_{[d]}, L_{\mathrm{in}}\right], \gamma^{\mathcal{F}}\left[X_{[d]}, L_{\mathrm{out}}\right] \right\rangle [L_{\mathrm{in}} = l_{\mathrm{in}}, L_{\mathrm{out}} = l_{\mathrm{out}}] = \langle f_{l_{\mathrm{in}}}, f_{l_{\mathrm{out}}} \rangle [\varnothing] \,.$$

If the formulas have disjoint model sets then

$$\left\langle \gamma^{\mathcal{F}}\left[X_{[d]}, L_{\mathrm{in}}\right], \gamma^{\mathcal{F}}\left[X_{[d]}, L_{\mathrm{in}}\right] \right\rangle [L_{\mathrm{in}} = l_{\mathrm{in}}, L_{\mathrm{out}} = l_{\mathrm{out}}] = \begin{cases} \langle f_{\mathrm{in}} \rangle [\varnothing] & \textit{if } l_{\mathrm{in}} = l_{\mathrm{out}} \\ 0 & \textit{else} \end{cases} \,.$$

In that case, the sensing matrix is a restricted isometry, in the sense that the norm of any mapped vector is its norm mutliplied by a factor between the smallest and the largest $\langle f_{\mathrm{in}} \rangle [\varnothing]$.

Example 36 (Sensing matrix for slice selection networks)**.** *For the slice selection network*

$$\begin{aligned} &\left\langle \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{\mathrm{in}}\right], \mathcal{H}_{\wedge,d,r}\left[X_{[d]}, L_{\mathrm{out}}\right] \right\rangle [L_{\mathrm{in}} = l_{\mathrm{in}}, L_{\mathrm{out}} = l_{\mathrm{out}}] \\ &= \begin{cases} 0 & \textit{if for a } \tilde{k} \in A^{l_{\mathrm{in}}} \cap A^{l_{\mathrm{out}}} \textit{ we have } x_{\tilde{k}}^{l_{\mathrm{in}}} \neq x_{\tilde{k}}^{l_{\mathrm{out}}} \\ \prod_{\tilde{k} \notin A^{l_{\mathrm{in}}} \cup A^{l_{\mathrm{out}}}} m_{\tilde{k}} & \textit{else} \end{cases} \,. \end{aligned}$$

Given a fixed l_{in}, the maximum value in the respective slice is thus taken at $l_{\mathrm{in}} = l_{\mathrm{out}}$.

17 Message Passing

In this chapter we introduce local contraction passed along tensor clusters to calculate global contractions exactly or approximatively. 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 to calculate contractions are motivated by commutations of contractions. We first show this property and then provide message passing schemata.

17.1 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 121 (Commutativity of Contractions)**.** *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 for any $\tilde{\mathcal{V}} \subset \mathcal{V}$*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \mathcal{T}^{\mathcal{G}_1} [X_{\mathcal{V}^1}] \cup \{\left\langle \mathcal{T}^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2 \cap (\mathcal{V}^1 \cup \tilde{\mathcal{V}})}\right]\} \right\rangle [X_{\tilde{\mathcal{V}}}] \, .$$

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 $x_{\mathcal{V}^2 \cap (\tilde{\mathcal{V}} \cup \mathcal{V}^1)}$

$$\left\langle \mathcal{T}^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2 \cap (\tilde{\mathcal{V}} \cup \mathcal{V}^1)} = x_{\mathcal{V}^2 \cap (\tilde{\mathcal{V}} \cup \mathcal{V}^1)}\right] = \left(\sum_{x_{\mathcal{V}^2/(\tilde{\mathcal{V}} \cup \mathcal{V}^1)}} \prod_{e \in \mathcal{E}_2} T^e [X_e = x_e] \right) .$$

It follows with the above, that

$$\begin{aligned}
\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] &= \sum_{x_{\mathcal{V}^1/\tilde{\mathcal{V}}}} \left(\prod_{e \in \mathcal{E}_1} T^e [X_e = x_e] \right) \cdot \left\langle \mathcal{T}^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2 \cap (\tilde{\mathcal{V}} \cup \mathcal{V}^1)} = x_{\mathcal{V}^2 \cap (\tilde{\mathcal{V}} \cup \mathcal{V}^1)}\right] \\
&= \left\langle \mathcal{T}^{\mathcal{G}_1} \cup \{\left\langle \mathcal{T}^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2 \cap (\mathcal{V}^1 \cup \tilde{\mathcal{V}})}\right]\} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \, .
\end{aligned}$$

□

We can interpret the inner contraction $\left\langle \mathcal{T}^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2 \cap (\mathcal{V}^1 \cup \tilde{\mathcal{V}})}\right]$ as a message, sent from $\mathcal{G}_2$ to $\mathcal{G}_1$. Based on this intuition, we will define message passing schemes in the next section.

17.2 Exact Contractions

We apply Theorem 121 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.

17.2.1 Construction of Cluster Graphs

Let us first introduce with the cluster graph a mechanism to coarse grain the hypergraph capturing a tensor network.

Definition 89 (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 $\mathcal{V}_i = \bigcup_{e \in C_i} e$.

We say, that the cluster graph satisfies the running intersection property, when for any clusters C_i and C_j and any $v \in \mathcal{V}_i \cup \mathcal{V}_j$ there is a path between C_i and C_j with $v \in \mathcal{V}_k$ for any cluster C_k along the path.

Given a cluster graph to a tensor network, we can execute any global contraction by a contraction of local contraction to each cluster.

Theorem 122. *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

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \{ \left\langle \mathcal{T}^{C_i} \right\rangle \left[X_{\mathcal{V}_i \cap (\tilde{\mathcal{V}}_i \cup \tilde{\mathcal{V}})} \right] : i \in [n] \} \right\rangle [X_{\tilde{\mathcal{V}}}] \, .$$

Proof. By Theorem 121 applied for each cluster seen as a subgraph. □

17.2.2 Message Passing to calculate contractions

Having a hypergraph $\mathcal{G}$, we iteratively apply Theorem 121 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

$$\delta_{j \to i} [X_{\mathcal{V}^i \cap \mathcal{V}^j}] = \left\langle \{ \delta_{\tilde{j} \to j} \left[X_{\mathcal{V}^{\tilde{j}} \cap \mathcal{V}^j} \right] \,:\, C_{\tilde{j}} \prec C_j \} \cup \mathcal{T}^{C_j} \right\rangle [X_{\mathcal{V}^i \cap \mathcal{V}^j}] \, .$$

We note, that the messages are well defined by these recursive equations, exactly when the cluster graph is a polytree.

When the cluster graph is a tree, we can choose a root cluster and order the clusters by the topological order $\prec$.

Lemma 31. *When the cluster graph is a tree satisfying the running intersection property, we have for neighbored clusters C_i and C_j with $C_j \prec C_i$*

$$\delta_{i \to \tilde{i}} \left[X_{\mathcal{V}^i \cap \mathcal{V}^{\tilde{i}}} \right] = \left\langle \{ \mathcal{T}^{C_j} \,:\, C_j \prec C_i \} \right\rangle \left[X_{\mathcal{V}^i \cap \mathcal{V}^{\tilde{i}}} \right] \, .$$

Proof. By induction over the cardinality n of the preceding clusters.

$n = 1$: For a single preceding cluster the statement holds trivial, since the preceding cluster is the cluster itself.

$n + 1 \to n$: Let us now assume, that the statement holds for up to n preceding clusters, and let there be $n + 1$ preceding clusters. We build another cluster graph for the cores different from C_i, by assigning each cluster $C_{\tilde{j}}$ to the neighbor C_j where $j \in \mathrm{N}(i)$, for which

$$C_{\tilde{j}} \prec C_j \, .$$

We use The. 122 on this constructed cluster graph and get

$$\left\langle \{ \mathcal{T}^{C_{\tilde{j}}} \left[X_{\mathcal{V}^{\tilde{j}}} \right] \,:\, \tilde{j} \neq i \} \right\rangle [X_{\mathcal{V}^i}] = \left\langle \left\{ \left\langle \{ \mathcal{T}^{C_{\tilde{j}}} \left[X_{\mathcal{V}^{\tilde{j}}} \right] \,:\, \tilde{j} \prec j \} \right\rangle [X_{\tilde{\mathcal{V}}^j}] \,:\, j \in \mathrm{N}(i) \right\} \right\rangle [X_{\mathcal{V}^i}]$$

Here by $\tilde{\mathcal{V}}^j$ we denote the intersection of

$$\tilde{\mathcal{V}}^j = \left(\bigcup_{\tilde{j} \prec j} \mathcal{V}^{\tilde{j}} \right) \cap \left(\bigcup_{\tilde{j} \not\prec j} \mathcal{V}^{\tilde{j}} \right)$$

By the running intersection property, we have $\mathcal{V}^j \cap \mathcal{V}^i = \tilde{\mathcal{V}}^j$.

We further have for any $j \in \mathrm{N}(i)$ that

$$\left|\{\tilde{j} \prec j\}\right| \leq n\,.$$

We can therefore apply the assumption of the induction and get

$$\left\langle \{\mathcal{T}^{C_{\tilde{j}}}\left[X_{\mathcal{V}^{\tilde{j}}}\right] : \tilde{j} \prec j\}\right\rangle \left[X_{\tilde{\mathcal{V}}^{j}}\right] = \delta_{j\to i}\left[X_{\mathcal{V}^{j}\cap\mathcal{V}^{i}}\right]$$

With the above, we arrive at

$$\delta_{i\to\tilde{i}}\left[X_{\mathcal{V}^{i}\cap\mathcal{V}^{\tilde{i}}}\right] = \left\langle \{\mathcal{T}^{C_j} : C_j \prec C_i\}\right\rangle \left[X_{\mathcal{V}^{i}\cap\mathcal{V}^{\tilde{i}}}\right]\,.$$

□

Theorem 123. *When the cluster graph is a tree satisfying the running intersection property, then we have for each cluster C_i with neighbors $\mathrm{N}(i)$*

$$\left\langle \mathcal{T}^{\mathcal{G}}\right\rangle \left[X_{\mathcal{V}_i}\right] = \left\langle \{\delta_{j\to i}\left[X_{\mathcal{V}^{i}\cap\mathcal{V}^{j}}\right] : j \in N(i)\} \cup \{\mathcal{T}^{C_i}\}\right\rangle \left[X_{\mathcal{V}_i}\right]\,. \tag{55}$$

Proof. We use the topological order $\prec$ of the clusters by the tree, when choosing a root by cluster C_i.

The claim then follows from The. 122 and Lem. 31.

□

While we have defined message passing along the topological order of a graph, we can also define messages against the topological order, that is

$$\delta_{j\leftarrow i} = \left\langle \{\delta_{i\leftarrow\tilde{j}} : C_i \prec C_{\tilde{j}}\} \cup \mathcal{T}^{C_i}\right\rangle \left[X_{\mathcal{V}_i\cap\mathcal{V}_j}\right]$$

To this end, we can get a similar statement for nodes, which are not the rotes of the cluster tree. The constractions at each cluster can then be computed batchwise, based on message passed along a topological order and against.

These message passing schemes can be derived from Lagrangian parameters given a local consistency polytope Wainwright and Jordan (2008).

17.2.3 Variable Elimination Cluster Graphs

Remark 30 (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,*

17.2.4 Bethe Cluster Graphs

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 38).

Definition 90. *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 $\Delta^{\mathcal{G}} = \left\{\{v^e : e \ni v\} : v \in \mathcal{V}\right\}$, each of which decorated by a delta tensor $\delta^{\{v^e : e\ni v\}}$*

By Lem. 20 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).

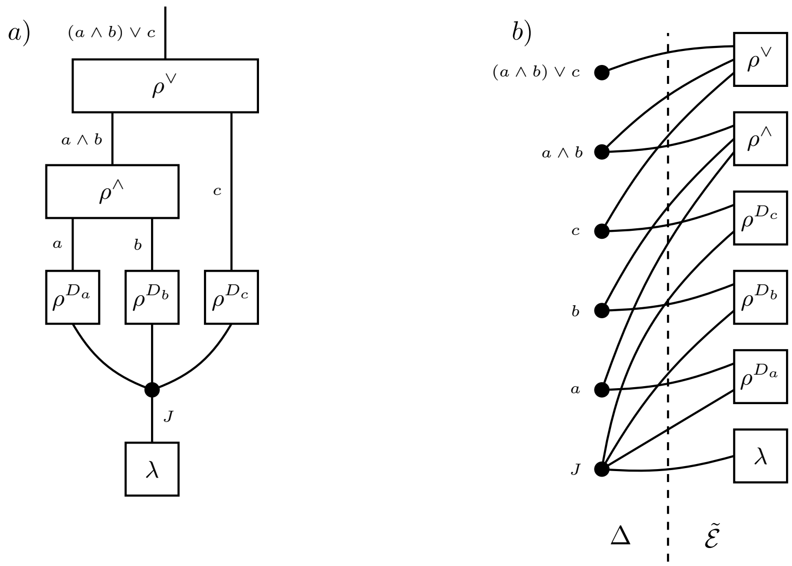


Figure 38: Example of a Bethe Cluster Graph. a) Example of a Tensor Network $\mathcal{T}^{\mathcal{G}}$, which represents the by λ averaged evaluation of the formula $(a \wedge b) \vee c$ on data D. b) Corresponding Bethe Cluster Hypergraph, which dual is bipartite by the sets Δ and $\tilde{\mathcal{E}}$.

17.2.5 Computational Complexity

Tree-width here: By building a cluster tree to any partition, simply by including variables for the running intersection property. The maximum number of variables at a cluster is the tree-width and provides a complexity bound for the local contractions.

Naive execution of $\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\tilde{\mathcal{V}}\right]$: $\prod_{v\in\mathcal{V}} m_v$ 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.

17.3 Boolean Message Passing

Instead of the exact calculation of a contraction, let us now investigate schemes to sparsify the tensors before a contraction. To this end, we first show underlying properties of contractions enabling these schemes.

17.3.1 Monotonocity of tensor contraction

To state the next theorem we use the nonzero function $\mathbb{I}_{\neq 0} : \mathbb{R} \rightarrow [2]$ by $\mathbb{I}_{\neq 0}(x) = 1$ if $x \neq 0$ and $\mathbb{I}_{\neq 0}(x) = 0$ else. Applied coordinatewise on tensors it marks the nonzero coordinates by 1.

We show that adding boolean tensor cores to an contraction orders the results by the partial ordering introduced in Def. 76.

Theorem 124 (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

$$\mathbb{I}_{\neq 0} \left(\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \right) = 1 \,.$$

□

17.3.2 Invariance of adding subcontractions

Let us now state an equivalence of the contraction, when we add the result of the same contraction. This property was used in the proof of The. 48.

Theorem 125 (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 31. *Similar statements hold, when dropping the non-negativity assumption on the, but demanding that all variables are left open.*

17.3.3 Basis Calculus as message passing scheme

Message Passing of directed and boolean message by relational encoding of functions can be interpreted as function evaluation. Each subfunction evaluation is passed in its one-hot encoding.

This is because any relational encoding of a function, the decomposition

$$\rho^f = \sum_{y \in \mathrm{im}(f)} \Big(\sum_{i : f(i) = y} e_i \Big) \otimes e_y$$

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 $e_{f(i)}$.

Note, that this is exact, whenever the graph is directed and acyclic. We do not need acyclicity of the underlying undirected graph.

Remark 32 (Basis Calculus as Message Passing)**.** *Given a tensor network of directed and binary tensor cores, each representing a function f_e depending on variables e^{in}. 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*

$$\left\langle \{ \rho^{f_e} [X_{e^{\text{out}}}, X_{e^{\text{in}}}] \, : \, e = (e^{\text{out}}, e^{\text{in}}) \in \mathcal{E} \} \right\rangle [\mathcal{V}^2] = e_{f(x_v \, : \, v \in \mathcal{V}^1)} \,.$$

17.3.4 Application

This properties can be applied as a sparsification of tensors before the execution of a contraction. The Knowledge Propagation Algorithm 7 produces in the knowledge cores conditions on non-vanishing coordinates. Thus, the knowledge cores can be locally contracted with the tensors, as a sparsification before performing a global contraction.

17.4 Discussion

Computing contractions by message passing is known to the graphical model community as belief propagation. There, the objective is the calculation of marginal probabilities of Markov Networks, which involve contractions of the corresponding factor tensors.

Remark 33 (Approximate Message Passing Schemes)**.** *When the cluster graphs are not trees, we cannot find a topological order of the clusters any more. Messages can still be defined implicitly by received neighbored messages, but the equivalence with global contractions cannot be established in general.*

Such algorithms are known in the graphical model community as loopy belief propagation.

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.

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

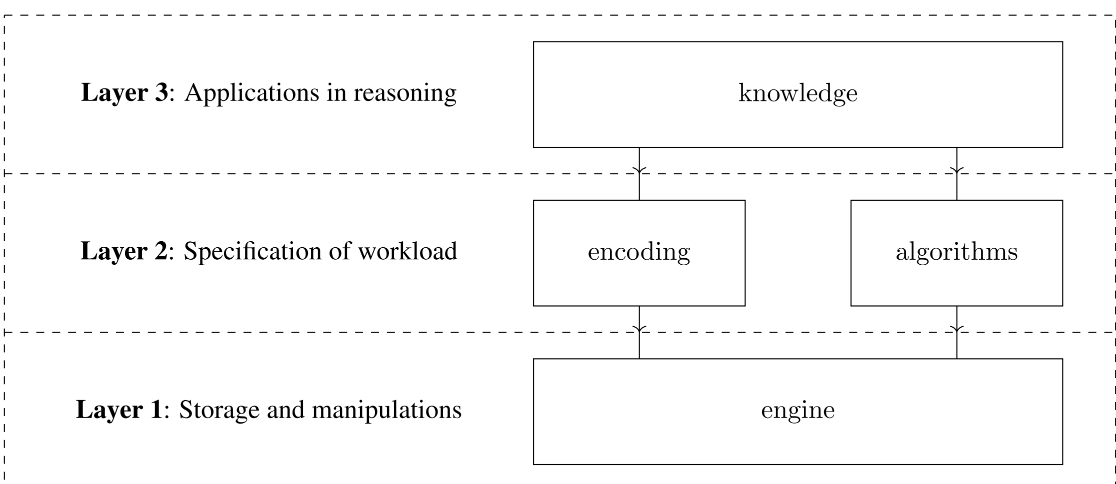
```
!pip install tnreason
```

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. 85). 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 spare cores (Polynomial and Pandas Core) exploit the matrix representation of Remark **??**.

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 $\mathrm{rank}^d(\cdot)$ as specified in Def. 85. 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, evidenceCo
```

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 Wolfram (1983) and popularized in the book Wolfram (2002). 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 37 (Encoding of the Wet Street example)**.** *For example we have*

- *Atomic variable* Rained *by*

  S(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"]]*

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 $\exp[\theta_f \cdot f]$. 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 8 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. 50 and **??**).

A connective selector (see Def. 50) is encoded by the list

$S(\circ) = [S(\circ_0), \ldots, S(\circ_{p\text{-}1})]$

and a formula selector (see Def. **??**) by

$S(\mathcal{H}) = [S(\circ_0), \ldots, S(\circ_{p\text{-}1})]$

A logical neuron of order n (see Def. 54), 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 38 (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

$S(\mathcal{A})$ *= { "neur1": [["imp","eq"],["neur2"],["Street"]] ,*
"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 $X_{\cdot}$: cVar / aVar
- Selection variables $L_{\cdot}$: sVar
- Term variables $O_{\cdot}$: 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

$$f = \langle \rho^f, e_1 \rangle [X_f] .$$

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. 54:

- 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. 54) 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. 4.8

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. 33) 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. 5.6.3) 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

B *

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, March 2016. URL http://arxiv.org/abs/1603.04467. arXiv:1603.04467 [cs].

Christian Agerbeck and Mikael Hansen. A Multi-Agent Approach to Solving NP-Complete Problems. 2008. URL https://www.semanticscholar.org/paper/A-Multi-Agent-Approach-to-Solving-NP-Complete-Agerbeck-Hansen/3762bf7893da14839e06ae000b9e04d63dac8af4.

Grigoris Antoniou, Paul Groth, Frank Van Harmelen, and Rinke Hoekstra. *A Semantic Web Primer, third edition*. The MIT Press, Cambridge (Mass.), third edition edition, August 2012. ISBN 978-0-262-01828-9.

Artur S. Avila Garcez and Gerson Zaverucha. The Connectionist Inductive Learning and Logic Programming System. *Applied Intelligence*, 11(1):59–77, July 1999. ISSN 1573-7497. doi: 10.1023/A:1008328630915. URL https://doi.org/10.1023/A:1008328630915.

Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic Tensor Networks. *Artificial Intelligence*, 303:103649, February 2022. ISSN 0004-3702. doi: 10.1016/j.artint.2021.103649. URL https://www.sciencedirect.com/science/article/pii/S0004370221002009.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor Factorization for Knowledge Graph Completion. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5184–5193, 2019. doi: 10.18653/v1/D19-1522. URL https://www.aclweb.org/anthology/D19-1522. Conference Name: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) Place: Hong Kong, China Publisher: Association for Computational Linguistics.

Albert-László Barabási. *Network Science*. Cambridge University Press, Cambridge, illustrated edition edition, July 2016. ISBN 978-1-107-07626-6.

Richard E. Bellman. *Adaptive Control Processes*. Princeton University Press, New Jersey, 1961. ISBN 978-1-4008-7466-8. Publication Title: Adaptive Control Processes.

Gregory Beylkin and Martin J. Mohlenkamp. Algorithms for Numerical Analysis in High Dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, January 2005. ISSN 1064-8275, 1095-7197. doi: 10.1137/040604959.

Alexander Bigerl, Felix Conrads, Charlotte Behning, Mohamed Ahmed Sherif, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Tentris – A Tensor-Based Triple Store. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, volume 12506, pages 56–73. Springer International Publishing, Cham, 2020. ISBN 978-3-030-62418-7 978-3-030-62419-4. doi: 10.1007/978-3-030-62419-4_4. URL https://link.springer.com/10.1007/978-3-030-62419-4_4. Series Title: Lecture Notes in Computer Science.

Peter G. Casazza, Gitta Kutyniok, and Friedrich Philipp. Introduction to Finite Frame Theory. In Peter G. Casazza and Gitta Kutyniok, editors, *Finite Frames: Theory and Applications*, pages 1–53. Birkhäuser, Boston, 2013. 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.

Andrzej Cichocki. Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions. *arXiv:1403.2048 [cs]*, March 2014. arXiv: 1403.2048.

Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and HUY ANH PHAN. Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, March 2015. ISSN 1558-0792. doi: 10.1109/MSP.2013.2297439. Conference Name: IEEE Signal Processing Magazine.

P. Clifford and J. M. Hammersley. Markov fields on finite graphs and lattices. *Unpublished*, 1971. URL https://ora.ox.ac.uk/objects/uuid:4ea849da-1511-4578-bb88-6a8d02f457a6. Publisher: University of Oxford.

William Cohen, Fan Yang, and Kathryn Rivard Mazaitis. TensorLog: A Probabilistic Database Implemented Using Deep-Learning Infrastructure. *Journal of Artificial Intelligence Research*, 67:285–325, February 2020. ISSN 1076-9757. doi: 10.1613/jair.1.11944. URL https://jair.org/index.php/jair/article/view/11944.

Vin de Silva and Lek-Heng Lim. Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, January 2008. ISSN 0895-4798, 1095-7162. doi: 10.1137/06066518X.

Morris H. DeGroot. *Probability and Statistics*. PEARSON INDIA, January 2016. ISBN 978-93-325-7387-1.

Caglar Demir and Axel-Cyrille Ngonga Ngomo. DRILL- Deep Reinforcement Learning for Refinement Operators in ALC. *CoRR*, abs/2106.15373, 2021. URL https://ris.uni-paderborn.de/record/25217.

Mike Espig, Wolfgang Hackbusch, Thorsten Rohwedder, and Reinhold Schneider. Variational calculus with sums of elementary tensors of fixed rank. *Numerische Mathematik*, 122(3):469–488, November 2012. ISSN 0945-3245. doi: 10.1007/s00211-012-0464-x.

Antonio Falco and Wolfgang Hackbusch. On Minimal Subspaces in Tensor Representations. *Foundations of Computational Mathematics*, 12:765–803, December 2012. doi: 10.1007/s10208-012-9136-6.

Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. 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.

Python Software Foundation. Python Language Reference, version 3.13.2, April 2025. URL https://docs.python.org/3/.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422, Rio de Janeiro Brazil, May 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488425. URL https://dl.acm.org/doi/10.1145/2488388.2488425.

Varun Ganapathi, David Vickrey, John Duchi, and Daphne Koller. Constrained approximate maximum entropy learning of Markov random fields. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, pages 196–203, Arlington, Virginia, USA, July 2008. AUAI Press. ISBN 978-0-9749039-4-1.

Artur d'Avila Garcez, Marco Gori, Luis C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning, May 2019. URL http://arxiv.org/abs/1905.06088. arXiv:1905.06088 [cs].

Patrick Gelß, Stefan Klus, Jens Eisert, and Christof Schütte. Multidimensional Approximation of Nonlinear Dynamical Systems. *Journal of Computational and Nonlinear Dynamics*, 14(6):061006–061006–12, April 2019. ISSN 1555-1415. doi: 10.1115/1.4043148.

Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press, September 2019. ISBN 978-0-262-53868-8.

Rafael Gillmann. *0/1-Polytopes: Typical and Extremal Properties*. PhD thesis, February 2007. URL https://depositonce.tu-berlin.de/items/urn:nbn:de:kobv:83-opus-14695.

Vincenzo Nicosia Giovanni Russo Vito Latora. *Complex Networks: Principles, Methods and Applications. With 58 exercises*. Cambridge University Press, Cambridge, United Kingdom ; New York, NY, September 2017. ISBN 978-1-107-10318-4.

Ivan Glasser, Ryan Sweke, Nicola Pancotti, Jens Eisert, and Ignacio Cirac. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

Alex Goeßmann, Ingo Roth, Gitta Kutyniok, Michael Götte, Ryan Sweke, and Jens Eisert. Tensor network approaches for data-driven identification of non-linear dynamical laws. In *Advances in Neural Information Processing Systems - First Workshop on Quantum Tensor Networks in Machine Learning*, page 21, 2020.

Alex Christoph Goeßmann. *Uniform Concentration of Tensor and Neural Networks: An Approach towards Recovery Guarantees*. PhD Thesis, Technische Universität Berlin, Berlin, 2021. URL https://depositonce.tu-berlin.de/handle/11303/15990. Accepted: 2021-12-30T15:00:58Z.

Lars Grasedyck. Hierarchical Singular Value Decomposition of Tensors. *SIAM J. Matrix Analysis Applications*, 31: 2029–2054, January 2010. doi: 10.1137/090764189.

W. Hackbusch and S. Kühn. A New Scheme for the Tensor Representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, October 2009. ISSN 1531-5851.

Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer Series in Computational Mathematics. Springer-Verlag, Berlin Heidelberg, 2012. ISBN 978-3-642-28026-9. doi: 10.1007/978-3-642-28027-6.

Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. Springer, Berlin, Heidelberg, 1993rd edition edition, October 1993. ISBN 978-3-540-56852-0.

Frank L. Hitchcock. The Expression of a Tensor or a Polyadic as a Sum of Products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927. ISSN 1467-9590. doi: https://doi.org/10.1002/sapm192761164.

Sepp Hochreiter. Toward a broad AI. *Communications of the ACM*, 65(4):56–57, January 2022. ISSN 0001-0782, 1557-7317. doi: 10.1145/3512715. URL https://dl.acm.org/doi/10.1145/3512715.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, Jose Emilio Labra Gayo, Roberto Navigli, and Sebastian Neumaier. *Knowledge Graphs*. Springer, Cham, 1st edition edition, November 2021. ISBN 978-3-031-00790-3.

Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120(4):701–731, April 2012. ISSN 0029-599X, 0945-3245. doi: 10.1007/s00211-011-0419-7. URL http://link.springer.com/10.1007/s00211-011-0419-7.

Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A Patterson. TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ISCA '23, pages 1–14, New York, NY, USA, June 2023. Association for Computing Machinery. ISBN 979-8-4007-0095-8. doi: 10.1145/3579371.3589350. URL https://dl.acm.org/doi/10.1145/3579371.3589350.

Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, August 2009. ISSN 0036-1445, 1095-7200. doi: 10.1137/07070111X.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, Mass., 1. edition edition, July 2009. ISBN 978-0-262-01319-2.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Neural Class Expression Synthesis, December 2022. URL http://arxiv.org/abs/2111.08486. arXiv:2111.08486 [cs].

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Neural Class Expression Synthesis. In Catia Pesquita, Ernesto Jimenez-Ruiz, Jamie McCusker, Daniel Faria, Mauro Dragoni, Anastasia Dimou, Raphael Troncy, and Sven Hertling, editors, *The Semantic Web*, volume 13870, pages 209–226. Springer Nature Switzerland, Cham, 2023. 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. Series Title: Lecture Notes in Computer Science.

J. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, December 2011. ISBN 978-0-8218-6907-9 978-0-8218-8481-2 978-0-8218-8483-6 978-1-4704-0923-4.

Jens Lehmann, Sören Auer, Lorenz Bühmann, and Sebastian Tramp. Class expression learning for ontology engineering. *Journal of Web Semantics*, 9(1):71–81, March 2011. ISSN 1570-8268. doi: 10.1016/j.websem.2011.01.001. URL https://www.sciencedirect.com/science/article/pii/S1570826811000023.

David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, illustrated edition edition, September 2003. ISBN 978-0-521-64298-9.

Alan K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, February 1977. ISSN 0004-3702. doi: 10.1016/0004-3702(77)90007-8. URL https://www.sciencedirect.com/science/article/pii/0004370277900078.

Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, 328:104062, March 2024. ISSN 0004-3702. doi: 10.1016/j.artint.2023.104062. URL https://www.sciencedirect.com/science/article/pii/S0004370223002084.

John McCarthy. Programs with Common Sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91. Her Majesty's Stationary Office, London, 1959. URL http://www-formal.stanford.edu/jmc/mcc59.html.

Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994. URL https://www.sciencedirect.com/science/article/pii/0743106694900353. Publisher: Elsevier.

Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts London, England, March 2022. ISBN 978-0-262-04682-4.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 809–816, Madison, WI, USA, June 2011. Omnipress. ISBN 978-1-4503-0619-5.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1):11–33, January 2016. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2015.2483592. URL https://ieeexplore.ieee.org/document/7358050/.

Goran S. Nikolić, Bojan R. Dimitrijević, Tatjana R. Nikolić, and Mile K. Stojcev. A Survey of Three Types of Processing Units: CPU, GPU and TPU. In *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, pages 1–6, June 2022. doi: 10.1109/ICEST55168.2022.9828625. URL https://ieeexplore.ieee.org/document/9828625.

Román Orús. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, September 2019. ISSN 2522-5820. doi: 10.1038/s42254-019-0086-7.

I. V. Oseledets and E. E. Tyrtyshnikov. Breaking the Curse of Dimensionality, Or How to Use SVD in Many Dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759, January 2009. ISSN 1064-8275. doi: 10.1137/090748330. Publisher: Society for Industrial and Applied Mathematics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv:1912.01703 [cs].

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, s.l., September 1988. ISBN 978-1-55860-479-7.

Judea Pearl. *Causality: Models, Reasoning and Inference. Ausgezeichnet: ACM Turing Award for Transforming Artificial Intelligence 2011*. Cambridge University Press, Cambridge New York, NY Port Melbourne New Delhi Singapore, 2 edition, November 2009. ISBN 978-0-521-89560-6.

Roger Penrose. *Spinors and Space-Time: Volume 1, Two-Spinor Calculus and Relativistic Fields*. Cambridge University Press, Cambridge, February 1987. ISBN 978-0-521-33707-6.

D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Information & Computation*, 7(5):401–430, July 2007. ISSN 1533-7146.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, February 2006. ISSN 0885-6125, 1573-0565. doi: 10.1007/s10994-006-5833-1. URL http://link.springer.com/10.1007/s10994-006-5833-1.

Elina Robeva and Anna Seigal. Duality of graphical models and tensor networks. *Information and Inference: A Journal of the IMA*, 8(2):273–288, June 2019. ISSN 2049-8772. doi: 10.1093/imaiai/iay009.

Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, reprint edition edition, January 1997. ISBN 978-0-691-01586-6.

Sebastian Rudolph. Foundations of Description Logics. In Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter Patel-Schneider, editors, *Reasoning Web. Semantic Technologies for the Web of Data: 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures*, pages 76–136. Springer, Berlin, Heidelberg, 2011. ISBN 978-3-642-23032-5. doi: 10.1007/978-3-642-23032-5_2. URL https://doi.org/10.1007/978-3-642-23032-5_2.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach, Global Edition: A Modern Approach, Global Edition*. Pearson, Boston, 4 edition, May 2021. ISBN 978-1-292-40113-3.

Chiaki Sakama, Katsumi Inoue, and Taisuke Sato. Linear Algebraic Characterization of Logic Programs. In Gang Li, Yong Ge, Zili Zhang, Zhi Jin, and Michael Blumenstein, editors, *Knowledge Science, Engineering and Management*, volume 10412, pages 520–533. Springer International Publishing, Cham, 2017. 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. Series Title: Lecture Notes in Computer Science.

Aaron Sander, Maximilian Fröhlich, Martin Eigel, Jens Eisert, Patrick Gelß, Michael Hintermüller, Richard M. Milbradt, Robert Wille, and Christian B. Mendl. Large-scale stochastic simulation of open quantum systems, January 2025. URL http://arxiv.org/abs/2501.17913. arXiv:2501.17913 [quant-ph].

Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. Neuro-symbolic artificial intelligence: Current trends. *AI Communications*, 34(3):197–209, March 2022. ISSN 18758452, 09217126. doi: 10.3233/AIC-210084. URL https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/AIC-210084.

Taisuke Sato. A linear algebraic approach to datalog evaluation. *Theory and Practice of Logic Programming*, 17(3):244–265, May 2017. 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. Publisher: Cambridge University Press.

Luciano Serafini and Artur S. d'Avila Garcez. Learning and Reasoning with Logic Tensor Networks. In *AI*IA 2016 Advances in Artificial Intelligence: XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 – December 1, 2016, Proceedings*, pages 334–348, Berlin, Heidelberg, November 2016. Springer-Verlag. ISBN 978-3-319-49129-5. doi: 10.1007/978-3-319-49130-1_25. URL https://doi.org/10.1007/978-3-319-49130-1_25.

Shalev-Schwartz, Shai and Ben-David, Shai. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, July 2014. ISBN 978-1-107-05713-5.

C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27 (3):379–423, 1948. ISSN 1538-7305. doi: 10.1002/j.1538-7305.1948.tb01338.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x.

Helmut Simonis. Sudoku as a constraint problem. In *CP Workshop on modeling and reformulating Constraint Satisfaction Problems*, volume 12, pages 13–27. Citeseer Sitges, Spain, 2005. URL https://ai.dmi.unibas.ch/_files/teaching/fs21/ai/material/ai26-simonis-cp2005ws.pdf.

Edwin Stoudenmire and David J Schwab. Supervised Learning with Tensor Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4799–4807. Curran Associates, Inc., 2016.

Michel Talagrand. *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*. Springer, Berlin, Heidelberg, 2014. ISBN 978-3-642-54074-5. doi: 10.1007/978-3-642-54075-2.

Geoffrey G. Towell and Jude W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70 (1):119–165, October 1994. ISSN 0004-3702. doi: 10.1016/0004-3702(94)90105-8. URL https://www.sciencedirect.com/science/article/pii/0004370294901058.

Théo Trouillon and Maximilian Nickel. Complex and Holographic Embeddings of Knowledge Graphs: A Comparison, July 2017. URL http://arxiv.org/abs/1707.01475. arXiv:1707.01475 [cs, stat].

Efthimis Tsilionis, Alexander Artikis, and Georgios Paliouras. A Tensor-Based Formalization of the Event Calculus. In *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*, pages 3584–3592, Jeju, South Korea, August 2024. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/397. URL https://www.ijcai.org/proceedings/2024/397.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, New York, NY, 1st edition edition, September 2018. ISBN 978-1-108-41519-4.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2019. ISBN 978-1-108-49802-9. doi: 10.1017/9781108627771.

Martin J. Wainwright and Michael Irwin Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc, 2008. ISBN 978-1-60198-184-4.

Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, July 1983. doi: 10.1103/RevModPhys.55.601. URL https://link.aps.org/doi/10.1103/RevModPhys.55.601. Publisher: American Physical Society.

Stephen Wolfram. *A New Kind of Science*. Wolfram Media, Champaign (Ill.), illustrated edition edition, May 2002. ISBN 978-1-57955-008-0.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. arXiv, August 2015. URL http://arxiv.org/abs/1412.6575. arXiv:1412.6575 [cs].

Günter M. Ziegler. Lectures on 0/1-Polytopes. In Gil Kalai and Günter M. Ziegler, editors, *Polytopes — Combinatorics and Computation*, pages 1–41. Birkhäuser, Basel, 2000. 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.

Günter M. Ziegler. *Lectures on Polytopes*. Springer, New York, 1995th edition edition, October 2013. ISBN 978-0-387-94365-7.