Funded by the
European Union

The Tensor Network Approach to Efficient and Explainable AI

Alex Goessmann

Technical report on the ENEXA project at DATEV eG

September 30, 2025

to my son Erik

Abstract

Recent advances in artificial intelligence, particularly in large language models, have achieved impressive performance but remain limited in efficiency and explainability, restricting their practical and trustworthy application. Training and inference of such black-box models have driven the development of extensive hardware and software infrastructures optimized for large-scale, parallelizable linear algebra workloads. In this work, we propose an alternative approach that leverages this infrastructure to build efficient and interpretable models rather than black-box systems. Specifically, we exploit the mathematical structure of tensor networks, a framework with deep roots in the logical and probabilistic traditions of artificial intelligence.

Tensors naturally arise in AI as factored representations of complex systems, but their full representations suffer from the curse of dimensionality. Tensor network decompositions mitigate this issue, enabling tractable reasoning while preserving essential structure. Classical logical and probabilistic AI approaches have historically employed related sparsity-inducing decompositions: logical approaches emphasize sparse syntactic descriptions, while probabilistic approaches exploit conditional independencies to develop sparse graphical models. In this work, we unify these sparsity mechanisms within the tensor network formalism and develop feasible reasoning algorithms based on tensor network contractions.

The first part of this work reviews the classical logical and probabilistic AI traditions in the formalism of tensor networks. Building on this foundation, the second part introduces a neuro-symbolic framework, Hybrid Logic Networks, integrating these approaches. The third part explores tensor network contraction schemes for reasoning and calculus in more detail.

We implement these concepts in the open-source python library tnreason, designed modularly to specify reasoning tasks as tensor network operations, which are then executed on existing AI software frameworks. This enables modern AI infrastructure to support reasoning that is both efficient and interpretable.

Acknowledgements

This project has been funded by the European Union's Horizon 2020 programme project ENEXA (Project 101070305). For close collaboration on the DATEV use case of the ENEXA project I would like to thank Maria Schnödt-Fuchs, Nadja Heimhuber, Martin Reimer and Konstantin Ehrmann. I would like to thank Axel-Cyrille Ngonga Ngomo, Paul Groth, Andreas Both, Alexander Artikis for the organization of the ENEXA project. For discussions during the initial phase of this work I would like to thank Efthimis Tsilonis from NCSR Demokritos. I want to further thank Alexander Bigerl, Caglar Demir and Nikolaos Karalis for discussions on the integration of tentris into the tnreason framework. I would like to thank Martin Braun, Tanja Döhler, Jaroslaw Schaller, Botond Simon and Jeanette Lorenz for collaboration within the QUAST project and Martin Eigel and Carsten Blank for discussions on the integration into the QROM project. Further, I would like to thank Janina Schütte and Maximilian Fröhlich for proofreading.

Contents

Introduction

Artificial intelligence is a long-standing vision, 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 infered 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 model's inference output.

The need for efficiency stems more from economic concerns on the resource demand of training and inferring a model. Tensors naturally represent states of systems with multiple variables, both in logical and probabilistic approaches towards artificial intelligence. However, even for 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 for neuro-symbolic artificial intelligence.

1 Background

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

1.1 Logic and Explainability in AI

The logical tradition of artificial intelligence is motivated by the resemblance of human thought in logics [McC59]. Historic approaches to artificial intelligence have focused on models by vast knowledge bases and inference by logical reasoning. The main problem 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 as Inductive Logic Programming [MD94]. Along that line the Amie method [Gal+13] has been developed to learn Horn clauses using a refinement operator. Class Expression Learning [Leh+11] is a more recent approach to assist in the design of reasoning capabilities in Knowledge Graphs. However, 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 [DN21] and neural networks [Kou+22; Kou+23].

Logical approaches are still dominant in the description of data. Here the semantic web initiative developed data storage formats based on Knowledge Graphs [Ant+12; Hog+21], which describe structured data based on description logic.

Towards extending the practical usage of logics, the field of Statistical Relational AI [Nic+16; GT19] studies statistical models of logical relations. This directly treats uncertainty and therefore unifies logics with statistical approaches. These aims have more recently reframed as neuro-symbolic AI [Hoc22; Sar+22], with close relations to statistical relational AI [Mar+24]. Neuro-symbolic AI focuses on the unification of the neural and the symbolic paradigm [Gar+19], where early approaches are [TS94; AZ99]. 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 layers. These decompositions provide both expressive and efficiently inferable model architectures. While modern black-box AI focuses on large neural networks, whose size prevents human understanding of the inference process, neuro-symbolic AI aims at a re-implementation of the symbolic paradigm into such architectures.

1.2 Tensor Networks in AI

Decomposition schemes of tensors have been developed in numerics to efficiently operate in high-dimensional tensor spaces [Hac12] and to avoid the curse of dimensionality [Bel61]. Each decomposition scheme has a graphical depiction, as we will introduce in Chapter 2.5, and decompositions are therefore referred to as networks. The Tucker decomposition schemes, originally introduced in [Hit27], suffered from exponential increases of the degrees of freedom with the tensor order. The CP format (see Chapter 15) can in principle establish storage with demand growing linear with the tensor order. Sets of tensors with fixed CP rank are however not closed [BM05] and approximation problems are often ill posed [SL08]. The Tensor Train decomposition [OT09], which appears in quantum mechanics as matrix product states [Per+07], overcomes these numerical problems [HRS12]. Hierarchical Tucker decompositions [HK09] are generalizations of tensor train decompositions, which have useful properties for tensor approximations [Gra10; FH12].

Tensors are used in the processing of big data [Cic14] and in many-body physics [Orú19]. In addition, pioneering approaches have been made to exploit them in the data-driven identification of governing equations [Gel+19; Goe+20], more general supervised learning [SS16] and the simulation of noisy quantum mechanics [San+25]. The duality between tensor networks and graphical models has been first discussed in [RS19] and motivated further expressivity studies such as [Gla+19]. Tensor networks have further been applied for batch logical inference [SIS17; Sat17; TAP24]. Whereas these are conceptually 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 led to TensorLog [CYM20], Real Logic [SG16] and to Logical Tensor Networks [Bad+22].

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 embedding schemes that encode relations between the objects in a latent space. The specific approaches distinguish between the format used, where [NTK11] and [BAH19] used Tucker decompositions, [Yan+15] the CP decomposition and [TN17] complex extensions. Beyond embeddings, tensor based storage of knowledge graphs has recently shown tremendous improvements in querying knowledge graphs [Big+20]. 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 [Aba+16] and PyTorch [Pas+19]. 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) [Nik+22; Jou+23]. Both the dedicated software and hardware design exploit the parallelization potential rooted in the contraction formalism of tensor networks. Besides these developments there exist several experimental libraries dedicated to the tensor-train tensor format [SH17; Wol24; Gel25; Pul+25].

1.3 Representation Schemes of Systems

We start with ontological commitments in the description of a system and follow the book [RN21] distinguishing atomic, factored and structured representations. While in atomic representation, the states of a system 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 this representation scheme extends 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 lose much of the structure of the representation, which we would like to exploit in reasoning processes.

A more generic representation of systems are structured representation. Structured representations involve objects of differing numbers and relations between them. As a consequence the numbers of variables can differ depending on the state of a system. This poses a challenge to the tensor representation, since a fixed number of variables is required to motivate a tensor space of representations. There are approaches to circumvent these difficulty by the development of template models such as Markov Logic Networks [RD06], which are instantiated on systems with differing number of objects. We will discuss those in Chapter 11.

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 epistemological 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 approaches to reasoning, probabilistic approaches thus bear a more expressive modelling.

2 Structure of the work

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

2.1 Part I: Foundations

The probabilistic and logical approaches towards artificial intelligence are reviewed in the tensor network formalism. In this part, we restrict the discussion to atomic and factored representations of systems. In probability theory (see Chapter 2 and Chapter 3), 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 normalized, 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 2 a tensor network representation, which structure is exploited for inference in Chapter 3. In logics (see Chapter 4),

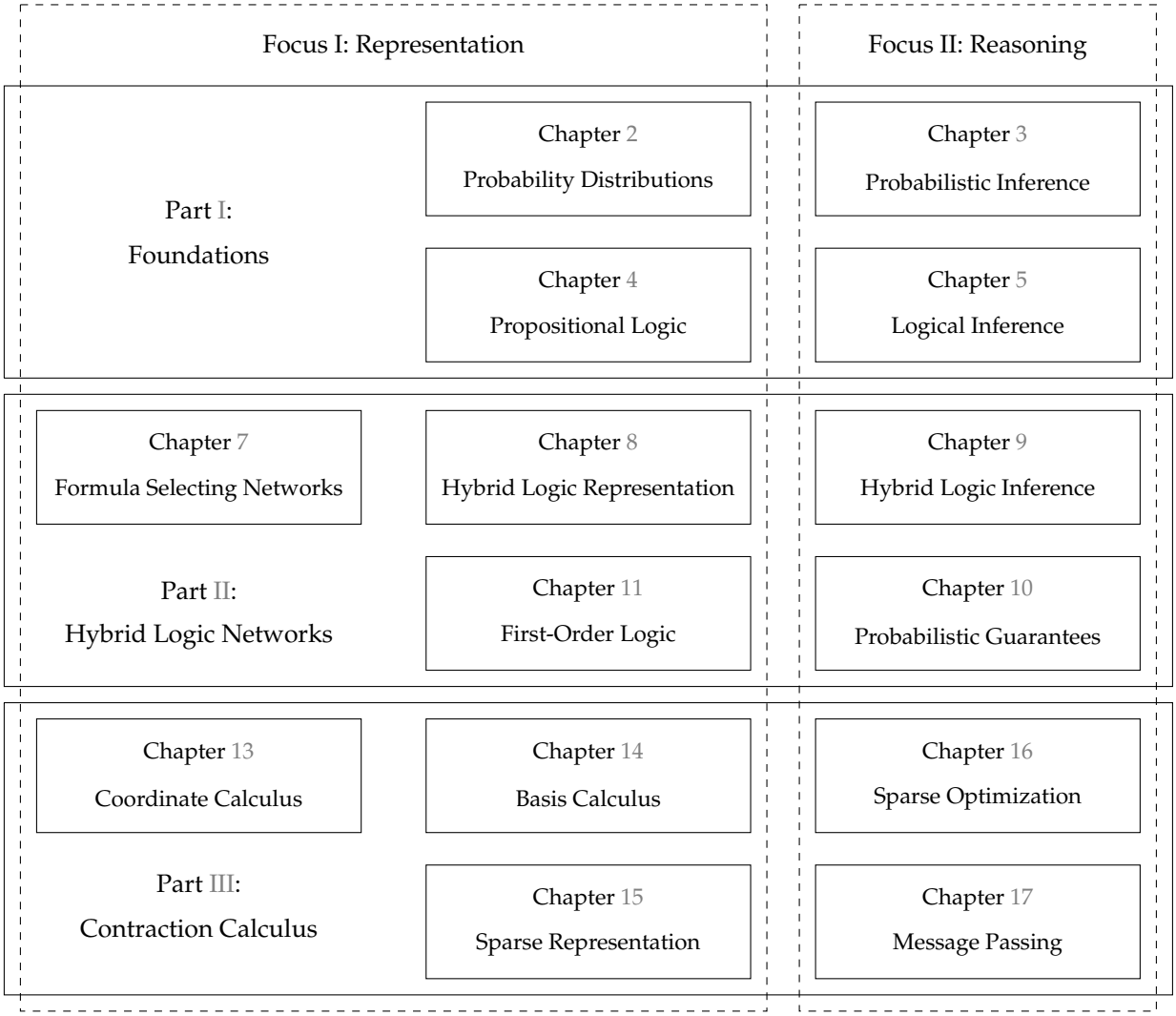


Figure 1: Sketch of the structure of this work. We assign the chapters to three parts and two foci. The parts distinguish the coarse topics of this work into classical, neuro-symbolic approaches and the applied contraction calculus. The assigned foci indicate whether the chapter orients more onto a representation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

we motivate boolean tensors as a natural representation of propositional semantics. Logical entailment is then in Chapter 5 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 4 and use them for efficient logical inference algorithms in and Chapter 5.

2.2 Part II: Hybrid Logic Networks

Motivated by the classical approaches we apply the tensor network formalism towards learning and inferring neuro-symbolic models, which we call Hybrid Logic Networks. We understand the decomposition of tensors into networks as an implementation of the neural paradigm of artificial intelligence. Further, the symbolic paradigm is eminent in the interpretation of tensor

networks using logical syntax, and enables the human-interpretable verbalization of learned models. Motivated by this central thoughts, we present vast classes of interpretable models in Chapter 8, 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 9 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 7 to represent hypothesis classes for new features. While these approaches rely on propositional logics, in Chapter 11 we extend towards more expressive first-order logic. With knowledge graphs serving as examples we 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 10 we further derive statistical guarantees for these learning methods given random data, based on probabilistic bounds on uniform concentration events.

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 7 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.

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 2.5. 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 2) and from logical syntax (see Chapter 4). Towards neuro-symbolic AI, we provide in Chapter 7 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 8 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.

2.5 Focus 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 3), contraction of logical formulas are model counts central to the formalism of logical entailment (see Chapter 5). We will further exploit them to calculate queries in first-order logic such as on knowledge graphs (see Chapter 11). 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 10. 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.

Notation and Basic Concepts

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

1 Categorical Variables and Representations

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

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

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

of cardinality m.

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

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

2 Tensors

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

Definition 0.3 (Tensor) Let there be numbers $m_k \in \mathbb{N}$ for $k \in [d]$ and categorical variables X_k taking their values in $[m_k]$. A Tensors $\tau[X_0, \dots, X_{d-1}]$ of order d and leg dimensions m_d is defined through its coordinates

$$\tau[X_0 = x_0, \dots, X_{d-1} = x_{d-1}] \in \mathbb{R}$$

for index tuples

$$x_0, \dots, x_{d-1} \in \bigtimes_{k \in [d]} [m_k] .$$

Tensors $\tau[X_0, \dots, X_{d-1}]$ are elements of the tensor space

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

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

multiplication.

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

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

Example 0.4 (Trivial Tensor) The trivial tensor

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

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

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

◊

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

Definition 0.5 We call a tensor $\tau\left[X_{[d]}\right]$ boolean, when $\mathrm{im}(\tau) \subset [2]$, i.e. all coordinates are either 0 or 1.

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

Definition 0.6 (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 $\epsilon_x[X]$ by the coordinates

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

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

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

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

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

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

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

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

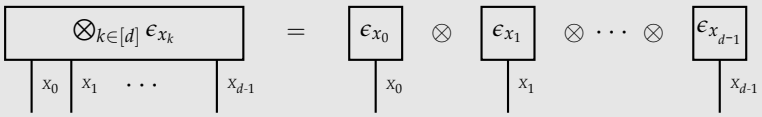
Definition 0.7 (One-hot encodings to Factored Representations) Let there be a factored system defined by a tuple $(X_0, \dots, X_{d-1})$ of variables taking values in $\bigtimes_{k\in[d]}[m_k]$. The one-hot encoding of its states is the tensor product of the one-hot encoding to each categorical variables, that is the map

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

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

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

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



In Chapter 13 we will investigate the image of ϵ in more detail and show that it is an orthonormal basis of the tensor space $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$.

Remark 0.8 (Flattening of Tensors) The use of 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. ◊

3 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.

3.1 Graphical Illustrations

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

Definition 0.9 A hypergraph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$, where each hyperedge $e \in \mathcal{E}$ is a subset of the nodes $\mathcal{V}$. A directed hypergraph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that each hyperedge $e \in \mathcal{E}$ is the tuple of two disjoint sets $e^{\text{in}}, e^{\text{out}} \subset \mathcal{V}$, that is

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

We will use the standard visualization by factor graphs as a diagrammatic illustration of sets of tensors, where tensors are represented by block nodes and each axis assigned with by a categorical variable X_k represented by a node, see Figure 1a). Different simplifications of these factor graph

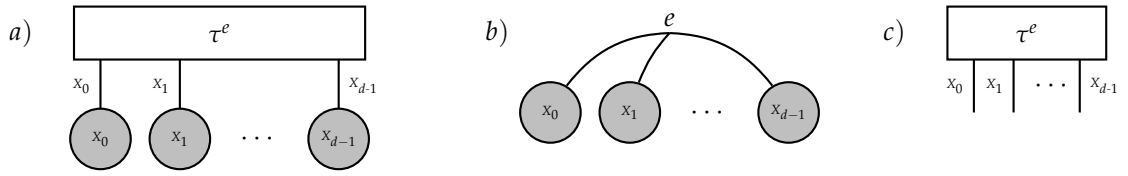


Figure 1: Depiction of Tensors a) As a factor in a factor graph, depicted by a block, and connected to categorical variables assigned to nodes. b) Highlighting only the variable dependencies by a hyperedge connecting the variables X_k to each axis $k \in [d]$. c) Highlighting the tensor by a blockwise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

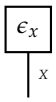
depictions have been evolved in different research fields. In the tradition of graphical models, which started with the work [Pea88], the categorical variables are highlighted and the tensor blocks just depicted by hyperedges. To depict dependencies with causal interpretations, the edges are further decorated by directions in the depiction of Bayesian networks, see for example [Pea09].

In the tensor network community on the other hand, a simplification scheme highlighting the tensors as blocks and omitting the depiction of categorical variables has been evolved. The variables, or sometimes their index or dimension, are then directly assigned to the lines depicting the axes of the tensor blocks. This depiction scheme has been established in the literature as wiring diagrams (see [Lan11] and dates back at least to the work [Pen87].

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

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

To depict vector calculus and its generalizations, we will apply the graphical notation (mainly version b) introduced in Chapter 2.5. Along this line, we represent vectors and their generalization to tensors by blocks with legs representing its indices. The basis vectors being one-hot encodings of states are in this scheme represented by



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

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

Definition 0.10 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

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

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

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

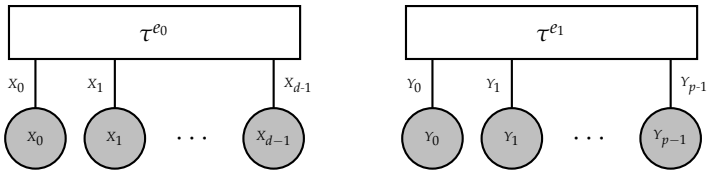
the Tensor Network of the decorated hypergraph $\mathcal{G}$. The set of tensor networks on $\mathcal{G}$, such

Figure 2: 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.

that all tensors have non-negative coordinates, is denoted by $\mathcal{T}^{\mathcal{G}}$.

3.2 Tensor Product

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



Definition 0.11 (Tensor Product) Let there be two tensor

$$\tau^{e_0}\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k} \quad \text{and} \quad \tau^{e_1}\left[Y_{[p]}\right] \in \bigotimes_{l\in[p]} \mathbb{R}^{m_l}$$

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

$$\left\langle \tau^{e_0}\left[X_{[d]}\right], \tau^{e_1}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}, Y_{[p]}\right] \in \left(\bigotimes_{k\in[d]} \mathbb{R}^{m_k}\right) \otimes \left(\bigotimes_{l\in[p]} \mathbb{R}^{m_l}\right)$$

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

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

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

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

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

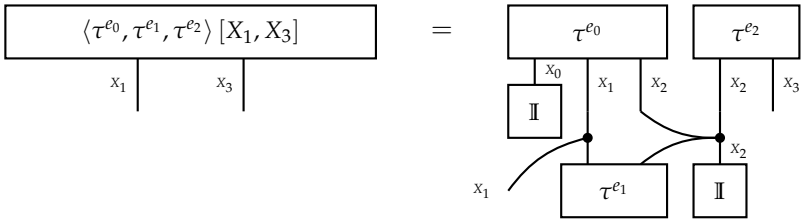


Figure 3: Example of a tensor network contraction of all but the variables X_1, X_3. Contraction of variables can always be depicted by closing the open legs with trivial tensors $\mathbb{I}$ performing index sums.

When the tensor $\tau^{e_1}\left[Y_{[p]}\right]$ coincides with the trivial tensor $\mathbb{I}\left[Y_{[p]}\right]$ (see Example 0.4), we further make a notation convention to omit that tensor, that is

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

3.3 Generic Contractions

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

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

Definition 0.12 Let $\tau^{\mathcal{G}}$ 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

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

defined coordinatewise by the sum

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

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

To ease notation, we will often omit the set notation by brackets $\{\cdot\}$ and specify the tensors to be contracted with the delimiter "," (see e.g. Example 0.14).

Remark 0.13 (Alternative Notations) Contractions can also denoted by the Einstein summations of the indices along connected edges, understood as scalar product in each subspace. This is as in Def. 0.12, just omitting the sums. We found it useful in this work to do the diagrammatic representation instead, since it offers a better possibility to depict hierarchical arrangements of shared variables. ♢

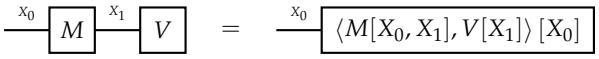
Further notations without usage of axis variables are mode products (see [KB09; Hac12; Cic+15]), often denoted by the operation $\times_n$. With our more generic variable-based notations, we can capture these more specific contractions by coloring the tensor axes, that is assignment of axis variables.

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

Example 0.14 Matrix Vector Products The matrix vector product is a special case of tensor contractions, where a matrix $M[X_0, X_1]$ shares a categorical variable with a vector $V[X_1]$. When leaving the variable unique to the matrix open we get the matrix vector product as

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

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



◊

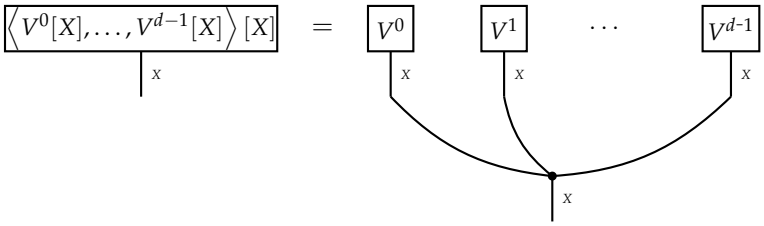
Example 0.15 Hadamard products of vectors A node appearing in arbitrarily many hyperedges denotes a Hadamard product of the axis of the respective decorating tensors. To give an example, let $V^k[X] \in \mathbb{R}^m$ be vectors for $k \in [d]$. Their hadamard product is the vector

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

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:



◊

3.4 Decompositions

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

Definition 0.16 A Tensor Network Decomposition of a tensor $\tau [X_{\mathcal{V}}]$ is a Tensor Network $\tau^{\mathcal{G}}$ such that

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

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

3.5 Directed Tensors and Normalizations

Directionality represents constraints on the structure of tensors, namely that the sum over outgoing trivializes the tensor.

Definition 0.17 A Tensor

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

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

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

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

By the normalization operation, tensors are turned into directed tensors.

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

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

The normalization of an on $\mathcal{V}^{\text{in}} \subset \mathcal{V}$ normalizable tensor is the tensor

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

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

We will investigate the contractions of directed tensors in Part III, where we show in Thm. 13.12 that normalizations 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:

4 Function encoding schemes

The tensor formalism, which treats tensors as collections of coordinates, can be used to encode functions. While we introduce only the basic notation here, which will be applied later in Part I and Part II, a more detailed discussion of these representation schemes is deferred to Part III.

4.1 Coordinate encodings

The most direct encoding scheme represents real valued functions on a discrete set of states by tensors. Since it identifies each function evaluation at a state with a coordinate, we refer to it as coordinate encoding.

Definition 0.19 (Coordinate encoding of real-valued functions) Let there be a factored representation

of a system by variables $X_{[d]}$ and a real-valued function

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

on the systems state set. Then the coordinate encoding of q is the tensor $\chi^q\left[X_{[d]}\right]$ with the coordinates

$$\chi^q\left[X_{[d]} = x_{[d]}\right] = q\left(x_{[d]}\right)$$

for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$.

To ease the notation, we will denote the coordinate encoding of a real-valued function by

$$q\left[X_{[d]}\right] \coloneqq \chi^q\left[X_{[d]}\right] .$$

For a more detailed discussion of coordinate encodings, see Chapter 13.

4.2 Basis encodings

Let us now encode functions between the state sets of systems in factored representation. The scheme is described in more generality and detail (encoding of subsets and relations) in Chapter 14, see Def. 14.8.

Definition 0.20 (Basis encoding of maps between state sets) Let there be two systems with factored representations by variables $X_{[d]}$ and $Y_{[p]}$, and a map

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

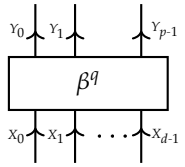
between there state sets. Then the basis encoding of q is a tensor

$$\beta^q\left[Y_{[p]}, X_{[d]}\right] \in \left(\bigotimes_{l\in[p]} \mathbb{R}^{m_l}\right) \otimes \left(\bigotimes_{k\in[d]} \mathbb{R}^{m_k}\right)$$

defined by

$$\begin{aligned}&\beta^q\left[Y_{[p]}, X_{[d]}\right]\\ &\quad= \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[m_k]} \epsilon_{q\left(x_{[d]}\right)}\left[Y_{[p]}\right] \otimes \epsilon_{x_{[d]}}\left[X_{[d]}\right] .\end{aligned}$$

Basis encodings are directed tensors (see Thm. 14.10) and are thus depicted as decorations of directed edges in hypergraphs:



To extend the basis encoding scheme to represent any function on the state set of a system in factored representation, we now define enumeration variables (for more detail see Chapter 14).

Definition 0.21 (Set enumeration variables) We say that an arbitrary finite 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 arbitrary function

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

and an enumeration of $\mathcal{U}$ by the variable $O_{\mathcal{U}}$, we define the basis encoding of q as

$$\beta^q \left[O_{\mathcal{U}}, X_{[d]}\right] \coloneqq \beta^{I^{-1}\circ q} \left[O_{\mathcal{U}}, X_{[d]}\right] .$$

We further define the basis encoding to vector valued functions

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

using multiple variables $O_{[p]} = \{O_0, \ldots, O_{p-1}\}$ as

$$\beta^{\mathcal{S}} \left[O_{[p]}, X_{[d]}\right] = \left\langle \{\beta^{s_l} \left[O_l, X_{[d]}\right] : l \in [p]\} \right\rangle \left[O_{[p]}, X_{[d]}\right] .$$

Here we denote for $l \in [p]$ by s_l the restriction of $\mathcal{S}$ to the coordinate l of the image.

4.3 Selection encodings

We now turn to tensor-valued functions, for which we introduce the selection encodings next. We call these tensor representation selection encodings, since the tensor structure of the image is captured by additional selection variables.

Definition 0.22 (Selection encoding of tensor-valued functions) Let there be a system with factored representation by the variables $X_{[d]}$, a tensor space $\bigotimes_{s\in[n]} \mathbb{R}^{p_s}$ described by categorical variables $L_0, \ldots, L_{n-1}$ and a tensor-valued function

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

The selection encoding of q is the tensor

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

defined as the sum

$$\sigma^q \left[X_{[d]}, L_{[n]}\right] = \sum_{x_{[d]} \in \times_{k\in[d]} [m_k]} \epsilon_{x_{[d]}} \left[X_{[d]}\right] \otimes q\left(x_{[d]}\right) \left[L_{[n]}\right] .$$

We exploit the selection encoding formalism to further define basis encodings of tensor-valued

functions. To this end, we enumerate the coordinates

$$\left\{ \sigma^q \left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]} \right] \,:\, x_{[d]} \in \bigtimes_{k \in [d]} [m_k],\, l_{[n]} \in \bigtimes_{s \in [n]} [p_s] \right\}$$

by an interpretation variable O_q and a map I_q. Let us construct a function $I_q^{-1} \circ \sigma^{X_{[d]}, L_{[n]}}$ by

$$\left(I_q^{-1} \circ \sigma^{X_{[d]}, L_{[n]}} \right) \left(x_{[d]}, l_{[n]} \right) = I_q^{-1}(\sigma^q \left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]} \right]) .$$

The basis encoding of σ^q is then defined as the basis encoding of this function, that is

$$\beta^{\sigma^q} \left[O_q, X_{[d]}, L_{[n]} \right] := \beta^{I_q^{-1} \circ \sigma^{X_{[d]}, L_{[n]}}} \left[O_q, X_{[d]}, L_{[n]} \right] .$$

This encoding scheme differs from the basis encoding of q, which has been defined in the previous section.

Part I

Foundations

1

Introduction to Part I

The tensor formalism naturally captures factored system representations, where states are determined by assignments to a fixed, finite set of variables. Given these modelling assumptions, we review in Part I based on the tensor formalism the two main paradigms of artificial intelligence to encode knowledge about a system:

- **Probability theory:** To each state we encode by a real number in $[0, 1]$ its probability given our knowledge about a system. A probability distribution is therefore a tensor, which coordinates are in $[0, 1]$ and sum to 1.
- **Propositional logic:** To each state we encode by a boolean its possibility given our knowledge about a system. A propositional knowledge base is therefore a tensor with boolean coordinates.

1.1 Sparse Representation and Interpretability

The interpretability of machine learning models describing probabilistic and logical knowledge about a system is a central aspect of artificial intelligence. In this work, we describe synergies between the interpretability and the sparse representation of a model by tensor networks. Both the probabilistic and the logic approaches provide a human-understandable interface to machine learning:

- **Graphical representations in probability theory:** Probability distributions are interpreted based on the encoded conditional independencies between their categorical variables. In graphical models, the variable dependence structure is commonly represented by graphs, where variables are assigned to nodes and the edges represent dependencies. While generic probability distributions correspond with dense graphs, the central sparsity mechanism of probability theory relies on sparse graphs. Graphical models on sparse graphs are both representable by least resource demand and more accessible for human interpretation. Although variable dependencies provide coarse interpretations of a probability distributions, they do not offer a precise description of the distribution, since multiple distributions coincide in their variable dependencies.
- **Syntactical representations in propositional logic:** Compared with probability theory, logical theories are more ambitious in providing human accessible interpretations. To this end, knowledge bases are verbalizable by logical syntax which is interpretable by humans. Categorical variables have leg dimension 2 and are understood as atomic formulas, where the first index is interpreted as indicating a False atomic formula and the second as a True atomic statement. More complex formulas, or propositional knowledge bases, are constructed by recursive combinations of the atomic formulas by logical connectives. This construction is reflected in the syntactical representation of knowledge bases. Compared with graphical descriptions of probability distributions, syntactical descriptions provide a precise description of the knowledge base with no representation freedom left.

Both the graphical and syntactical representation schemes show a synergy between the sparse representation of the knowledge about a system and the interpretability of that knowledge. In

Part I we investigate two mechanisms to identify tensor network decompositions of probability distributions:

- **Independence mechanism:** Conditional independence of random variables is a concept of probability theory and describes the absence of correlations between variables when conditioning on certain states of further variables. As we will show in Chapter 2, the absence of correlations corresponds with the existence of elementary decompositions of tensors representing probability distributions. While each conditional independence statement reflects a specific local decomposition mechanism, collections of conditional independence statements provide global tensor network decompositions of distributions. Graphical models visualize the independence structure of distributions in their graphical representation and the Hammersley-Clifford theorem ensures a corresponding tensor network decomposition.
- **Computation mechanism:** When there are sufficient statistics providing probabilities, we construct tensor networks decompositions by computation of the statistics. Whenever the functions to be computed are compositions of functions of lower numbers of arguments, we utilize these representations to construct tensor network decompositions. This mechanism motivates the representation of distributions by tensor networks, which contain a subnetwork computing a statistic and a subnetwork activating the statistic. We call such tensor networks Computation-Activation Networks and provide representations of probability distributions and propositional knowledge bases by them. In propositional logic, statistics are provided by logical syntax as we will exploit in Chapter 4. In probability theory, we will make use of this approach in the efficient representation of sufficient statistics.

The representation schemes of probability theory and propositional logic have limitations, which can be mitigated in a unifying tensor network framework. While probability theory does not provide a human-understandable intuition about generic events, propositional syntax describes such events and is accessible for human interpretation. Propositional logic on the other hand has a limited expressivity, since it cannot express generic uncertainties about states. This limitation is addressed by probability theory, which allows for the representation of distributions over the models of a knowledge base, thereby enhancing its expressivity. Further expressivity limitations are overcome by more expressive logical frameworks such as first-order logic to be treated in Chapter 11. To leverage the strengths of both approaches, we adopt the tensor formalism as a unified framework towards hybrid representation and reasoning schemes investigated in Part II.

1.2 Reasoning by Contractions

In Part I, we frame tensor network contractions as fundamental operation for reasoning in artificial intelligence. In the probabilistic and logical approaches tensor network contractions appear as basic retrieval operations of knowledge about a system:

- **Probability theory:** The contraction of probability distributions is the central operation to compute marginal distributions. Here the contraction provides a sum over atomic events to calculate the probability of a combination of random variables being in a specific state. Conditioned probability distributions can be instantiated from normalized contractions of probability distributions with event indicator tensors. When contracting such conditioned probability distributions, the contraction calculates the conditional probability.
- **Propositional logic:** The contraction of knowledge bases is the central operation to count the number of possible states, or models, of the knowledge base. In such way, the satisfiability of a knowledge base can be decided on the positivity of the contraction. Conjunctions of propositional formulas are on the other side represented by contractions leaving all atomic variables open. We can thus decide entailment by contracting such conjunctions with no open variables.

Based on these motivations, we formulate in this work all algorithms as orchestrations of contractions.

1.3 Dictionary

The logical and probabilistic approaches towards artificial intelligence, and the tensor formalism, refer to similar concepts with different expressions. A rough dictionary is provided by:

Concept	**Tensors**	**Probability theory**	**Propositional logic**
Atomic Representation	Vector	Random Variable	Atomic Variable
Factored Representation	Tensor	Joint Distribution	Propositional Formula
State	Index	Atomic Event	World
Knowledge about a state	Coordinate	Probability (in $[0, 1]$)	Possibilities (in $\{0, 1\}$)
Retrieval of knowledge	Contractions	Marginalization, Conditioning	Model counts

1.4 Outline

We structure Part I in four chapters dedicated to the foundations of probabilistic and logical methods of artificial intelligence. In Chapter 2 we investigate tensor network representations of probability distributions and review in Chapter 3 reasoning schemes based on contractions of these representations. We then turn to syntactical representations of propositional knowledge bases in Chapter 4 and show corresponding tensor network representations. Based on probabilistic interpretations of these knowledge bases we then investigate logical reasoning schemes in Chapter 5.

2

Probability Distributions

In this chapter, we will establish relations between the formalism of tensor networks and basic concepts of probability theory. We will first introduce joint distributions by tensors and connect their marginalizations and conditionings to basic tensor operations. Sparse representations of these tensors by tensor networks will be motivated based on independence properties and sufficient statistics. These mechanisms lead towards exponential families and graphical models, for which we provide tensor network representations. We then investigate the representation of probability distributions by mean parameters, which build convex polytopes and relate properties of distributions with the position of their mean parameters in the corresponding polytope.

2.1 Tensor Formalism in Probability Theory

Let us relate the main terminology of probability theory with the tensor network formalism. As such, any joint distribution of finitely valued variables is a tensor. Further, the operations of marginalization and conditioning of distributions are equal to contractions and normalizations of tensors.

2.1.1 Joint Probability Distributions

Let there be uncertainties of the assignments x_k to the categorical variables X_k of a system. In this scenario we understand each X_k as random variables, which together have a joint distribution defined by the uncertainties of the state assignments. The uncertainty of a state $x_{[d]}$ is then quantified by probabilities

$$\mathbb{P}\left[X_{[d]} = x_{[d]}\right] \in [0,1]\,.$$

Joint probability distributions are therefore naturally represented by tensors, whose coordinates store the probabilities to each state. These tensors have additional properties as we state in the next definition.

Definition 2.1 (Joint Probability Distribution) 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 function

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

which is non-negative, that is for any $x_{[d]} \in \bigtimes_{k\in[d]} [m_k]$

$$\mathbb{P}\left[X_{[d]} = x_{[d]}\right] \geq 0\,,$$

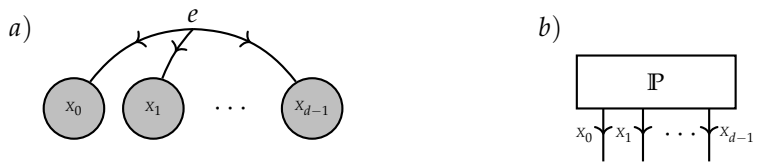


Figure 2.1: Probability distributions of variables $X_0, \dots, X_{d-1}$, sketched a) by a directed edge e with all variables outgoing, which is decorated b) by a directed tensor $\mathbb{P}\left[X_{[d]}\right]$.

and which is normalized, that is

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

In a slight abuse of notation, we directly applied the coordinate encoding scheme (see Def. 0.19 and for more details Chapter 13) to represent a probability distribution as a tensor. Along this scheme, any probability distribution is a weighted sum of one-hot encodings (see Lem. 13.1) by

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

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 normalization 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. 2.1 reflects the first probability axiom of Kolmogorov, the assumption of normalization, that is a contraction 1, implements the second axiom (see for example [DeG16]). Since probability distributions contract to 1, they are directed (see Def. 0.17) with all distributed variables outgoing and no incoming variables (see Figure 2.1).

2.1.2 Base Measures

From a measure theoretic perspective, probabilities are measurable functions called probability densities, which integrals are 1 (see for example [DeG16]). 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 referred to as state counting base measure. The distributions $\mathbb{P}\left[X_{[d]}\right]$ 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 normalization 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 probabilistic 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 2.2 We say that a probability distribution $\mathbb{P}$ is representable with respect to a boolean base measure ν, if for all $x_{[d]}$ with $\nu\left[X_{[d]} = x_{[d]}\right] = 0$ we have $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0$. We denote the set of by ν representable distributions by $\Lambda^{\delta,\text{MAX},\nu}$.

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

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

Starting with Chapter 4 we will further investigate boolean tensors and relate them with propositional formulas. In Chapter 5 we will connect the representation and positivity with respect to boolean base measures with the formalism of entailment. The notation $\Lambda^{\delta,\text{MAX},\nu}$ of by ν representable distributions will later in this chapter relate with distributions computable given a statistic.

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 5 orders based on entailment and show the minimality in Thm. 5.10. Here, we just introduce the minimality criterion as positivity of a distribution with respect to a base measure.

Definition 2.3 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 $\nu[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] = 1$ we have $\mathbb{P}[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] > 0$. We further say that a distribution is positive, if it is positive with respect to the trivial base measure $\mathbb{I}\left[X_{[d]}\right]$.

2.1.3 Marginal Distribution

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

Definition 2.4 (Marginal Probability) Given a distribution $\mathbb{P}[X_0, X_1]$ of the categorical vari-

ables X_0 and X_1 the marginal distribution of the categorical variable X_0 is the tensor

$$\mathbb{P}[X_0] \in \mathbb{R}^{m_0}$$

defined by the contraction

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

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

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

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 $\mathbb{P}[X_0]$ of X_0. Let us now justify this terminology and show, that any marginal distribution is a probability distribution as introduced in Def. 2.1.

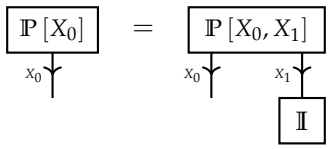
Theorem 2.5 *Any marginal distribution is a probability distribution.*

Proof. We further have that any marginal distribution is normalized, since by the commutativity of contractions (see for more details Thm. 17.1 in Part III)

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

Further any coordinate is non-negative, since it is a sum of non-negative coordinates. It follows from Def. 2.1, 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 $\mathbb{I}[X_1]$. Following this notation, we depict the marginal distribution in Def. 2.4 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. 2.4 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.

2.1.4 Conditional Probabilities

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

Definition 2.6 (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 normalizable on $\{X_1\}$. Then the distribution of

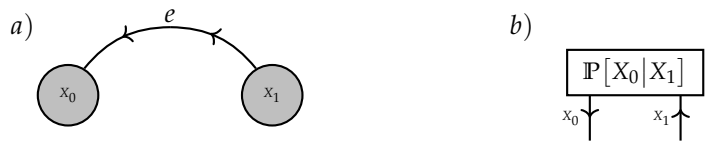
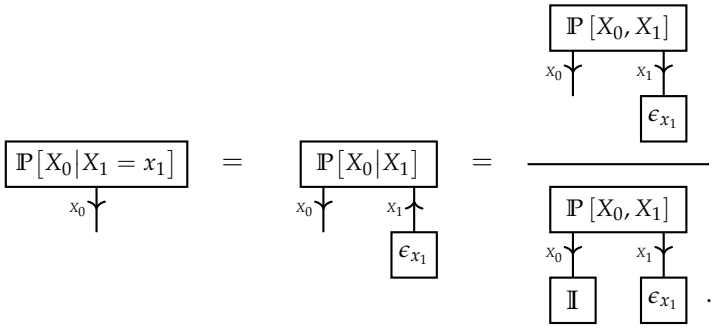


Figure 2.2: Depiction of conditional probability distributions a) by an edge with the incoming variable X_1 and the outgoing variable X_1, which is decorated by b) the directed tensor $\mathbb{P}[X_0|X_1]$.

X_0 conditioned on X_1 is defined by

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

Since conditional probabilities are normalizations of probability tensors, they are directed and therefore depicted by directed hyperedges (see Figure 2.2). For any $x_1 \in [m_1]$ we depict the slice $\mathbb{P}[X_0|X_1 = x_1]$ defined by a normalization operation as



As we have done before for marginal distribution, we relate Def. 2.6 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

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

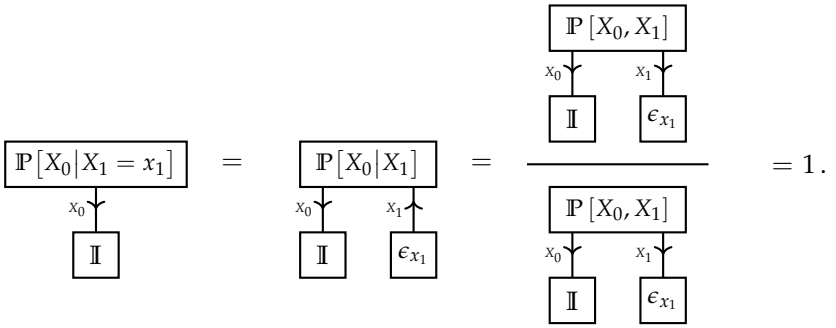
The distribution of X_0 conditioned on X_1 is the normalized collection of slice of the probability distribution $\mathbb{P}[X_0, X_1]$. Each slice of the conditioned distribution with respect to incoming variables is a probability distribution itself, as we show next.

Theorem 2.7 *For any $x_1 \in [m_1]$ the tensor $\mathbb{P}[X_0|X_1 = x_1]$ is a probability tensor.*

Proof. As a normalization 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 normalized. We can visualize this calculation exploiting our diagramatic notation as



Since for any $x_1 \in [m_1]$ the slice $\mathbb{P}\big[X_0 \big| X_1 = x_1\big]$ 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 2.8 *Any tensor with non-negative coordinates is a conditional distribution, if and only if it is directed with the conditioned variables incoming and the other outgoing.*

Proof. "⇒": By Thm. 2.7 a conditional probability tensor $\mathbb{P}\big[X_0 \big| X_1\big]$ is the normalization of a tensor and by Thm. 13.12 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 normalizations.

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

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

is a probability tensor, since

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

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

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

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

Thm. 2.8 specifies a broad class of tensors to represent conditional probabilities. In combination with Thm. 14.10, which states that basis encodings are directed, we get that any basis encoding of a function is a conditional probability tensor.

2.1.5 Bayes Theorem and the Chain Rule

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

Theorem 2.9 (Bayes Theorem) *For any probability distribution $\mathbb{P}[X_0, X_1]$ with positive $\mathbb{P}[X_1]$ we have*

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

Proof. This theorem follows from the more generic contraction equation Thm. 13.13 to be shown in Chapter 13. We note that by positivity of $\mathbb{P}[X_1]$, the tensor network $\mathbb{P}$ is normalizable with respect to X_1. Thm. 13.13 therefore implies choosing $\mathcal{V} = \{0, 1\}$, $\mathcal{V}^{\text{in}} = \{1\}$ and $\mathcal{V}^{\text{out}} = \{0\}$, that For our tensor

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

■

Following the insight of the Bayes Thm. 2.9, probability distributions of arbitrary numbers of variables can be decomposed as a contraction of conditional probabilities, as we show in the next theorem.

Theorem 2.10 (Chain Rule) *For any probability distribution $\mathbb{P}\left[X_{[d]}\right]$ we have*

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

provided that all conditional probability distributions exist.

Proof. The claim can be derived by an iterative application of the Bayes Thm. 2.9 theorem. We will proof this statement in more generality in Chapter 13 as Thm. 13.14, deducing it from the generalization of the Bayes Thm. 2.9 by Thm. 13.13. The claim here then follows from Thm. 13.14 using $\mathcal{V} = [d]$ and $\tau[X_{\mathcal{V}}] = \mathbb{P}\left[X_{[d]}\right]$, since

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

■

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 independencies in the distribution, which we will introduce in the next section.

2.2 Decomposition of Probability Distributions

The number of coordinates in a tensor representation of probability distributions is the product

$$\prod_{k \in [d]} m_{X_k} ,$$

and therefore scales exponentially in the number of coordinates. To find efficient representation schemes of probability distributions by tensor networks, we need to exploit additional properties of the distribution. In this section, we introduce the independence and the computation mechanisms towards tensor network decompositions of probability distributions.

2.2.1 Independence mechanism

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 2.11 (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 $x_1 \in [m_1]$ the distribution satisfies

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

In this case we denote $(X_0 \perp X_1)$.

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

Theorem 2.12 (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 Thm. 2.5 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

$$\forall x_0 \in [m_0], x_1 \in [m_1] : \mathbb{P}[X_0 = x_0, X_1 = x_1] = \mathbb{P}[X_0 = x_0] \cdot \mathbb{P}[X_1 = x_1]$$

is equivalent to

$$\begin{aligned} &\sum_{x_0 \in [m_0]} \sum_{x_1 \in [m_1]} \mathbb{P}[X_0 = x_0, X_1 = x_1] \cdot \epsilon_{x_0}[X_0] \, \epsilon_{x_1}[X_1] \\ &\quad = \sum_{x_0 \in [m_0]} \sum_{x_1 \in [m_1]} \mathbb{P}[X_0 = x_0] \cdot \mathbb{P}[X_1 = x_1] \cdot \epsilon_{x_0}[X_0] \, \epsilon_{x_1}[X_1] . \end{aligned}$$

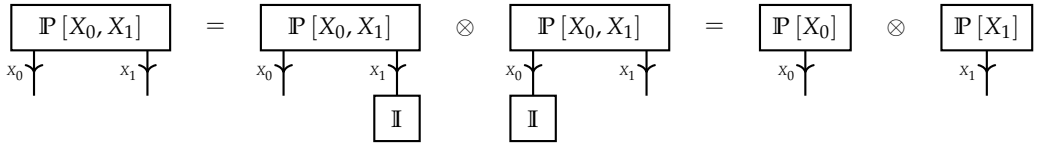
We reorder the summations and arrive at

$$\begin{aligned} &\sum_{x_0 \in [m_0]} \sum_{x_1 \in [m_1]} \mathbb{P}[X_0 = x_0, X_1 = x_1] \cdot \epsilon_{x_0, x_1}[X_0, X_1] \\ &\quad = \left(\sum_{x_0 \in [m_0]} \mathbb{P}[X_0 = x_0] \, \epsilon_{x_0}[X_0] \right) \cdot \left(\sum_{x_1 \in [m_1]} \mathbb{P}[X_1 = x_1] \cdot \epsilon_{x_1}[X_1] \right) \end{aligned}$$

which is by Lem. 13.1 equal to the claim

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

Two jointly distributed variables are by Thm. 2.12 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 into 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 2.13 (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 $x_2 \in [m_2]$

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

In this case we denote $(X_0 \perp X_1) | X_2$.

Conditional independence stated in Def. 2.13 has a close connection with independence stated in Def. 2.11. 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 $\mathbb{P}\left[X_0, X_1 \middle| X_2 = x_2\right]$ of the conditional distribution $\mathbb{P}\left[X_0, X_1 \middle| X_2\right]$. Analogously to Thm. 2.12 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 2.14 (Conditional Independence as a Contraction Equation) *Given a distribution $\mathbb{P}$ of variables X_0, X_1 and X_2, the variable X_0 is independent of X_1 conditioned on X_2, if and only if the equation*

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

holds.

Proof. With the same argumentation as in the proof of Thm. 2.12, we notice that the contraction equation holds, if and only if for any $x_0 \in [m_0]$, $x_1 \in [m_1]$ and $x_2 \in [m_2]$

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

This is equivalent to conditional independence by Def. 2.13. ∎

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

Corollary 2.15 *If and only if X_0 is independent of X_1 conditioned on X_2 the probability distribution $\mathbb{P}$ satisfies (see Figure 2.3)*

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

Proof. With the Bayes Thm. 2.9 it holds that

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

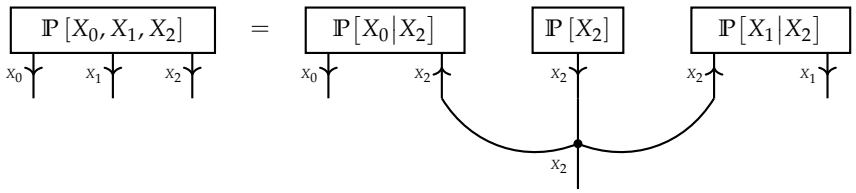


Figure 2.3: Diagrammatic visualization of the contraction equation in Cor. 2.15. Conditional independence of X_0 and X_1 given X_2 holds if the contraction on the right side is equal to the probability tensor on the left side.

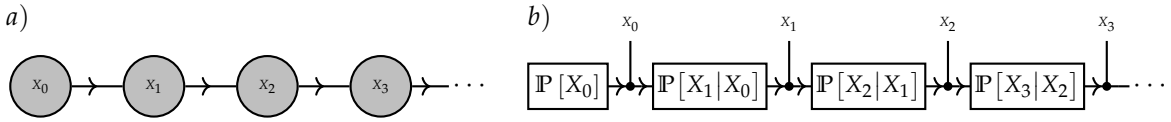


Figure 2.4: Depiction of a Markov Chain Decomposition by a a) hypergraph with the nodes $\mathcal{V} = [d]$ and edges $\mathcal{E} = \{\{0\} \cup \{\{k, k+1\} : k \in [d], k > 1\}\}$ and b) a decorating Tensor Network representing the sparsified conditional probabilities.

Decomposing the first tensor in the contraction, Thm. 2.14 implies, that X_0 is independent of X_1 conditioned on X_2, if and only if

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

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 Thm. 2.10. As we state as the next theorem, such sparsifications follow from conditional independence.

Theorem 2.16 *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 Thm. 2.9 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

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

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

Theorem 2.17 (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

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

We depict this decomposition in Figure 2.4.

Proof. By the chain rule shown in Thm. 2.10 we have

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

Using that X_k is conditionally independent of $X_{[k-1]}$ conditioned on X_{k-1} we further have by Thm. 2.16

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

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. Thm. 2.17 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 $k \geq 1$

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

The conditional probability matrices are therefore called stochastic transition matrices.

We notice that the decomposition scheme of Thm. 2.17 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.

2.2.2 Computation mechanism

We have seen, that conditional independence of variables corresponds with decomposition properties of probability tensors. We now explore a further approach towards tensor network decomposition based on sufficient statistics, which leads to exponential families. When restricting exponential families to graphical models in Sect. 2.5, we will see that both the indepence and the computation mechanisms are related through the Hammersley-Clifford theorem.

Let us consider a tuple of random variables $X_{[d]}$, which take values in $\bigtimes_{k\in[d]}[m_k]$. We now understand the probability $\mathbb{P}\left[X_{[d]}\right]$ as another random variable taking values in $[0,1]$, which has a deterministic dependence on $X_{[d]}$.

Definition 2.18 (Sufficient Statistics) Let $X_{[d]}$ be a list of by $\mathbb{P}\left[X_{[d]}\right]$ jointly distributed random variables and $\mathcal{S}\left(X_{[d]}, L\right)$ be a tensor. Consider the tuple of random variables $(X_{[d]}, \mathbb{P}\left[X_{[d]}\right], \mathcal{S}\left(X_{[d]}, L\right))$, which takes for $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ with probability $\mathbb{P}\left[X_{[d]} = x_{[d]}\right]$ the value

$$\left(x_{[d]}, \mathbb{P}\left[X_{[d]} = x_{[d]}\right], \mathcal{S}\left(X_{[d]} = x_{[d]}, L\right)\right) .$$

We say that $\mathcal{S}$ is a sufficient statistic for $\mathbb{P}$, if this tuple obeys

$$\left(X_{[d]} \perp \mathbb{P}\left[X_{[d]}\right]\right) \Big| \mathcal{S}\left(X_{[d]}, L\right) .$$

When a probability distribution has a sufficient statistic, we can represent the distribution as a tensor network involving the basis encoding of the statistic, as we show next.

Theorem 2.19 *If and only if $\mathcal{S}\left(X_{[d]}, L\right)$ is a sufficient statistic, i.e. $X_{[d]}$ is independent of $\mathbb{P}\left[X_{[d]}\right]$ conditioned on $\mathcal{S}\left(X_{[d]}, L\right)$, there is a tensor $\xi\left[Y_{[p]}\right]$ with*

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}\right] .$$

Proof. We exploit conditional entropies, for which definition we refer to Chapter 2 in [CT06]. By the data processing inequality (see e.g. Theorem 2.8.1 in [CT06]), we have

$$\mathbb{H}\left[\mathbb{P}|\mathcal{S}\right] \leq \mathbb{H}\left[\mathbb{P}|X_{[d]}\right] = 0$$

and thus $\mathbb{H}\left[\mathbb{P}|\mathcal{S}\right] = 0$. Moreover, $\mathbb{H}\left[\mathbb{P}|\mathcal{S}\right] = 0$ is equivalent to a straight satisfaction of the data processing inequality, and $\mathcal{S}$ being a sufficient statistic for $\mathbb{P}$. $\mathbb{H}\left[\mathbb{P}|\mathcal{S}\right] = 0$ is further equivalent to the existence of a function $q : \mathbb{R}^p \to [0,1]$, such that for each $x_{[d]}$

$$q_{\mathcal{S}\left(X_{[d]}=x_{[d]}, L\right)} = \mathbb{P}\left[X_{[d]} = x_{[d]}\right] .$$

For an index interpretation function I, enumerating $\bigtimes_{l\in[p]} \mathrm{im}\,(s_l)$ using the variables $Y_{[p]}$, we define

$$\tau := q \circ I .$$

Using basis calculus (see Chapter 14) and in particular Thm. 14.13 we have

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}\right] . \qquad \blacksquare$$

Given a statistic $\mathcal{S}$ we can thus characterize the set of probability distributions, for which $\mathcal{S}$ is sufficient, as

$$\Lambda^{\mathcal{S},\mathrm{MAX}} := \left\{ \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}|\varnothing\right] \,:\, 0\left[Y_{[p]}\right] \prec \xi\left[Y_{[p]}\right] \right\} ,$$

where by $0 \left[Y_{[p]}\right] \prec \xi \left[Y_{[p]}\right]$ we restrict to the activation tensors with non-negative coordinates.

We further extend the definition of sufficient statistics to generic boolean base measures ν, where we say that a probability distribution has a sufficient statistic $\mathcal{S}$ with respect to a base measure ν, if

$$\left(X_{[d]} \perp \mathbb{P}\left[X_{[d]}\right]\right) \Big| \mathcal{S}\left(X_{[d]}, L\right), \nu\left[X_{[d]}\right].$$

Analogously, the set of distributions with sufficient statistic $\mathcal{S}$ with respect to a base measure ν is the set

$$\Lambda^{\mathcal{S},\mathrm{MAX},\nu} := \left\{ \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right], \nu\left[X_{[d]}\right] \right\rangle \left[X_{[d]} | \varnothing\right] \,:\, 0\left[Y_{[p]}\right] \prec \xi\left[Y_{[p]}\right] \right\}.$$

2.2.3 Computation-Activation Networks

The curse of dimensionality appears again in cases of large cardinality of the image of the statistic $\mathcal{S}$. To describe classes of probability distributions with efficient representations in these cases, we now define classes of distributions with tensor network decompositions of their activation cores.

Definition 2.20 (Computation-Activation Network) Given a statistic $\mathcal{S} : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}^p$, and a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $[p] \subset \mathcal{V}$ containing the image coordinates of $\mathcal{S}$, we define the by $\mathcal{S}$ computable and by $\mathcal{G}$ activated family of distributions by

$$\Lambda^{\mathcal{S},\mathcal{G}} = \left\{ \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \langle\xi\rangle\left[Y_{[p]}\right] \right\rangle \left[X_{[d]} | \varnothing\right] \,:\, \xi\left[Y_{\mathcal{V}}\right] \in \mathcal{T}^{\mathcal{G}} \right\}.$$

We refer to any member $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{S},\mathcal{G}}$ as a Computation-Activation Network.

The set $\Lambda^{\mathcal{S},\mathrm{MAX}}$ of probability distributions with a sufficient statistic $\mathcal{S}$ is the special case of the maximal graph

$$\mathrm{MAX} = ([p], \{[p]\})$$

which has a single hyperedge containing all nodes $\mathcal{V} = [p]$. For any hypergraph $\mathcal{G}$ we further have

$$\Lambda^{\mathcal{S},\mathcal{G}} \subseteq \Lambda^{\mathcal{S},\mathrm{MAX}}.$$

To represent a distribution $\mathbb{P}\left[X_{[d]}\right]$ in $\Lambda^{\mathcal{S},\mathcal{G}}$ we need two tensor networks:

- **Computation network:** This a tensor network, which contraction is the basis encoding of the statistic

  $$\beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right] = \left\langle \tau^{\mathcal{G}} \right\rangle \left[Y_{[p]}, X_{[d]}\right].$$

  Further decompositions of the basis encoding of the statistic is often necessary to avoid the curse of dimensions and possible in case of sparse statistics. We refer to these as computation cores, since they are computing the variables $Y_{[p]}$ using the basis calculus scheme (see Chapter 14).

- **Activation network:** The activation tensor $\langle\xi\rangle\left[Y_{[p]}\right]$ activates the computed statistics, in the sense that the contraction with the computation network provides a non-trivial factor, if and only if the activation tensor itself is not trivial. When demanding $\xi\left[Y_{\mathcal{V}}\right] \in \mathcal{T}^{\mathcal{G}}$, the activation tensor decomposes into a tensor network of tensors $\xi^e\left[Y_e\right]$ for each hyperedge $e \in \mathcal{E}$.

2.3 Exponential Families

Thm. 2.19 states the existence of an activation core, once a sufficient statistic has been identified. However, since the dimension of the activation tensor space is increasing exponentially with the number of features, the representation of generic activation tensors is not feasible when having many features. We now restrict the activation cores to specific elementary tensors, which correspond with further assumptions on the dependence of $\mathbb{P}$ and $\mathcal{S}$ made by exponential families.

2.3.1 Definition

Definition 2.21 (Exponential Family) Given a statistic function

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

and a boolean base measure

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

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

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

is called the exponential family to $\mathcal{S}$. We further define for each member with parameters θ the associated energy tensor

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

and the cumulant function

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

We used the selection encoding to represent the weighted summation over the statistics, that is the tensor (see Def. 0.22)

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

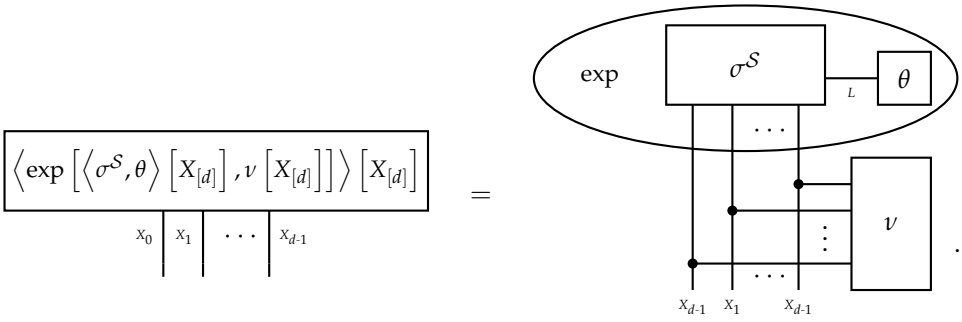
defined for $x_{[d]} \in \bigtimes_{k\in[d]} [m_k]$ and $l \in [p]$ as

$$\sigma^{\mathcal{S}} \left[X_{[d]} = x_{[d]}, L = l\right] = \mathcal{S}_l \left[X_{[d]} = x_{[d]}\right] .$$

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

$$\sum_{l\in[p]} \theta [L = l] \cdot \mathcal{S}_l \left[X_{[d]}\right] = \left\langle \sigma^{\mathcal{S}} \left[X_{[d]}, L\right], \theta [L]\right\rangle \left[X_{[d]}\right] .$$

For more details on this representation scheme, we refer to Thm. 14.24 in Chapter 13. Up to normalization, 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 $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$ 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 $\left\langle \nu, \exp\left[\left\langle \sigma^{\mathcal{S}}, \theta \right\rangle \left[X_{[d]}\right]\right]\right\rangle [\varnothing]$ diverges and the normalization $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$ is not well-defined. In that cases, the canonical parameters need to be chosen from a subset where the partition function is finite [WJ08].

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 $\Gamma^{\mathcal{S},\nu}$ is positive with respect to ν (see Def. 2.3). In Chapter 8 we will investigate distributions, where the base measures and the sufficient statistics share a common decomposition framework.

Lemma 2.22 *For any member of an exponential family $\Gamma^{\mathcal{S},\nu}$ we have*

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

Proof. By definition we have

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

■

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

Definition 2.23 (Minimal Statistics) We say that a statistic $\mathcal{S}$ 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

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

If a statistic is not minimal, we can omit coordinates of it without affecting the expressivity $\Gamma^{\mathcal{S},\nu}$. As long as we find a non-vanishing vector $V[L]$ and $\lambda \in \mathbb{R}$ as in Def. 2.23, we can choose

a coordinate s_l such that $V[L = l] \neq 0$, conclude that the coordinate is linear dependent on the others and drop it as redundant.

2.3.2 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 basis encoding formalism to represent members of exponential families by a single contraction without nonlinear transforms.

Theorem 2.24 (Tensor Network Representation of Exponential Families) *Given any base measure ν and a sufficient statistic $\mathcal{S}$ we enumerate for each coordinate $l \in [p]$ the image* $\mathrm{im}(s_l)$ *by a variable Y_l taking values in* $[|\mathrm{im}(s_l)|]$ *(see for more details on this scheme Chapter 14), given an interpretation map*

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

For any canonical parameter vector $\theta[L] \in \mathbb{R}^p$ we build the activation cores $\alpha^{l,\theta}[Y_l]$ for each coordinate $y_l \in [|\mathrm{im}(s_l)|]$ by

$$\alpha^{l,\theta}[Y_l = y_l] = \exp[\theta[L = l] \cdot I_l(y_l)]$$

and have

$$\mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right] = \left\langle \{\nu\left[X_{[d]}\right]\} \cup \{\beta^{s_l}\left[Y_l, X_{[d]}\right] : l \in [p]\} \cup \{\alpha^{l,\theta}[Y_l] : l \in [p]\}\right\rangle \left[X_{[d]}|\varnothing\right] .$$

Proof. We embed the image of $\mathcal{S}$ in the cartesian product of the coordinate images

$$\mathrm{im}(\mathcal{S}) \subset \bigtimes_{l\in[p]} \mathrm{im}(s_l)$$

and design enumerate the embedded image of $\mathcal{S}$ by the variables $Y_{[p]}$. Thm. 14.13, to be shown in Chapter 14, implies

$$\exp\left[\left\langle \sigma^{\mathcal{S}}, \theta\right\rangle \left[X_{[d]}\right]\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \exp[\langle\cdot, \theta[L]\rangle]|_{\times_{l\in[p]}\mathrm{im}(s_l)}\right\rangle \left[X_{[d]}\right] .$$

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

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

We further have (see Thm. 14.17 in Chapter 14)

$$\beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right] = \left\langle \{\beta^{s_l}\left[Y_l, X_{[d]}\right] : l \in [p]\}\right\rangle \left[Y_{[p]}, X_{[d]}\right] .$$

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

In the proof of Thm. 2.24 we have observed, that the basis encoding $\beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right]$ of the statistics decomposed into a tensor network of basis encodings $\beta^{s_l}\left[Y_l, X_{[d]}\right]$ 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 coordindates 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 8, 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 Thm. 2.24 is split into computation and activation cores (see Sect. 2.2.3):

- **Computation Cores:** The basis encodings β^{s_l} of the coordinates of a statistic form the 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.

- **Activation Cores:** The influence of the contraction is performed by the activation cores $\alpha^{l,\theta}[Y_l]$, which exploit the computed statistic variables $Y_{[p]}$ and provide in combination with the basis encoding a factor

  $$\left\langle \beta^{s_l}\left[Y_l, X_{[d]}\right], \alpha^{l,\theta}[Y_l]\right\rangle \left[X_{[d]}\right]$$

  to the tensor network representing the distribution of the observed variables $X_{[d]}$.

For the implementation of an exponential family as a Computation-Activation Network, see Example A.1.

When the canonical parameter is vanishing at a coordinate, that is $\theta[L = l] = 0$, then the corresponding factor is trivial, since $\alpha^{l,0}[Y_l] = \mathbb{I}[Y_l]$ and as a consequence of the directionality of basis encodings we have

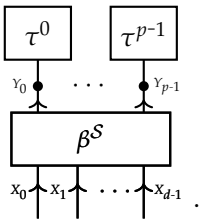
$$\left\langle \beta^{s_l}\left[Y_l, X_{[d]}\right], \alpha^{l,\theta}[Y_l]\right\rangle \left[X_{[d]}\right] = \left\langle \beta^{s_l}\left[Y_l, X_{[d]}\right], \mathbb{I}[Y_l]\right\rangle \left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right].$$

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

By Thm. 2.24 any member of an exponential family is represented by the normalized contraction of a collection of unary activation cores contracted with the computation cores $\beta^{s_l}\left[Y_l, X_{[d]}\right]$. The unary activation cores form a tensor network on the elementary hypergraph

$$\mathrm{EL} = ([p], \{\{l\} : l \in [p]\}),$$

which hyperedges contain single variables. We call this graph elementary, since it corresponds with elementary tensor network formats consistent of tensor products of vectors. For unary activation cores, that is for the elementary graph EL, any member of $\Lambda^{\mathcal{S},\mathrm{EL}}$ has up to a normalization factor a tensor network decomposition by the diagram

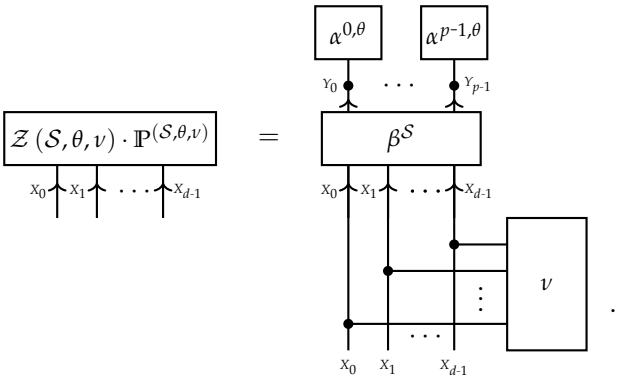


Comparing this representation scheme with Thm. 2.24, we conclude as the next corollary, that any member of an exponential family with trivial base measure can be represented by an elementary activation tensors.

Corollary 2.25 (of Thm. 2.24) *For any statistic* $\mathcal{S} : \bigtimes_{k\in[d]}[m_k] \to [p]$ *and trivial base measure* $\nu\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$ *we have*

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

For elements of the exponential family with general boolean base measure we have with the activation cores constructed in Thm. 2.24



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. 2.25 to families of nontrivial base measures. We will investigate such schemes later in Chapter 8, 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 non-linear 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 7) in neuro-symbolic reasoning (see Chapter 9).

Based on Cor. 2.25 a further natural question is, whether $\Gamma^{\mathcal{S},\mathbb{I}}$ is a proper subset of $\Lambda^{\mathcal{S},\mathrm{EL}}$. This is the case for most statistics $\mathcal{S}$, since members of exponential families are positive with respect to their base measure, which is in the corollaries setting trivial, while in $\Lambda^{\mathcal{S},\mathrm{EL}}$ we allow also for activation cores with vanishing coordinates, which in general do not produce positive distributions. The only statistics where $\Gamma^{\mathcal{S},\mathbb{I}}$ is not a proper subset of $\Lambda^{\mathcal{S},\mathcal{G}}$ 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 5, and will use the formats $\Lambda^{\mathcal{S},\mathcal{G}}$ 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.

2.4 Polytopes of Mean Parameters

We in this section investigate properties of probability distributions based on their mean parameters. Given a statistic $\mathcal{S}$, we first define a mean parameter to any distribution by the expectation of the statistic.

Definition 2.26 Let there be a statistic $\mathcal{S}$ and a boolean base measure $\nu\left[X_{[d]}\right]$. We call the tensor

$$\mu\left[L\right] = \left\langle \mathbb{P}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle\left[L\right]$$

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

$$\mathcal{M}_{\mathcal{S},\nu} = \left\{\left\langle \mathbb{P}, \sigma^{\mathcal{S}}, \nu\right\rangle\left[L\right] : \mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\delta,\mathrm{MAX},\nu}\right\}$$

is called the polytope of realizable mean parameters. Here we denote by $\Lambda^{\delta,\mathrm{MAX},\nu}$ the set of all probability distributions representable with respect to ν (see Def. 2.2).

While introduced here as a property of a distribution, the mean parameters will be central to probabilistic inference in Chapter 3. We in the reminder of this section prepare for this application and derive tensor network representations for distributions having sufficient statistics $\mathcal{S}$, depending on their corresponding mean parameter in the polytope.

2.4.1 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 (see Figure 2.5). Convex hulls of finite vectors are called $\mathcal{V}$-polytopes (see Lecture 1 in [Zie13]).

Theorem 2.27 *For any statistic $\mathcal{S}$ the polytope of mean parameters is the convex hull of the slices of $\sigma^{\mathcal{S}}$ with fixed indices to $X_{[d]}$, that is*

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

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

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

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 normalized.

The polytope of mean parameters is a linear transform of the elements in $\Lambda^{\delta,\mathrm{MAX},\nu}$, since the contraction with $\sigma^{\mathcal{S}}$ is linear. It follows that

$$\mathcal{M}_{\mathcal{S},\nu} = \mathrm{conv}\left(\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \epsilon_{x_{[d]}}\left[X_{[d]}\right]\right\rangle\left[L\right] : \nu\left[X_{[d]} = x_{[d]}\right] = 1\right)$$

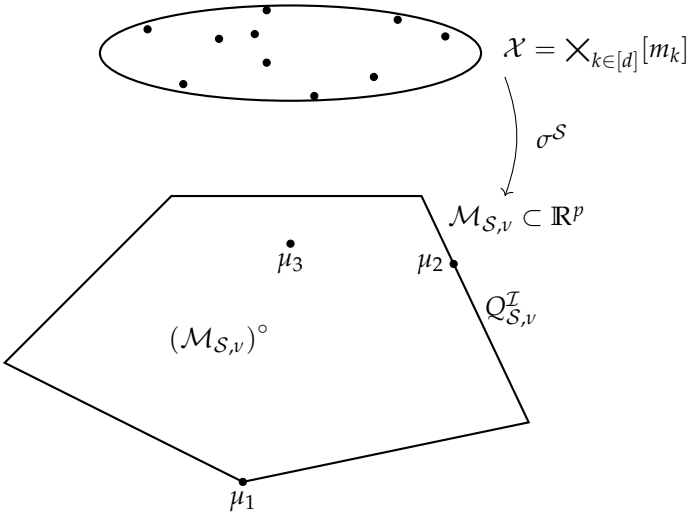


Figure 2.5: Sketch of the mean polytope $\mathcal{M}_{\mathcal{S},\nu}$ to a statistic $\mathcal{S}$, which is minimal with respect to ν. The mean polytope is a bounded subset of $\mathbb{R}^p$ (here sketched as a 2-dimensional projection). For any vertex μ_1 we find $x_{[d]}$ such that $\mu_1[L] = \sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right]$. Generic mean parameters μ_2 outside the interior are on a face $Q^{\mathcal{I}}_{\mathcal{S},\nu}$. Interior points $\mu_3 \in \mathcal{M}_{\mathcal{F},\mathbb{I}}{}^\circ$ are exactly those reproducible by positive distributions with respect to ν.

$$= \operatorname{conv}\left(\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right). \qquad \blacksquare$$

We thus understand the map

$$\sigma^{\mathcal{S}} \,:\, \bigtimes_{k\in[d]}[m_k] \to \mathcal{M}_{\mathcal{S},\nu} \quad , \quad x \to \sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right]$$

as an encoding of states with respect to a statistic $\mathcal{S}$. When the statistic is the universal statistic $\mathcal{S} = \delta$, this statistic encoding coincides with the one-hot encoding and the mean polytope is the simplex of dimension $\langle\nu\rangle[\varnothing] - 1$

$$\mathcal{M}_{\delta,\nu} = \operatorname{conv}\left(\epsilon_{x_{[d]}}\left[X_{[d]}\right] \,:\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

A generic mean polytope $\mathcal{M}_{\mathcal{S},\nu}$ is then the linear transform of the simplex by the contraction with $\sigma^{\mathcal{S}}\left[X_{[d]}, L\right]$, where L is left open.

2.4.2 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 [Zie13]. We state next, that the polytope $\mathcal{M}_{\mathcal{S},\nu}$ of mean parameters is a $\mathcal{H}$-polytopes.

Theorem 2.28 *The set $\mathcal{M}_{\mathcal{S},\nu}$ is for any statistic $\mathcal{S}$ and base measure ν a $\mathcal{H}$-polytope, i.e. 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

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

Proof. By Thm. 2.27, the set $\mathcal{M}_{\mathcal{S},\nu}$ 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 [Mot36], which states the equivalence of $\mathcal{V}$-polytopes and $\mathcal{H}$-polytope, for which a proof can be found as Theorem 1.1 in [Zie13]. Therefore, $\mathcal{M}_{\mathcal{S},\nu}$ 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. [WJ08]).

2.4.3 Characterization of the Interior

The interior of the mean polytope consists of the mean parameters to positive distributions as we show next.

Theorem 2.29 *For any minimal statistics $\mathcal{S}$ with respect to a boolean base measure ν (see Def. 2.23) and a with respect to ν positive distribution $\mathbb{P}\left[X_{[d]}\right]$ we have*

$$\left\langle \mathbb{P}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right] \right\rangle [L] \in (\mathcal{M}_{\mathcal{S},\nu})^{\circ}.$$

Proof. Since by assumption the statistics is minimal, the convex set $\mathcal{M}_{\mathcal{S},\nu}$ is full dimensional (see e.g. Appendix B in [WJ08]). We thus use a well-known property for full-dimensional convex sets (see [Roc97; HL93]), that $\mu \in {\mathcal{M}_{\mathcal{S},\nu}}^{\circ}$ 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

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

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

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

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

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

and thus

$$\frac{\partial}{\partial \epsilon}\left\langle \mathbb{P}^{\epsilon}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], V[L] \right\rangle [\varnothing]|_{\epsilon=0} = \left\langle \mathbb{P}\left[X_{[d]}\right], (\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], V[L] \right\rangle [])^2 \right\rangle \left[X_{[d]}\right] - \left(\left\langle \mathbb{P}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], V[L] \right\rangle \left[X_{[d]}\right]\right)^2.$$

We can interpret this quantity as the variance of the random variable

$$\left\langle \sigma^{\mathcal{S}} \left[X_{[d]}, L \right], V[L] \right\rangle \left[X_{[d]} = x_{[d]} \right] ,$$

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 $\mathcal{S}$ with respect to ν it follows, that this variable is not constant and we therefore have

$$0 < \frac{\partial}{\partial \epsilon} \left\langle \mathbb{P}^{\epsilon} \left[X_{[d]} \right], \sigma^{\mathcal{S}} \left[X_{[d]}, L \right], V[L] \right\rangle [\varnothing] \mid_{\epsilon=0} .$$

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

$$\langle V[L], \mu[L] \rangle [\varnothing] < \left\langle V[L], \sigma^{\mathcal{S}} \left[X_{[d]}, L \right], \mathbb{P}^{\epsilon} \left[X_{[d]} \right] \right\rangle [\varnothing] . \qquad \blacksquare$$

While Thm. 2.29 only states that the mean parameter of each positive distribution is in the interior, we will construct for each interior point a positive distribution in Chapter 3 by a member of the corresponding exponential family.

2.4.4 Characterization of the Boundary by Faces

Let us now continue with the investigation of the faces of the mean parameter polytope.

Definition 2.30 Given a mean parameter polytope $\mathcal{M}_{\mathcal{S},\nu}$ in the half space representation of Thm. 2.28, and any subset $\mathcal{I} \subset [n]$ we say that the set

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

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 defined faces by the additional sharp satisfaction of a subset of the half-space inequalities. In this way, the faces build the boundary of $\mathcal{M}_{\mathcal{S},\nu}$. This can be easily verified, since for any vector $\mu[L] \in \mathcal{M}_{\mathcal{S},\nu}$, 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 $\mathcal{S}$ is not minimal with respect to ν, we find a non-vanishing vector $V[L]$ and a scalar $\lambda \in \mathbb{R}$ such that

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

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

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

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

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

Thus, the polytope $\mathcal{M}_{\mathcal{S},\nu}$ 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 $\mathcal{M}_{\mathcal{S},\nu}$ is a face itself, since it satisfies these half-space inequalities sharply.

Lemma 2.31 *For each face $Q^{\mathcal{T}}_{\mathcal{S},\nu}$ we have*

$$Q^{\mathcal{T}}_{\mathcal{S},\nu} = \operatorname{conv}\left(\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \in (\sigma^{\mathcal{S}})^{-1}(Q^{\mathcal{T}}_{\mathcal{S},\nu}),\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

Proof. This holds, since each face is the convex hull of the contained vertices (see Proposition 2.2 and 2.3 in [Zie13]). Since the vertices are contained in the image of the statistic encoding $\sigma^{\mathcal{S}}$, the vertices contained in $Q^{\mathcal{T}}_{\mathcal{S},\nu}$ are contained in the set

$$\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \in (\sigma^{\mathcal{S}})^{-1}(Q^{\mathcal{T}}_{\mathcal{S},\nu}).$$ ∎

Lem. 2.31 implies in particular, that faces are mean parameter polytopes with respect to refined base measures. For reference in later chapters, we define these refined base measures next as face measures.

Definition 2.32 The face measure to the face $Q^{\mathcal{T}}_{\mathcal{S},\nu}$ of $\mathcal{M}_{\mathcal{S},\nu}$ is the boolean tensor $\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right]$ with coordinates to $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ by

$$\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]} = x_{[d]}\right] = \begin{cases} 1 & \text{if} \quad \gamma^{x_{[d]}} \in Q^{\mathcal{T}}_{\mathcal{S},\nu} \\ 0 & \text{else} \end{cases}.$$

We now specify the mean parameter polytope to any face using the face measure as a refinement of the base measure.

Theorem 2.33 *For any face $Q^{\mathcal{T}}_{\mathcal{S},\nu}$ of $\mathcal{M}_{\mathcal{S},\nu}$, we have with the refined base measure*

$$\tilde{\nu}\left[X_{[d]}\right] = \left\langle \nu\left[X_{[d]}\right], \nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right]\right\rangle \left[X_{[d]}\right]$$

that

$$Q^{\mathcal{T}}_{\mathcal{S},\nu} = \mathcal{M}_{\mathcal{S},\tilde{\nu}}.$$

Proof. We notice that for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$, $x_{[d]} \in (\sigma^{\mathcal{S}})^{-1}(Q^{\mathcal{T}}_{\mathcal{S},\nu})$ is equal to $\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]} = x_{[d]}\right] = 1$ and thus

$$\left\{x_{[d]} \,:\, x_{[d]} \in (\sigma^{\mathcal{S}})^{-1}(Q^{\mathcal{T}}_{\mathcal{S},\nu}),\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right\} = \left\{x_{[d]} \,:\, \tilde{\nu}\left[X_{[d]} = x_{[d]}\right] = 1\right\}.$$

In combination with Lem. 2.31 we then get

$$\begin{aligned} Q^{\mathcal{T}}_{\mathcal{S},\nu} &= \operatorname{conv}\left(\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \in (\sigma^{\mathcal{S}})^{-1}(Q^{\mathcal{T}}_{\mathcal{S},\nu}),\, \nu\left[X_{[d]} = x_{[d]}\right] = 1\right) \\ &= \operatorname{conv}\left(\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \,:\, \tilde{\nu}\left[X_{[d]} = x_{[d]}\right] = 1\right) \\ &= \mathcal{M}_{\mathcal{S},\tilde{\nu}}. \end{aligned}$$ ∎

Positivity of a distribution with respect to face measures is an equivalent condition for the mean parameter of a distribution to be on a face, as we show next.

Theorem 2.34 *If and only if for a distribution* $\mathbb{P}\left[X_{[d]}\right]$ *and a face* $\mathcal{I}$ *we have*

$$\left\langle \mathbb{P}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle [L] \in Q^{\mathcal{I}}_{\mathcal{S},\nu},$$

then $\mathbb{P}\left[X_{[d]}\right]$ *is representable with respect to the base measure*

$$\left\langle \nu\left[X_{[d]}\right], \nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right]\right\rangle \left[X_{[d]}\right].$$

Proof. We have

$$\mu[L] = \sum_{x_{[d]}} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] \cdot \gamma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right].$$

Now, the $x_{[d]}$ with $\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]} = x_{[d]}\right] = 1$ are exactly those, for which the conditions $\mathcal{I}$ hold straight. If and only if for a $x_{[d]}$ with $\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]} = x_{[d]}\right] = 0$ we have $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0$, one of the conditions $\mathcal{I}$ would not hold straight. Thus, if and only if $\mathbb{P}\left[X_{[d]}\right]$ is representable with respect to $\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right]$, we have $\mu[L] \in Q^{\mathcal{I}}_{\mathcal{S},\nu}$. ∎

For members of exponential families, we can make a stronger statement than Thm. 2.34. If for any $\mathbb{P}\left[X_{[d]}\right] \in \Gamma^{\mathcal{S},\nu}$ and a face $\mathcal{I}$ we have $\left\langle \mathbb{P}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle [L] \in \left(Q^{\mathcal{I}}_{\mathcal{S},\nu}\right)^{\circ}$ then $\mathbb{P}\left[X_{[d]}\right]$ is positive with respect to the base measure

$$\left\langle \nu\left[X_{[d]}\right], \nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right]\right\rangle \left[X_{[d]}\right].$$

Let us now investigate tensor network representations of face measures, based on the basis encoding $\beta^{\mathcal{S}}$ of a statistic. Vertices of $\mathcal{M}_{\mathcal{S},\nu}$ are faces with single elements, that is $\{\mu[L]\}$. By Lem. 2.31 there must be μ must lie in the image of $\sigma^{\mathcal{S}}$, since otherwise $\mathcal{M}_{\mathcal{S},\nu}$ would be empty. The vertex measure is then

$$\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \epsilon_{\mu}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right]$$

Here we use that each $\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)$ has integer-valued coordinates and denote

$$\epsilon_{\mu}\left[Y_{[p]}\right] = \bigotimes_{l \in [p]} \epsilon_{\mu[L=l]}\left[Y_l\right].$$

Theorem 2.35 (Face measure representation) *For any face* $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ *of* $\mathcal{M}$ *we have*

$$\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{\mathcal{I}}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right]$$

where

$$\kappa^{\mathcal{I}}\left[Y_{[p]}\right] = \sum_{\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)} \epsilon_{\mu}\left[Y_{[p]}\right].$$

Proof. For any $\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)$ the tensor

$$\tau^{\mu}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \epsilon_{\mu}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right]$$

is the indicator of the preimage of μ under $\sigma^{\mathcal{S}}$. Since preimages the elements in $Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)$ are disjoint, the support of $\tau^{\mu}\left[X_{[d]}\right]$ is disjoint and their sum

$$\sum_{\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)} \tau^{\mu}\left[X_{[d]}\right]$$

is the indicator of the preimage of $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ under $\sigma^{\mathcal{S}}$, which is the face measure $\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right]$. Exploiting linearity of contraction we have

$$\begin{aligned}
\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right] &= \sum_{\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)} \tau^{\mu}\left[X_{[d]}\right] \\
&= \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \sum_{\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)} \epsilon_{\mu}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right] \\
&= \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{\mathcal{I}}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right] .
\end{aligned}$$

■

Let us now investigate, which normalized face measures can be computed using $\mathcal{S}$ and a hypergraph $\mathcal{G}$.

Example 2.36 (Vertices) Vertices $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ are proper faces of affine dimension 0, that is they consist in single vectors. Since all vertices are in the image $\sigma^{\mathcal{S}}\left(\mathcal{X}\right)$, there exists an index tuple $x_{[d]} \in \mathcal{X}$ such that $\nu\left[X_{[d]} = x_{[d]}\right] = 1$ and

$$Q^{\mathcal{I}}_{\mathcal{S},\nu} = \{\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right]\} .$$

Then $\kappa^{\mathcal{I}}\left[Y_{[p]}\right]$ is the one-hot encoding of the by an interpretation map I assigned index to $\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right]$, that is

$$\kappa^{\mathcal{I}}\left[Y_{[p]}\right] = \epsilon_{I^{-1}\left(\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right]\right)}\left[Y_{[p]}\right] .$$

In particular, the activation core is elementary and the face measure to any vertex is in $\Lambda^{\mathcal{S},\mathrm{EL},\nu}$. ♢

Extending Example 2.36, we can provide a coarse estimation of the hypergraph $\mathcal{G}$ required to decompose $\kappa^{\mathcal{I}}$ for generic faces $Q^{\mathcal{I}}_{\mathcal{S},\nu}$. We notice that $\kappa^{\mathcal{I}}\left[Y_{[p]}\right]$ in Thm. 2.35 is a sparse tensor with basis CP rank $\left|Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \operatorname{im}\left(\sigma^{\mathcal{S}}\right)\right|$ (see Chapter 15).

$$\ell_0\left(\kappa^{\mathcal{I}}\left[Y_{[p]}\right]\right) = \left|\sigma^{\mathcal{S}}\left(\nu^{\mathcal{S},\mathcal{I}}\right)\right|$$

where $\left|\sigma^{\mathcal{S}}\left(\nu^{\mathcal{S},\mathcal{I}}\right)\right|$ is the number of different statistic encoding vectors to the support of the face measure $\nu^{\mathcal{S},\mathcal{I}}$. Using the formalism of sparse CP decompositions Chapter 15, this characterize the basis CP rank of $\kappa^{\mathcal{I}}\left[Y_{[p]}\right]$. However, the basis CP rank is only an upper bound to generic CP rank, which can be loose. By Example 2.37 we provide with the maximal face an example, where the basis CP rank is given by the tensor space dimension, whereas the generic CP rank is one and the

normalized face measure is thus still in $\Lambda^{\mathcal{S},\mathrm{EL}}$.

Example 2.37 (Maximal face) The maximal face $Q^{\varnothing}_{\mathcal{S},\nu} = \mathcal{M}_{\mathcal{S},\nu}$ coincides with the mean polytope itself. In this case the corresponding activation tensor to the face measure is trivial, that is

$$\kappa^{\varnothing}\left[Y_{[p]}\right] = \mathbb{I}\left[Y_{[p]}\right] .$$

$\kappa^{\varnothing}$ is elementary and the normalized face measure $\nu^{\mathcal{S},\varnothing}$ to the maximal face is in $\Lambda^{\mathcal{S},\mathrm{EL}}$. $\lozenge$

2.4.5 Partition into Effective Interiors of Faces

Let us now introduce effective interiors, which enables us to find disjoint partitions of the mean polytope.

Definition 2.38 (Effective Interior) Let $\mathcal{U} \subset \mathbb{R}^p$ be an arbitrary set and $\mathcal{L}$ the minimal affine subspace of $\mathbb{R}^p$ containing $\mathcal{U}$. Then the effective interior, denoted $(\mathcal{U})^{\circ}$ is the the interior of $\mathcal{U}$ in the space $\mathcal{L}$.

Lemma 2.39 *Any polytope is a disjoint union of the effective interiors of its faces, that is*

$$\mathcal{M}_{\mathcal{S},\nu} = \dot{\bigcup_{\mathcal{I}\subset[n]}} \left(Q^{\mathcal{I}}_{\mathcal{S},\nu}\right)^{\circ} .$$

Proof. For any $\mu \in \mathcal{M}_{\mathcal{S},\nu}$ we find a face such that $\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu}$. If $\mu \notin \left(Q^{\mathcal{I}}_{\mathcal{S},\nu}\right)^{\circ}$, then there is a face $Q^{\tilde{\mathcal{I}}}_{\mathcal{S},\nu} \subset Q^{\mathcal{I}}_{\mathcal{S},\nu}$ of smaller affine dimension such that $\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu}$. When continuing this process we reach a face such that $\mu \notin \left(Q^{\mathcal{I}}_{\mathcal{S},\nu}\right)^{\circ}$, since the faces with affine dimension 0 are vertices and they coincide with their effective interior because they contain a single vector. ∎

In this way, we find to each $\mu \in \mathcal{M}_{\mathcal{S},\nu}$ a unique exponential family with statistics $\mathcal{S}$ and base measure by a face measure, such that μ is reproduced by an element of that exponential family. We will show in Chapter 3, that these reproducing distributions maximize the entropy among any other reproducing distribution.

2.4.6 Centers

We provide further intuition on the position of the mean parameter, when interpreting ν and $\mathcal{S}$ as soft and hard constraint mechanisms on a distribution. To this end, let $\mathcal{X}$ be an arbitrary finite state set, where we so far had the set $\bigtimes_{k\in[d]}[m_k]$ for factored representations.

Definition 2.40 We call the mean parameter of a normalized base measure, that is the vector

$$\mu_{\mathcal{S},0,\nu}[L] = \left\langle \langle\nu\rangle\left[X_{[d]}|\varnothing\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle[L] .$$

the center of $\mathcal{M}_{\mathcal{S},\nu}$ (see Figure 2.6).

We have

$$\mu_{\mathcal{S},0,\nu}[L] = \frac{1}{\langle\nu\rangle[\varnothing]} \sum_{x_{[d]}\,:\,\nu\left[X_{[d]}=x_{[d]}\right]=1} \sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right]$$

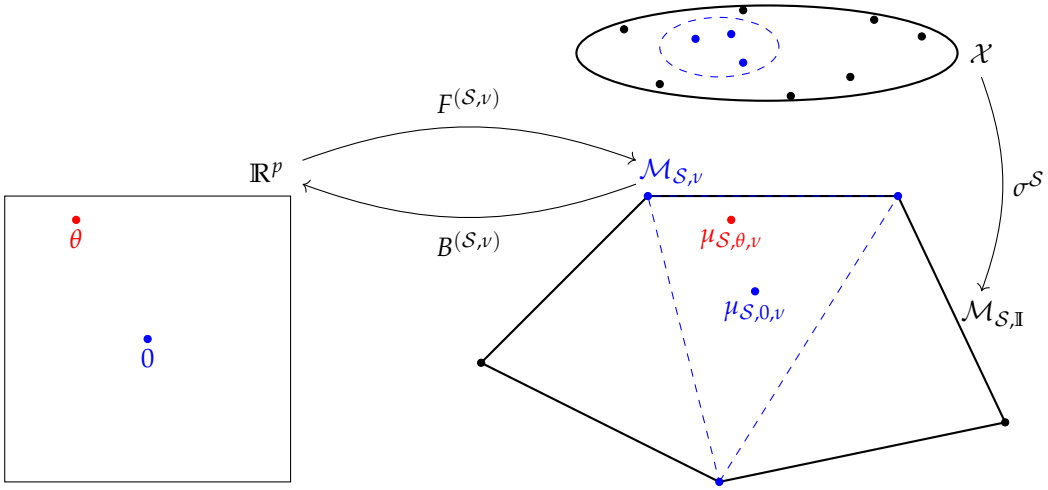


Figure 2.6: Sketch of hard constraints (blue) and soft constraints (red) determining the position of the mean parameter in $\mu_{\mathcal{S},\mathbb{I}}$. In the set of states $\mathcal{X}$, the support of a base measure ν is marked blue, and the base measure is the corresponding subset encoding. The normalized ν has the mean parameter $\mu_{\mathcal{S},0,\nu}$, and the mean polytope $\mathcal{M}_{\mathcal{S},\nu}$ is the convex hull of the image to the statistics encoding $\sigma^{\mathcal{S}}$ restricted on the support set of ν. Given a base measure ν, the corresponding forward and backward maps implement soft contraints depending on canonical parameters θ. The forward mapping maps the zero canonical parameter to $\mu_{\mathcal{S},0,\nu}$, correspond with the normalized base measure. Generic parameters θ (red) are mapped onto the interior $\mathcal{M}_{\mathcal{S},\nu}{}^{\circ}$.

and therefore understand $\mu_{\mathcal{S},0,\nu}$ as the weighted center of $\mathcal{M}_{\mathcal{S},\nu}$, where weights are allocated on the image of the statistics encoding by the cardinality of their pre-images. These concepts are sketched in Figure 2.6 in blue.

The choice of a base measure ν can be regarded as the hard structure of a distribution. Given ν, we further implement soft structure in form of distributions in exponential families $\Gamma^{\mathcal{S},\nu}$. While the normalized base measure ν reproduces only the weighted center of $\mathcal{M}_{\mathcal{S},\nu}$, canonical parameters can be chosen to alter to any interior point in $\mathcal{M}_{\mathcal{S},\nu}$. This alternations are sketched in Figure 2.6 in red.

2.5 Graphical Models

Graphical models are specific instances of exponential families [WJ08; Mur22]. They combine both the independence approach and the computation approach to tensor network representations of probability distributions. Graphical models provide a 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.

2.5.1 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 [KF09]).

Definition 2.41 (Markov Network) Let $\tau^{\mathcal{G}}$ be a tensor network of non-negative tensors decorating a hypergraph $\mathcal{G}$. Then the Markov Network $\mathbb{P}^{\mathcal{G}}$ to $\tau^{\mathcal{G}}$ is the probability distribution of X_v defined by the tensor

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

We call the denominator

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

the partition function of the tensor network $\tau^{\mathcal{G}}$.

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

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

This can be derived from Thm. 17.1, which established an equivalence of contractions with sequences of consecutive contractions.

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

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

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 2.42 (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 [KF09].

Definition 2.43 (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 Thm. 2.47.

Theorem 2.44 (Hammersley-Clifford) *Given a clique-capturing hypergraph $\mathcal{G}$, the set of positive Markov Networks on the hypergraph coincides with the set of positive probability distributions, such that each for each disjoint subsets of variables A, B, C we have X_A is independent of X_B conditioned on X_C, when C separates A and B in the hypergraph.*

Proof. "⇒": Let there be a hypergraph $\mathcal{G}$, a Markov Network $\tau^{\mathcal{G}}$ 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

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

By Thm. 2.14, it now follows that X_A is independent of X_B conditioned on X_C.

"⇐": The converse direction, i.e. that positive distributions respecting the conditional independence assumptions are representable as Markov Networks, is known as the Hammersley-Clifford Theorem (see [CH71]), which we will proof later in Sect. 13.4 of Chapter 13. ∎

From the proof of Thm. 2.44 Markov Networks with zero coordinates still satisfy the conditional independence assumption. However, the reverse is not true, that is there are distributions with vanishing coordinates, which satisfy the conditional independence assumptions, but cannot be represented as a Markov Network (see Example 4.4 in [KF09]).

2.5.2 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 normalized according to the direction. We will observe, that if the hypergraph is in addition acyclic, then each tensor core coincides with the conditional distribution of the underlying Markov Network. To introduce Bayesian Networks, we extend Def. 0.9 by introducing the property of acyclicity for hypergraphs.

Definition 2.45 A directed path is a sequence $v_0, \dots v_r$ such that for any $l \in [r]$ there is an hyperedge $e = (e^{\text{in}}, e^{\text{out}}) \in \mathcal{E}$ such that $v_l \in e^{\text{in}}$ and $v_{l+1} \in e^{\text{out}}$. We call the hypergraph $\mathcal{G}$ acyclic, if there is no path with $r > 0$ such that $v_0 = v_r$. Given a directed hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we define for any node $v \in \mathcal{V}$ its parents by

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

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

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

Definition 2.46 (Bayesian Network) Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic hypergraph with edges of the form

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

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

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

which represents the probability distribution

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

By definition each tensor decorating a hyperedge is directed with $X_{\text{Pa}(v)}$ incoming and X_v outgoing. Thus, the directionality of the hypergraph is reflected in each tensor decorating a

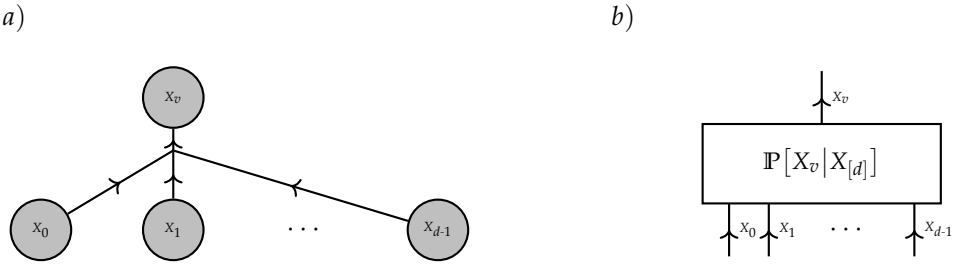


Figure 2.7: Example of a Factor of a Bayesian Network to the node X_v with parents $X_0, \ldots, X_{d-1}$, as an directed edge a) which is decorated by a directed tensor b).

directed hyperedge. This allows us to verify with Thm. 13.15 that their contraction defines a probability distribution.

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 2.47 (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.

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

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

Since $\prec$ is a topological ordering we have

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

We apply the assumed conditional independence with Thm. 2.16 and get

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

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

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

$$= \left\langle \{\mathbb{P}\left[X_{\mathrm{NonDes}(v)} \middle| X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)}\right], \mathbb{P}\left[X_v \middle| X_{\mathrm{Pa}(v)} = x_{\mathrm{Pa}(v)}\right]\} \right\rangle \left[X_v, X_{\mathrm{NonDes}(v)}\right]$$

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 Thm. 2.14, it now follows that X_v is independent of $\mathrm{NonDes}(v)$ conditioned on $\mathrm{Pa}(v)$. ∎

2.5.3 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 2.48 *Let $\tau^{\mathcal{G}}$ be a tensor network on a directed acyclic hypergraph, such that the edges are of the structure*

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

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

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

In particular, $\tau^{\mathcal{G}}$ 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 $\tau^{(\varnothing,\{v\})}$ is then normalized with no incoming variables and we thus have

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

and

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

$|\mathcal{V}| - 1 \to |\mathcal{V}|$: 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 $\tau^{\tilde{\mathcal{G}}}$ the tensor network on the hypergraph $\tilde{\mathcal{G}} = \{\mathcal{V}/\{v\}, \mathcal{E}/\{(\mathrm{Pa}(v), \{v\})\}\}$ with decorations inherited from $\tau^{\mathcal{G}}$. With Thm. 17.1, the directionality of $\tau^{(\mathrm{Pa}(v),\{v\})}$ and the induction assumption on $\tau^{\tilde{\mathcal{G}}}$ we have

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

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

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

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

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

Thm. 2.48 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 normalization 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.

Example 2.49 (Hidden Markov Models) Hidden Markov Models are examples of Bayesian Networks, constructed as follows. Let us recall Markov Chains as investigated in Thm. 2.17 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 Thm. 2.10 given the order of variables

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

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 Thm. 2.16. This results in the decomposition (see Figure 2.8b)

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

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

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

We notice that this is a Bayesian Network on a directed acyclic hypergraph $\mathcal{G}$ (see Figure 2.8a) 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

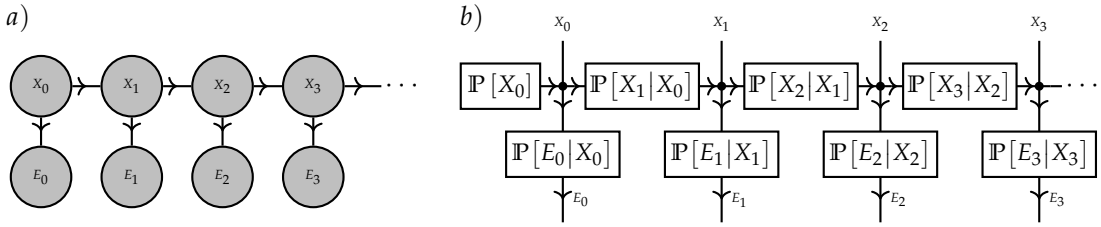


Figure 2.8: Decomposition of a probability distribution in the Hidden Markov Model, consistent of state variables X_k and observation variables E_k. Given the models conditional independence assumptions, the distribution is a Bayesian Network on the directed hypergraph a). The hypergraph is decorated 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 $\mathbb{P}[E_k|X_k]$.

- $(\{X_k\}, \{E_k\})$ 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 Thm. 2.47. ♢

2.5.4 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 an exponential family to any hypergraph, such that any positive Markov Networks on this hypergraph is in the exponential family.

Theorem 2.50 (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*

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

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

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

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

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

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

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

where

$$\theta_e[X_e] = \ln[\tau^e[X_e]]$$

and L enumerates a concatenation of the states of X_e. Then, the Markov Network is the member with

canonical parameter θ of the exponential family with trivial base measure, statistic $\mathcal{S}^{\mathcal{G}}$, which we denote by $\Gamma^{\mathcal{S}^{\mathcal{G}},\mathbb{I}}$.

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

$$\begin{aligned}\prod_{e\in\mathcal{E}} \tau^e\left[X_e = x_e\right] &= \exp\left[\sum_{e\in\mathcal{E}} \ln\left[\tau^e\left[X_e = x_e\right]\right]\right] \\ &= \exp\left[\sum_{e\in\mathcal{E}} \theta_e\left[X_e = x_e\right]\right] \\ &= \exp\left[\sum_{e\in\mathcal{E}} \left\langle \theta_e\left[X_e\right], s_e(x_{\mathcal{V}})\right\rangle[\varnothing]\right] .\end{aligned}$$

By contraction, we further have

$$\left\langle \mathcal{S}\left(X_{\mathcal{V}}, L\right), \theta\left[L\right]\right\rangle\left[X_{\mathcal{V}}\right] = \sum_{e\in\mathcal{E}} \left\langle \mathcal{S}_e\left[X_{\mathcal{V}}, X_e\right], \theta_e\left[X_e\right]\right\rangle\left[X_{\mathcal{V}}\right]$$

we thus get with the above

$$\left\langle \left\{\tau^e \,:\, e\in\mathcal{E}\right\}\right\rangle\left[X_{\mathcal{V}}\right] = \exp\left[\left\langle \mathcal{S}\left(X_{\mathcal{V}}, L\right), \theta\left[L\right]\right\rangle\left[X_{\mathcal{V}}\right]\right] .$$

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

$$\left\langle \left\{\tau^e \,:\, e\in\mathcal{E}\right\}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] = \left\langle \exp\left[\left\langle \theta, \mathcal{S}\right\rangle\left[X_{\mathcal{V}}\right]\right]\right\rangle\left[X_{\mathcal{V}}|\varnothing\right] .$$

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 $\mu_e\left[X_e\right]$. They are often referred to as beliefs in the literature, as introduced by [Pea88]. 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 Thm. 2.50, in case of tree hypergraphs, the

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

That is, the polytope of realizable mean parameters consists of those non-negative and normalized 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 [WJ08]).

2.5.5 Representation of Generic Distributions

We now present a universal exponential family, which contains all positive distributions with respect to a base measure. This implies that 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

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

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 $\mathrm{MAX} = (\mathcal{V}, \{\mathcal{V}\})$. We will further revisit this family in Chapter 8, where we will refer to it by the minterm family in order to connect with terminology developed for logical reasoning in Chapter 5.

2.6 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 normalizations 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. [RS19]). The duality becomes clear, when one interprets 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 referred to as "factors" and their coordinatewise logarithm as "features" [KF09]. 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. 2.43) 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 measurable 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 normalizable. By restriction in this work to finite state spaces of factored systems, we were able to exclude such situations.

3

Probabilistic 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.

3.1 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.

3.1.1 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_q to tensors q, which are enumerating the set $\mathrm{im}\,(q)$. More details on this scheme are provided in Chapter 14, see Def. 14.8 therein.

Definition 3.1 The marginal query of a probability distribution $\mathbb{P}\left[X_{[d]}\right]$ by a query statistic

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

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

$$\mathbb{P}\left[Y_q\right] = \left\langle \mathbb{P}\left[X_{[d]}\right], \beta^q\left[Y_q, X_{[d]}\right]\right\rangle \left[Y_q\right] .$$

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

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

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 q conditioned

on g is the matrix $\mathbb{P}\left[Y_q \middle| Y_g\right] \in \mathbb{R}^{|\mathrm{im}(q)|} \otimes \mathbb{R}^{|\mathrm{im}(g)|}$ defined as the normalization

$$\mathbb{P}\left[Y_q \middle| Y_g\right] = \left\langle \mathbb{P}\left[X_{[d]}\right], \beta^q\left[Y_q, X_{[d]}\right], \beta^g\left[Y_g, X_{[d]}\right] \right\rangle \left[Y_q \middle| Y_g\right] .$$

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 $q : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}$. When understanding q 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 q, since

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

This contraction equation follows from the more general Cor. 14.14, which will be shown in Chapter 14.

Expectation queries compute mean parameters to a distribution, see Def. 2.26. Given a statistic $\mathcal{S}$ and a distribution $\mathbb{P}\left[X_{[d]}\right]$, each feature s_l for $l \in [p]$ corresponds with a mean parameter

$$\mathbb{E}\left[q\right] = \left\langle \mathbb{P}\left[X_{[d]}\right], s_l \right\rangle [\varnothing] = \mu\left[L = l\right] .$$

3.1.2 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 3.2 Given a tensor $\tau\left[X_{[d]}\right]$ the mode query is the problem

$$\mathrm{argmax}_{x_{[d]} \in \bigtimes_{k\in[d]}[m_k]} \tau\left[X_{[d]} = x_{[d]}\right] .$$

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

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

With this we have

$$\begin{aligned}\max_{x_{[d]}} \tau\left[X_{[d]} = x_{[d]}\right] &= \max_{x_{[d]}} \left\langle \tau\left[X_{[d]}\right], \epsilon_{x_{[d]}}\left[X_{[d]}\right] \right\rangle [\varnothing] \\ &= \max_{\mu\left[X_{[d]}\right] \in \Lambda^{\delta,\mathrm{MAX},\mathbb{I}}} \left\langle \tau\left[X_{[d]}\right], \mu\left[X_{[d]}\right] \right\rangle [\varnothing] .\end{aligned}$$

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

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

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

Let us note, that we can also approach modified mode queries, where we restrict to $\mathcal{U} \subset \bigtimes_{k\in[d]}[m_k]$, namely

$$\mathrm{argmax}_{x_{[d]}\in\mathcal{U}}\tau\left[X_{[d]} = x_{[d]}\right] .$$

We can choose the base measure by the subset encoding (see Chapter 14) of $\mathcal{U}$, namely

$$\nu\left[X_{[d]}\right] = \sum_{x_{[d]}\in\mathcal{U}} \epsilon_{x_{[d]}}\left[X_{[d]}\right] ,$$

and conclude that

$$\mathrm{argmax}_{\mu[X_{[d]}]\in\Lambda^{\delta,\mathrm{MAX},\nu}}\left\langle\tau\left[X_{[d]}\right],\mu\left[X_{[d]}\right]\right\rangle[\varnothing] = \mathrm{conv}\left(\epsilon_{x_{[d]}}\left[X_{[d]}\right] : x_{[d]} \in \mathrm{argmax}_{x_{[d]}\in\mathcal{U}}\tau\left[X_{[d]} = x_{[d]}\right]\right) .$$

3.1.3 Energy Representations

For exponential families (see Sect. 2.3) 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 3.3 *Let $\phi\left[X_{[d]}\right]$ be an energy tensor and $\mathbb{P}\left[X_{[d]}\right] = \left\langle\exp\left[\phi\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right]$ the corresponding distribution. For disjoint subsets $A, B \subset [d]$ with $A \cup B = [d]$ and any x_B we have*

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

Proof. To show the theorem, we use a generic simplification property of coordinatewise transforms, which we will show as Lem. 13.6 in Chapter 13 and get

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

Based on this we get

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

■

Importantly, Thm. 3.3 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

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

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

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

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.

3.2 Sampling

Let us here investigate how to draw samples from a probability 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.

3.2.1 Forward Sampling

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 Thm. 2.10) of a probability distribution, namely

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

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

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

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

Input: Probability distribution $\mathbb{P}$
Output: Exact sample $x_0,\ldots,x_{d-1}$ of $\mathbb{P}$

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

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

end for
return $x_0,\ldots,x_{d-1}$

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 Thm. 2.16 in combination with Thm. 2.47) to get

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

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 [KF09]). Sampling along the topological variable order requires only tractable to answer conditional queries on the Bayesian Network.

3.2.2 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 [KF09]).

Algorithm 2 Gibbs Sampling

Input: Probability distribution $\mathbb{P}$
Output: Approximative sample $x_0, \ldots, x_{d-1}$ of $\mathbb{P}$

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

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

  end for
end while
return $x_0, \ldots, x_{d-1}$

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 $\tau^{\mathcal{G}}$ on $\mathcal{G} = ([d], \mathcal{E})$ and any $k \in [d]$ we have

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

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

$$\mathbb{P}\left[X_k \middle| X_{[d]/\{k\}} = x_{[d]/\{k\}}\right] = \left\langle \tau^{\mathcal{G}} \right\rangle \left[X_k, X_{[d]/\{k\}} \middle| \varnothing\right]$$

$$= \left\langle \{\tau^e \left[X_k, X_{e/\{k\}} = x_{e/\{k\}}\right] : e \in \mathcal{E}, k \in e\} \right\rangle [X_k|\varnothing] \ .$$

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, referred to as Markov blanket, such that the other variables are conditionally independent. This follows from the characterization of the conditional independences in Markov Networks, which has been shown in Thm. 2.44, 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 $\mathbb{P}\left[X_{[d]}\right] = \left\langle \exp\left[\phi\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$. Using Thm. 3.3, we have

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

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.

3.2.3 Simulated Annealing

Simulated annealing is an adapted sampling scheme targeting 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 $\mathbb{P}\left[X_{[d]}\right] = \left\langle \exp\left[\phi\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$. We 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

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

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

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

i 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 $\left\langle \exp\left[\beta \cdot \phi\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$. 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 referred to as simulated annealing.

3.3 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 likelihood 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.

3.3.1 Empirical Distributions

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

Definition 3.4 Given a dataset $\left((x_0^j, \dots, x_{d-1}^j) : j \in [m]\right)$ of samples of the factored system we define the sample selector map

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

elementwise by

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

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

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

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

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

$$\beta^D \left[X_{[d]}, J\right] = \sum_{j \in [m]} \epsilon_{x_0^j, \dots, x_{d-1}^j} \left[X_{[d]}\right] \otimes \epsilon_j [J] .$$

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

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

We can therefore interpret each coordinate of the empirical distribution as the relative frequency of the corresponding state in the observed data.

The basis 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 3.1) based on matrices encoding its variables.

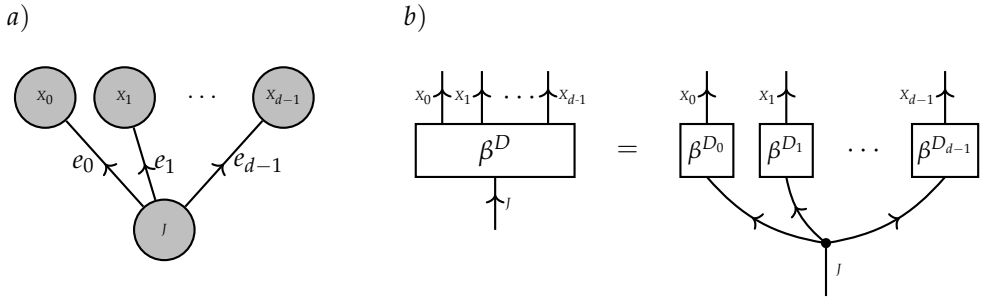


Figure 3.1: Decomposition of the basis encoding of a sample selector map to a dataset $\left((x_0^j, \ldots, x_{d-1}^j) : j \in [m]\right)$. a) Interpretation as a sample selection variable J selecting states for the variables $X_{[d]}$ according to the enumerated dataset. b) Corresponding decomposition of the basis encoding β^D into a tensor network in the basis CP Format (see Sect. 15.1.2), where $\tau^{e_k} = \beta^{D_k}$.

Theorem 3.5 *Given a data map $D : [m] \to \bigtimes_{k\in[d]}[m_k]$ we define for $k \in [d]$ its coordinate maps*

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

by

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

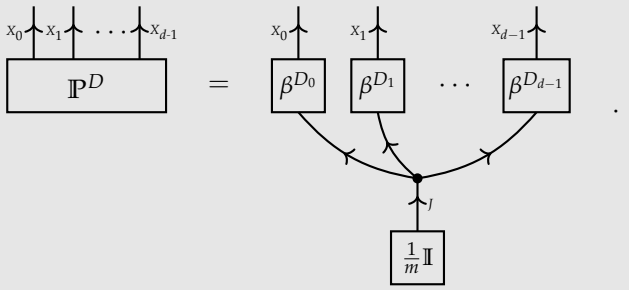
We then have

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

and

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

In a contraction diagram this decomposition is represented as



Proof. The first claim is a special case of Thm. 15.20, to be shown in Chapter 14. To show the second claim we notice

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

With the first claim it now follows that

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

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

From the proof of Thm. 3.5 we notice that the scalar $\frac{1}{m}$ could be assigned with any core in a representation of $\mathbb{P}^D$, and the core $\mathbb{I}[J]$ is thus redundant in the contraction representation. However, creating the core $\frac{1}{m}\mathbb{I}[J]$ provides us with a simple interpretation of the empirical distribution. We can understand $\frac{1}{m}\mathbb{I}[J]$ as the uniform probability distribution over the samples, which is by the map D forwarded to a distribution over $\bigtimes_{k\in[d]}[m_k]$. The one-hot encoding of each sample is itself a probability distribution, which is understood as conditioned on the respective state of the sample selection variable J. The conditional distribution β^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 3.1), the variable J serves as single parent for each categorical variable X_k.

3.3.2 Likelihood Loss

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

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

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

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

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 [Mac03].

Let us now transform the likelihood to find an representation involving the empirical distribution (see Def. 3.4), for which efficient tensor network decompositions have been derived in Thm. 3.5. Applying a scaled logarithm we get

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

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 3.6 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

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

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

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

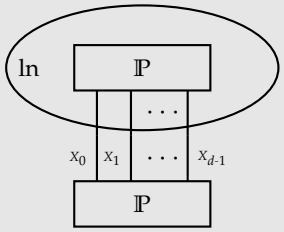
3.3.3 Entropic Interpretation

The Shannon entropy, which has been introduced in the seminal paper [Sha48], 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 3.7 (Shannon entropy) The Shannon entropy of a distribution $\mathbb{P}\left[X_{[d]}\right]$ is the quantity

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

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

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

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 [Mac03]). 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 q^l (see Def. 3.1) 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

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

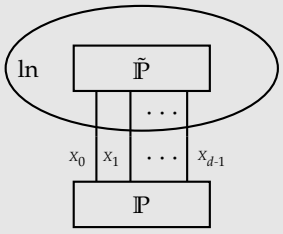
where $\Lambda^{\delta,\text{MAX},\mathbb{I}}$ is the set of probability distributions given a factored system. We study instances of this maximal entropy problem later in Sect. 3.5.1, where we show that its solution is a member of the exponential family, which statistic is build by the query tensors. 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 3.8 (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

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

The cross-entropy is captured by the tensor network diagram

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

The Kullback-Leiber divergence or relative entropy 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 [CT06]) 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 and vanishes if and only if both distributions coincide.

We in the next lemma provide an entropic interpretation of maximum likehlihood estimation as defined in Def. 3.6.

Lemma 3.9 *The maximum likelihood estimation Problem* ($\mathrm{P}^{\mathrm{M}}_{\Gamma,\mathbb{P}^D}$) *is equivalent to the minimization of cross-entropy and Kullback-Leibler divergence, that is*

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

Proof. Comparing the log-likelihood loss in Def. 3.6 with the cross-entropy in Def. 3.8, 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. Further, 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. ∎

More general than Maximum Likelihood Estimation, we define the moment projection of an arbitrary distribution $\mathbb{P}^*$ onto a set Γ of probability distributions as the problem

$$\operatorname{argmin}_{\mathbb{P}\in\Gamma}\mathbb{H}\left[\mathbb{P}^*,\mathbb{P}\right] . \tag{$\mathrm{P}^{\mathrm{M}}_{\Gamma,\mathbb{P}^*}$}$$

With Lem. 3.9 we have established, that maximum likelihood estimation is the moment projection of the empirical distribution onto a set Γ.

We further define the information projection an arbitrary distribution $\mathbb{P}^*$ onto a set Γ of probability distributions as the problem

$$\operatorname{argmin}_{\mathbb{P}\in\Gamma}\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}||\mathbb{P}^*\right] . \tag{$\mathrm{P}^{\mathrm{I}}_{\Gamma,\mathbb{P}^*}$}$$

The relative entropy is not symmetric, thus the information projection ($\mathrm{P}^{\mathrm{I}}_{\Gamma,\mathbb{P}^*}$) and the moment projections do not coincide in general. The differences of both are discussed in Chapter 8 in [KF09].

Example 3.10 (Cross entropy with respect to exponential families) If $\tilde{\mathbb{P}}$ is a member of an exponential family, we have

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

The last term vanishes, given the convention $0\cdot\ln\left[0\right] = 0$, 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. 2.2). If $\mathbb{P}$ is representable with respect to ν, we can abbreviate the cross-entropy to

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

♢

3.4 Variational Inference in Exponential Families

With the maximum likelihood method we have already formulated inference tasks by optimization problem with variations of the probability distribution. In a broader perspective, we now investigate variational inference problems using the canonical and mean parametrization of exponential families.

3.4.1 Forward and Backward Mappings

While defined for arbitrary distributions (see Def. 2.26), we now consider mean parameters to members of exponential families. First of all, they provide an alternative parameterization to the canonical parameter θ. The computation of the mean parameter to a given canonical parameter and vice versa are the central inference problems in exponential families. We first formalize these inference problems by the forward and backward mapping and then provide in this section further insights into these mappings.

Definition 3.11 Let $\mathcal{S}$ be a statistic and ν a base measure and consider the exponential family $\Gamma^{\mathcal{S},\nu}$ The function

$$F^{(\mathcal{S},\nu)} : \mathbb{R}^p \to \mathcal{M}_{\mathcal{S},\nu} \subset \mathbb{R}^p \quad , \quad F^{(\mathcal{S},\nu)}(\theta) = \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)} \left[X_{[d]} \right], \sigma^{\mathcal{S}} \left[X_{[d]}, L \right] \right\rangle [L]$$

is called the forward mapping of the exponential family. Any inverse, that is a function

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

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

We notice that the domain of $F^{(\mathcal{S},\nu)}$ is always $\mathbb{R}^p$, since the coordinates of $F^{(\mathcal{S},\nu)}(\theta)$ are for any $\theta \in \mathbb{R}^p$ summations over finitely many products.

We already know by Thm. 2.29, that distributions representable by ν have mean parameters in the interior of $\mathcal{M}_{\mathcal{S},\nu}$. We now state that the elements of the corresponding exponential family $\Gamma^{\mathcal{S},\nu}$, which are by construction representable by ν, are expressive enough to reproduce the whole interior of $\mathcal{M}_{\mathcal{S},\nu}$.

Theorem 3.12 *For any exponential family $\Gamma^{\mathcal{S},\nu}$ the image of the forward mapping is the effective interior (see Def. 2.38) of $\mathcal{M}_{\mathcal{S},\nu}$, that is*

$$\operatorname{im}\left(F^{(\mathcal{S},\nu)} \right) = \left(\mathcal{M}_{\mathcal{S},\nu} \right)^{\circ}.$$

Proof. In case of minimal statistics, we refer for the proof of this statement to Theorem 3.3 in [WJ08]. If $\mathcal{S}$ is not minimal, we find a subset $\tilde{\mathcal{S}}$ of its features such that $\tilde{\mathcal{S}}$ is minimal with respect to ν and there is a matrix $M[L, \tilde{L}]$ such that for any distribution

$$\left\langle \mathbb{P}\left[X_{[d]} \right], \sigma^{\tilde{\mathcal{S}}} \left[X_{[d]}, \tilde{L} \right], M[L, \tilde{L}] \right\rangle [L] = \left\langle \mathbb{P}\left[X_{[d]} \right], \sigma^{\mathcal{S}} \left[X_{[d]}, L \right] \right\rangle [L] \,.$$

This subset $\tilde{\mathcal{S}}$ can be found by iteratively identifying $V[L]$ and λ such that the condition in Def. 2.23 is violated, and dropping a feature s_l with $V[L = l] \neq 0$. At each manipulation step the expressivity of the exponential family stays constant and thus $\Gamma^{\tilde{\mathcal{S}},\nu} = \Gamma^{\mathcal{S},\nu}$. The matrix $M[L, \tilde{L}]$ can be constructed based on the linear dependencies of the dropped features on the remaining. The procedure terminates, when there is no pair $V[L], \lambda$, which is equal to $\tilde{\mathcal{S}}$ being minimal with

respect to ν. We then have

$$\mathcal{M}_{\mathcal{S},\nu} / \bigcup_{Q^{\mathcal{T}}_{\mathcal{S},\nu} \neq \mathcal{M}_{\mathcal{S},\nu}} Q^{\mathcal{T}}_{\mathcal{S},\nu} = \{ \langle \mu\left[\tilde{L}\right], M[L,\tilde{L}] \rangle [L] : \mu\left[\tilde{L}\right] \in \left(\mathcal{M}_{\mathcal{S},\nu}\right)^{\circ} \}.$$

and using that $\tilde{\mathcal{S}}$ is minimal we get

$$\operatorname{im}\left(F^{(\mathcal{S},\nu)}\right) = \mathcal{M}_{\mathcal{S},\nu} / \bigcup_{Q^{\mathcal{T}}_{\mathcal{S},\nu} \neq \mathcal{M}_{\mathcal{S},\nu}} Q^{\mathcal{T}}_{\mathcal{S},\nu}. \qquad \blacksquare$$

Forward and backward mappings in an exponential family $\Gamma^{\mathcal{S},\nu}$ are the central classes of inference, which transform the description of a member by a canonical parameter into mean parameters and vise versa. Forward mappings calculate to a canonical parameter $\theta[L]$ the corresponding mean parameter $\mu[L]$. For any $\theta[L]$ we have a closed form representation of this expectation query by the moment matching condition

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

The forward map is thus a collection of expectation queries (see Def. 3.1) to compute the coordinates of the mean parameter. The query s_l asked against $\mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right]$ computes the coordinate $\mu[L = l]$. The contraction by the moment matching condition can, however, be infeasible, since it requires the instantiation of the probability distribution, which can be done by basis encodings of the statistic. In this section, we provide alternative characterization of the forward mapping and approximations of it, which can be computed based on the selection encoding instead. Following [WJ08], we can characterize the forward mapping to exponential families as a variational problem and provide an alternative characterization to this contraction.

3.4.2 Variational Formulation

We now formulate the forward and backward inference problems in exponential families as convex optimization problems. To this end we use the cumulative function $A^{(\mathcal{S},\nu)}$ and its conjugate dual

$$\left(A^{(\mathcal{S},\nu)}\right)^*(\mu) = \max_{\theta \in \mathbb{R}^p} \langle \mu, \theta \rangle [\varnothing] - A^{(\mathcal{S},\nu)}(\theta).$$

Lemma 3.13 *Let $\mathcal{S}$ be a statistic, ν a basemeasure. Let us further choose $\theta[L] \in \mathbb{R}^p$ and let $\mu[L]$ be the mean parameter to $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$. For the gradient of $A^{(\mathcal{S},\nu)}$ evaluated at θ we have*

$$\nabla_{\tilde{\theta}[L]}|_{\theta} A^{(\mathcal{S},\nu)}(\tilde{\theta}) = \mu[L]$$

If the statistic $\mathcal{S}$ is minimal with respect to ν, then also

$$\nabla_{\tilde{\mu}[L]}|_{\mu} \left(A^{(\mathcal{S},\nu)}\right)^*(\tilde{\mu}) = \theta[L] .$$

Proof. We have

$$\begin{aligned}
\nabla_{\theta}|_{\tilde{\theta}} A^{(\mathcal{S},\nu)}(\tilde{\theta}) &= \nabla_{\theta}|_{\tilde{\theta}} \ln\left[\left\langle \exp\left[\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \theta[L]\right\rangle \left[X_{[d]}\right]\right], \nu\right\rangle [\varnothing]\right] \\
&= \frac{\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \exp\left[\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \theta[L]\right\rangle \left[X_{[d]}\right]\right], \nu\right\rangle [L]}{\left\langle \exp\left[\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \theta[L]\right\rangle \left[X_{[d]}\right]\right]\right\rangle [\varnothing]}
\end{aligned}$$

$$= \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)} \left[X_{[d]} \right], \sigma^{\mathcal{S}} \left[X_{[d]}, L \right] \right\rangle [L] \,.$$

For the proof of the second claim we refer to Appendix B.2 in [WJ08]. ∎

Theorem 3.14 (Variational backward mapping) *For any $\mu \in \mathcal{M}^\circ$ choose*

$$\theta\,[L] \in \operatorname{argmax}_\theta \langle \mu, \theta \rangle\,[\varnothing] - A^{(\mathcal{S},\nu)}(\theta)\,.$$

Then μ is the mean parameter to $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$.

Proof. The maximization over θ is an unconstrained concave maximization problem and the optimum is characterized by

$$0 = \nabla_\theta|_{\tilde{\theta}} \left(\langle \mu, \theta \rangle\,[\varnothing] - A^{(\mathcal{S},\nu)}(\theta) \right)$$

which reads

$$\mu\,[L] = \nabla_\theta|_{\tilde{\theta}[L]} A^{(\mathcal{S},\nu)}(\theta)\,.$$

With Lem. 3.13, the gradient vanishes if and only if μ is the mean parameter to $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$. ∎

The backward mapping is closely connected with the conjugate dual of $A^{(\mathcal{S},\nu)}$. In particular, while the conjugate dual is defined by the maximization problem

$$\left(A^{(\mathcal{S},\nu)}\right)^*(\mu) = \max_{\theta \in \mathbb{R}^p} \langle \mu, \theta \rangle\,[\varnothing] - A^{(\mathcal{S},\nu)}(\theta)$$

the backward mapping returns the position of the maximum in $\mathbb{R}^p$.

Lemma 3.15 *For any θ with mean parameter μ we have*

$$\left(A^{(\mathcal{S},\nu)}\right)^*(\mu) = -\mathbb{H}\left[\mathbb{P}^{(\mathcal{S},\theta,\nu)}\right] \,.$$

Proof. We have

$$\theta \in \operatorname{argmax}_\theta \langle \mu, \theta \rangle\,[\varnothing] - A^{(\mathcal{S},\nu)}(\theta)$$

if and only if

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

Therefore

$$\begin{aligned}
\left(A^{(\mathcal{S},\nu)}\right)^*(\mu) &= \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)} \left[X_{[d]} \right], \sigma^{\mathcal{S}} \left[X_{[d]}, L \right], \theta \right\rangle [\varnothing] - A^{(\mathcal{S},\nu)}(\theta) \\
&= \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)} \left[X_{[d]} \right], \ln \left[\mathbb{P}^{(\mathcal{S},\theta,\nu)} \left[X_{[d]} \right] \right] \right\rangle [\varnothing] \\
&= -\mathbb{H}\left[\mathbb{P}^{(\mathcal{S},\theta,\nu)}\right] \,.
\end{aligned}$$

∎

We can use this insight to provide a variational characterization of the forward mapping.

Theorem 3.16 (Variational forward mapping) *Let $\mathcal{S}$ be a minimal statistic with respect to ν. Given θ, there is a unique μ with*

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

and μ is the mean parameter to $\mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right]$. Here, we denote by $\mathbb{P}^{\mathcal{S},\mu,\nu}$ the member of the exponential family with the mean parameter μ.

Proof. By strong duality, we have $\left(A^{(\mathcal{S},\nu)}\right)^{**} = A^{(\mathcal{S},\nu)}$ and

$$A^{(\mathcal{S},\nu)}(\theta) = \max_{\mu \in \mathcal{M}} \langle \mu, \theta \rangle [\varnothing] - \left(A^{(\mathcal{S},\nu)}\right)^*(\mu)\,.$$

The statement then follows from the first order condition, where we use the gradient Lem. 3.13, and the characterization of $\left(A^{(\mathcal{S},\nu)}\right)^*$ by Lem. 3.15. ∎

3.4.3 Maximum Likelihood Problem on Exponential Families

We now characterize the solution of the Maximum Likelihood Problem for exponential families as hypothesis classes using the backward mapping of the family.

Lemma 3.17 *Let $\mathbb{P}^*\left[X_{[d]}\right]$ be a distribution, $\mathcal{S}$ a statistic, and ν a boolean base measure. We build the mean parameter $\mu^*[L] = \left\langle \mathbb{P}^*\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle [L]$ and have the following:*

(1) If $\mu^*[L] \in (\mathcal{M}_{\mathcal{S},\nu})^\circ$ then

$$\min_{\theta[L] \in \mathbb{R}^p} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{\mathcal{S},\theta[L],\nu}\right] = \mathbb{H}\left[\mathbb{P}^{\mathcal{S},B^{\mathcal{S},\nu}(\mu^*[L]),\nu}\right]\,.$$

(2) If $\mu^*[L] \notin (\mathcal{M}_{\mathcal{S},\nu})^\circ$ and $\mu^*[L] \in \overline{\mathcal{M}_{\mathcal{S},\nu}}$ then there is a sequence $(\mu_n[L])_{n\in\mathbb{N}} \subset \mathcal{M}_{\mathcal{F},\mathbb{I}}$ converging coordinatewise to $\mu^*[L]$ and

$$\min_{\theta[L] \in \mathbb{R}^p} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{\mathcal{S},\theta[L],\nu}\right] = \lim_{\mu_n[L] \to \mu^*[L]} \mathbb{H}\left[\mathbb{P}^{\mathcal{S},B^{\mathcal{S},\nu}(\mu_n[L]),\nu}\right]\,.$$

(3) If $\mu^*[L] \notin \overline{\mathcal{M}_{\mathcal{S},\nu}}$ then

$$\min_{\theta[L] \in \mathbb{R}^p} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{\mathcal{S},\theta[L],\nu}\right] = \infty\,.$$

Proof. See Theorem 3.4 in [WJ08]. ∎

The case $\mu^*[L] \notin \mathcal{M}_{\mathcal{S},\nu}$ corresponds with the case that $\mathbb{P}^*\left[X_{[d]}\right]$ is not representable with ν. In more generality, the cross entropy between two distributions is finite, if and only if the support of first in contained in the support of the second. We will relate this property to logical entailment in Chapter 4.

3.4.4 Mode Queries by Annealing

The mode of a distribution is related to the forward mapping of $\beta \cdot \theta$ in the limit $\beta \to \infty$ of low temperatures. To sketch this relation, we recall the variational formulation of mode queries by

$$\operatorname{conv}\left(\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \in \operatorname{argmax}_{x_{[d]}} \left\langle \theta, \mathcal{S}(x_{[d]}) \right\rangle [\varnothing]\right) = \operatorname{argmax}_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \langle \mu, \theta \rangle [\varnothing] \ .$$

Further, for any by the inverse temperature $\beta \neq 0$ annealed canonical parameter $\theta[L]$ (see Sect. 3.2.3) we have

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

In the annealing limit, that is for large β, the entropy term becomes negligible and the forward mapping tends to the convex hull

$$\operatorname{conv}\left(\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \in \operatorname{argmax}_{x_{[d]}} \left\langle \theta, \mathcal{S}(x_{[d]}) \right\rangle [\varnothing]\right) = \operatorname{argmax}_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \langle \mu, \theta \rangle [\varnothing] \ .$$

For a more detailed discussion of this relation, see Theorem 8.1 in [WJ08].

3.5 Maximum Entropy Distributions

We characterize in this section to any mean parameter the reproducing distribution with maximum entropy, and investigate the tensor network representation of them and their modes. As we will show, the distributions with maximal entropy are in exponential families with base measure by the minimal face, which contains the mean parameter.

3.5.1 Entropy Maximization Problem

The entropy maximization problem with respect to matching expected statistics $\mu^* \in \mathcal{M}_{\mathcal{S},\nu}$ is the optimization problem

$$\operatorname{argmax}_{\mathbb{P} \in \Lambda^{\delta,\mathrm{MAX},\nu}} \mathbb{H}\left[\mathbb{P}\right] \quad \text{subject to} \quad \left\langle \mathbb{P}, \sigma^{\mathcal{S}} \right\rangle [L] = \mu^*[L] \tag{$\mathrm{P}^{\mathbb{H}}_{\mathcal{S},\nu,\mu^*}$}$$

where by $\Lambda^{\delta,\mathrm{MAX},\nu}$ we denote all distributions, which are representable with respect to a base measure ν.

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

$$\mu[L = l] = \mathbb{E}[s_l] = \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L = l\right], \mathbb{P}\left[X_{[d]}\right] \right\rangle [\varnothing] \ .$$

We first show, that any solution has a sufficient statistic $\mathcal{S}$ (see Def. 2.18) and is therefore in $\Lambda^{\mathcal{S},\mathrm{MAX},\nu}$.

Theorem 3.18 *Any maximum entropy distribution with respect to a moment constraint on $\mathcal{S}$ and a base measure ν has the sufficient statistic $\mathcal{S}$.*

Proof. Let $\mathbb{P}\left[X_{[d]}\right]$ be a feasible distribution for Problem ($\mathrm{P}^{\mathbb{H}}_{\mathcal{S},\nu,\mu^*}$), which does not have a sufficient statistic $\mathcal{S}$. Then we find $x_{[d]}, \tilde{x}_{[d]} \in \bigtimes_{k \in [d]} [m_k]$ with $x_{[d]} \neq \tilde{x}_{[d]}$, $\mathcal{S}\left(x_{[d]}\right) = \mathcal{S}\left(\tilde{x}_{[d]}\right)$,

$\nu\left[X_{[d]} = x_{[d]}\right] \neq 0$, $\nu\left[X_{[d]} = \tilde{x}_{[d]}\right]$ and $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] \neq \mathbb{P}\left[X_{[d]} = \tilde{x}_{[d]}\right]$. We then define a distribution $\tilde{\mathbb{P}}[X_{[d]}]$ coinciding with $\mathbb{P}\left[X_{[d]}\right]$ except for the coordinates $x_{[d]}, \tilde{x}_{[d]}$, where we set

$$\tilde{\mathbb{P}}[X_{[d]} = x_{[d]}] = \tilde{\mathbb{P}}[X_{[d]} = \tilde{x}_{[d]}] = \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right] + \mathbb{P}\left[X_{[d]} = \tilde{x}_{[d]}\right]}{2}$$

We have by strict convexity of $u \cdot \ln[u]$ and Jensens inequality

$$\begin{aligned}\mathbb{H}\left[\tilde{\mathbb{P}}\right] - \mathbb{H}\left[\mathbb{P}\left[X_{[d]}\right]\right] &= \mathbb{P}\left[X_{[d]} = x_{[d]}\right] \cdot \ln\left[\mathbb{P}\left[X_{[d]} = x_{[d]}\right]\right] + \mathbb{P}\left[X_{[d]} = \tilde{x}_{[d]}\right] \cdot \ln\left[\mathbb{P}\left[X_{[d]} = \tilde{x}_{[d]}\right]\right] \\ &\quad - \left(\mathbb{P}\left[X_{[d]} = x_{[d]}\right] + \mathbb{P}\left[X_{[d]} = \tilde{x}_{[d]}\right]\right) \cdot \ln\left[\frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right] + \mathbb{P}\left[X_{[d]} = \tilde{x}_{[d]}\right]}{2}\right] \\ &> 0\,.\end{aligned}$$

Since $\tilde{\mathbb{P}}[X_{[d]}]$ is also a feasible distribution with larger entropy than $\mathbb{P}\left[X_{[d]}\right]$, $\mathbb{P}\left[X_{[d]}\right]$ is not a maximum entropy distribution. ∎

We in the following provide a more detailed characterization.

3.5.2 Characterization based on the Mean Polytope

By definition of the mean polytope, Problem ($\mathrm{P}^{\mathbb{H}}_{\mathcal{S},\nu,\mu^*}$) has a feasible distribution if and only if $\mu^*[L] \in \mathcal{M}_{\mathcal{S},\nu}$. If this condition holds, we now characterize the solution of Problem ($\mathrm{P}^{\mathbb{H}}_{\mathcal{S},\nu,\mu^*}$). First of all, we show that the maximum entropy distribution is in the exponential family $\Gamma^{\mathcal{S},\nu}$, when μ does not lie on a proper face of $\mathcal{M}_{\mathcal{S},\nu}$. We then drop this assumption and generalize the statement to exponential families with refined base measures.

Theorem 3.19 *If the only face $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ of $\mathcal{M}_{\mathcal{S},\nu}$ with $\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu}$ is $\mathcal{M}_{\mathcal{S},\nu}$ itself, then the unique solution of the maximum entropy problem Problem ($\mathrm{P}^{\mathbb{H}}_{\mathcal{S},\nu,\mu^*}$) is the distribution*

$$\mathbb{P}^{\mathcal{S},\mu,\nu}[X_{[d]}] \in \Gamma^{\mathcal{S},\nu}$$

with $\left\langle \mathbb{P}^{\mathcal{S},\mu,\nu}[X_{[d]}], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle[L] = \mu[L]$.

Proof. By Thm. 3.12, since by assumption

$$\mu[L] \in \mathcal{M}_{\mathcal{S},\nu} / \bigcup_{Q^{\mathcal{I}}_{\mathcal{S},\nu} \neq \mathcal{M}_{\mathcal{S},\nu}} Q^{\mathcal{I}}_{\mathcal{S},\nu}\,,$$

there is a canonical parameter θ with

$$\left\langle \mathbb{P}^{\mathcal{S},\theta,\nu}[X_{[d]}], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle[L] = \mu[L]\,.$$

For any other feasible distribution $\tilde{\mathbb{P}}[X_{[d]}]$ we also have $\left\langle \tilde{\mathbb{P}}[X_{[d]}], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle[L] = \mu[L]$ and thus

$$\begin{aligned}\mathbb{H}\left[\tilde{\mathbb{P}}, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right] &= -\left\langle \tilde{\mathbb{P}}, \ln\left[\mathbb{P}^{(\mathcal{S},\theta,\nu)}[X_{[d]}]\right]\right\rangle[\varnothing] \\ &= -\left\langle \tilde{\mathbb{P}}, \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \theta[L]\right\rangle\left[X_{[d]}\right]\right\rangle[\varnothing] + A^{(\mathcal{S},\nu)}(\theta)\end{aligned}$$

$$
\begin{aligned}
&= -\langle \theta, \mu \rangle [\varnothing] + A^{(\mathcal{S},\nu)}(\theta) \\
&= \mathbb{H}\left[\mathbb{P}^{(\mathcal{S},\theta,\nu)}\right].
\end{aligned}
$$

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

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

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

Let us highlight the fact, that in Problem ($\mathrm{P}^{\mathbb{H}}_{\mathcal{S},\nu,\mu^*}$) 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. Thm. 3.19 states, that when the maximum entropy problem has a solution (i.e. $\mu^* \in \mathcal{M}_{\mathcal{S},\nu}$), then the solution is in the exponential family to the statistic $\mathcal{S}$.

When $\mu^* \notin (\mathcal{M}_{\mathcal{S},\nu})^\circ$, the mean paramater is by Thm. 2.29 not reproducible by a member of the exponential family $\Gamma^{\mathcal{S},\nu}$. We show that the solution is in this case in a refined exponential family.

Theorem 3.20 *Let $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ be the minimal face of $\mathcal{M}_{\mathcal{S},\nu}$ such that $\mu[L] \in Q^{\mathcal{I}}_{\mathcal{S},\nu}$. Then, the solution of the maximum entropy problem is the distribution*

$$
\mathbb{P}^{\mathcal{S},\mu,\tilde{\nu}}
$$

where the base measure $\tilde{\nu}$ is the refinement of ν by the face measure $\nu^{\mathcal{S},\mathcal{I}}$, that is

$$
\tilde{\nu}\left[X_{[d]}\right] = \left\langle \nu\left[X_{[d]}\right], \nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right].
$$

Proof. By Thm. 2.34 all feasible distributions for the maximum entropy problem have to be representable by the face measure $\nu^{\mathcal{S},\mathcal{I}}$. Since the feasible distributions are further restricted to those representable by ν, they are also representable by the refined base measure $\tilde{\nu}$. Now, by Thm. 2.33 the face itself is a polytope $\mathcal{M}_{\mathcal{S},\tilde{\nu}}$ and the smallest face containing μ is the polytope itself. We thus arrive at the claim by applying Thm. 3.19 on the polytope $\mathcal{M}_{\mathcal{S},\tilde{\nu}}$. ∎

Thm. 3.20 implies, that the by the face measure refined base measure $\tilde{\nu}$ 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 to vanish to lie on the respective face.

3.5.3 Tensor Network Representation

Let us now state a criterion based on face measures for maximum entropy distributions to be in the set of by $\mathcal{S}$ and $\mathcal{G}$ computable distributions $\Lambda^{\mathcal{S},\mathcal{G},\nu}$.

Theorem 3.21 *Given $\mathcal{S}$ and $\mu[L] \in \mathcal{M}_{\mathcal{S},\nu}$, let $\mathcal{I}$ be the smallest face of $\mathcal{M}_{\mathcal{S},\nu}$ such that*

$$
\mu[L] \in Q^{\mathcal{I}}_{\mathcal{S},\nu}\,.
$$

Then the corresponding maximum entropy distribution is in $\Lambda^{\mathcal{S},\mathcal{G},\nu}$ if and only if the face measure (see

Def. 2.32)

$$\kappa^{\mathcal{I}}\left[Y_{[p]}\right] = \sum_{\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu} \cap \mathrm{im}\left(\sigma^{\mathcal{S}}\right)} \epsilon_{\mu}\left[Y_{[p]}\right]$$

is in $\mathcal{T}^{\mathcal{G}}$.

Proof. By Thm. 3.20 the maximum entropy distribution is an element of the exponential family with by the face measure refined base measure $\tilde{\nu}$. Let $\theta\left[L\right]$ be a canonical parameter such that

$$\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \mathbb{P}^{\mathcal{S},\theta,\tilde{\nu}}\left[X_{[d]}\right]\right\rangle [L] = \mu\left[L\right],$$

that is $\mathbb{P}^{\mathcal{S},\theta,\tilde{\nu}}\left[X_{[d]}\right]$ is the maximum entropy distribution. We apply Thm. 2.35 to represent the face measure by

$$\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{\mathcal{I}}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right]$$

Then for the tensor

$$\tau\left[Y_{[p]}\right] = \left\langle \{\alpha^{l,\theta}\left[Y_l\right] : l \in [p]\} \cup \{\kappa^{\mathcal{I}}\left[Y_{[p]}\right]\}\right\rangle \left[Y_{[p]}\right]$$

we have

$$\mathbb{P}^{\mathcal{S},\theta,\tilde{\nu}}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right], \tau\left[Y_{[p]}\right], \nu\left[X_{[d]}\right]\right\rangle \left[X_{[d]}|\varnothing\right].$$

Thus, the maximum entropy distribution is in $\Lambda^{\mathcal{S},\mathcal{G},\nu}$, if τ admits a tensor network decomposition with respect to $\mathcal{G}$. Since the hard activation cores are elementary, this is the case when $\kappa^{\mathcal{I}}$ admits a tensor network decomposition with respect to $\mathcal{G}$. ∎

In the special case of a mean parameter in the interior, Thm. 3.21 implies that the maximum entropy distribution is in $\Lambda^{\mathcal{S},\mathrm{EL},\nu}$, as we show next.

Theorem 3.22 *For any statistic* $\mathcal{S}$*, base measure* ν *and a mean parameter* $\mu\left[L\right] \in \left(\mathcal{M}_{\mathcal{S},\nu}\right)^{\circ}$ *in the interior of the mean parameter polytope, the corresponding maximum entropy distribution is in* $\Lambda^{\mathcal{S},\mathrm{EL},\nu}$.

Proof. If $\mu\left[L\right] \in \mathcal{M}_{\mathcal{S},\nu}{}^{\circ}$ then the only face of $\mathcal{M}_{\mathcal{S},\nu}$ containing $\mu\left[L\right]$ is $\mathcal{M}_{\mathcal{S},\nu}$ itself, that is the face with $\mathcal{I} = \varnothing$. In this case we have $\kappa^{\varnothing}\left[Y_{[p]}\right] = \mathbb{I}\left[Y_{[p]}\right]$, which is an elementary tensor. The claim then follows from Thm. 3.21, when choosing $\mathcal{G} = \mathrm{EL}$. ∎

In general, we find a CP decomposition to the face measure as sketched in Figure 3.2.

Example 3.23 (Representation by the maximal graph) Let us consider the maximal graph $\mathrm{MAX} = ([p], \{[p]\})$, which has a single hyperedge containing all head variables. Since any tensor admits a tensor network decomposition with respect to MAX, by Thm. 3.21 all maximum entropy distributions are in $\Lambda^{\mathcal{S},\mathrm{MAX}}$. This reproduces the claim of Thm. 3.18, that maximum entropy distributions with respect to constraints $\mu\left[L\right] = \left\langle \mathbb{P}, \sigma^{\mathcal{S}}\right\rangle [L]$ therefore always have the sufficient statistic $\mathcal{S}$. ◊

3.6 Modes of Exponential Distributions

We now show that modes of the members of the exponential family coincide with face measures. To this end we introduce face normals and show that the mode of an exponential distribution is

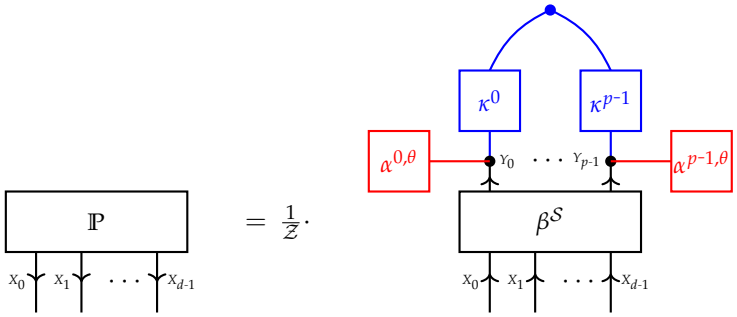


Figure 3.2: Tensor network decomposition of maximum entropy distributions to the constraint $\mu\,[L] = \left\langle \mathbb{P}, \sigma^{\mathcal{S}} \right\rangle [L]$. Blue: Constraint activation cores κ^l in a CP decomposition, representing the face measure to the minimal face, such that $\mu \in Q^{\mathcal{I}}_{\mathcal{S},\nu}$. Red: Probabilistic activation cores $\alpha^{l,\theta}\,[Y_l]$ in an elementary decomposition, where each leg core is a scaled exponentials evaluated on the enumerated image $\mathrm{im}\,(s_l)$.

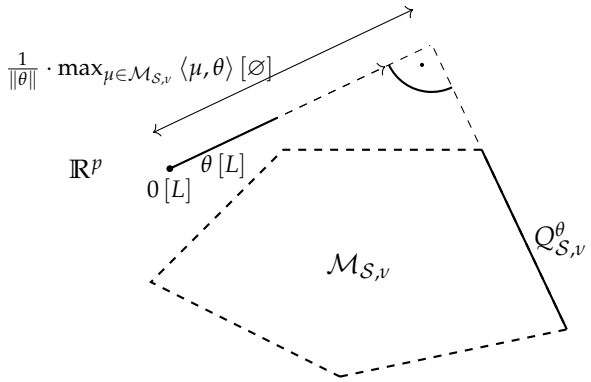


Figure 3.3: Sketch of a face $Q^{\theta}_{\mathcal{S},\nu}$ to the normal $\theta\,[L]$. The face is characterized by those mean parameters in $\mathcal{M}_{\mathcal{S},\nu}$, which maximize the contraction with $\theta\,[L]$. The sketched distance of the origin $0\,[\theta]$ of $\mathbb{R}^p$ to the affine hull of the face is further related to the maximal contraction.

the face measure of the unique face, which has the face normal by its canonical parameter.

3.6.1 Face Normals

Let us now investigate, that faces of $\mathcal{M}_{\mathcal{S},\nu}$ are the solutions of mode queries, which are linear optimization problems constrained by $\mathcal{M}_{\mathcal{S},\nu}$ (see Figure 3.3). Since such solution sets are intersections of the boundary of $\mathcal{M}_{\mathcal{S},\nu}$ with half-spaces, they are faces. In the next theorem we show, how linear optimization problems are constructed to match a given face.

Lemma 3.24 *For any non-empty face $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ to a subset $\mathcal{I} \subset [n]$ there is a vector $\theta\,[L]$, which we call a normal of the face, such that*

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

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 $Q^{\mathcal{I}}_{\mathcal{S},\nu}$.

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 $\mu \in \mathcal{M}_{\mathcal{S},\nu}$ obey

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

it follows that

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

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 $\mu\,[L] \in Q^{\mathcal{I}}_{\mathcal{S},\nu}$. ∎

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

Lemma 3.25 *For any* $\theta\,[L]$ *we find a face* $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ *such that*

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

We denote this face by $Q^{\theta}_{\mathcal{S},\nu} := Q^{\mathcal{I}}_{\mathcal{S},\nu}$.

Proof. We first notice, that

$$\begin{aligned}
&\operatorname{argmax}_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \langle \theta\,[L]\,, \mu\,[L] \rangle\,[\varnothing] \\
&\quad = \operatorname{conv}\left(\sigma^{\mathcal{S}} \left[X_{[d]} = x_{[d]}, L \right] \,:\, x_{[d]} \in \operatorname{argmax}_{x_{[d]} \in \times_{k \in [d]} [m_k]} \left\langle \theta\,[L]\,, \mathcal{S}\left(x_{[d]} \right) \right\rangle [\varnothing] \right)\,.
\end{aligned}$$

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

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

Based on Lem. 3.25 we can now characterize the solution of mode queries. To this end, we recall the statistic encoding from Sect. 2.4.1, which is defined as

$$\sigma^{\mathcal{S}} \,:\, \mathcal{X} \to \mathbb{R}^p \quad , \quad \sigma^{\mathcal{S}}\,(x) = \sigma^{\mathcal{S}} \left[X_{[d]} = x_{[d]}, L \right]\,.$$

Theorem 3.26 *For any* $\theta\,[L]$ *we have*

$$\operatorname{argmax}_{x_{[d]} \in \times_{k \in [d]} [m_k]} \left\langle \theta\,[L]\,, \sigma^{\mathcal{S}} \left[x_{[d]}, L \right] \right\rangle [\varnothing] = \left(\sigma^{\mathcal{S}} \right)^{-1} \left(Q^{\theta}_{\mathcal{S},\nu} \right)\,.$$

Any $x_{[d]} \in \bigtimes_{k \in [d]} [m_k]$ *is therefore a mode of* $\mathbb{P}^{\mathcal{S},\theta,\nu}$, *if and only if the face measure of* $Q^{\theta}_{\mathcal{S},\nu}$ *is supported on* $x_{[d]}$.

Proof. We have

$$\max_{x_{[d]} \in \bigtimes_{k\in[d]}[m_k]} \left\langle \theta\left[L\right], \sigma^{\mathcal{S}}\left[x_{[d]}, L\right]\right\rangle[\varnothing] = \max_{\mu[L]\in\mathcal{M}_{\mathcal{S},\nu}} \langle \theta\left[L\right], \mu\left[L\right]\rangle[\varnothing]$$

since $\mathcal{M}_{\mathcal{S},\nu}$ is the convex hull of the vectors $\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right]$. By Lem. 3.25 the maximum is taken at the face $Q^{\theta}_{\mathcal{S},\nu}$. We further have, since any face contains vertices, that $\gamma^{\mathcal{X}} \cap Q^{\theta}_{\mathcal{S},\nu} \neq \varnothing$. The solutions of the mode query are thus the states $x_{[d]}$, which statistics encoding is in the face $Q^{\mathcal{I}}_{\mathcal{S},\nu}$. This set is the pre-image $\left(\sigma^{\mathcal{S}}\right)^{-1}\left(Q^{\theta}_{\mathcal{S},\nu}\right)$.

From Def. 2.32 we have that $x_{[d]} \in \left(\sigma^{\mathcal{S}}\right)^{-1}\left(Q^{\theta}_{\mathcal{S},\nu}\right)$ if and only if the face measure of $Q^{\theta}_{\mathcal{S},\nu}$ is supported on $x_{[d]}$. ∎

Thm. 3.26 provides a geometric perspective for mode queries. An arbitrary positive tensor $\tau\left[X_{[d]}\right]$ can be understood to be a canonical parameter of the exponential family with statistic δ, and base measure $\mathbb{I}\left[X_{[d]}\right]$. For this exponential family, $\Lambda^{\delta,\mathrm{MAX},\mathbb{I}}$ coincides with the polytope of mean parameters, and is a standard simplex. Continuing our discussion in Sect. 3.1.2, we have for any $\mathcal{U} \subset \bigtimes_{k\in[d]}[m_k]$ and ν being the subset encoding of $\mathcal{U}$ that

$$\max_{x_{[d]}\in\mathcal{U}} \tau\left[X_{[d]} = x_{[d]}\right] = \max_{\mu\left[X_{[d]}\right]\in\Lambda^{\delta,\mathrm{MAX},\nu}} \left\langle \tau\left[X_{[d]}\right], \mu\left[X_{[d]}\right]\right\rangle[\varnothing] \ .$$

The maximum is attained exactly for the mean parameters

$$\mu\left[X_{[d]}\right] \in \mathrm{conv}\left(\epsilon_{x_{[d]}}\left[X_{[d]}\right] \ : \ x_{[d]} \in \mathrm{argmax}_{x_{[d]}\in\mathcal{U}}\tau\left[X_{[d]} = x_{[d]}\right]\right) \ .$$

Answering the mode query is thus the characterization of the face of the standard simplex with face normal $\tau\left[X_{[d]}\right]$.

Example 3.27 (Mode queries on tensor networks) Let us now consider cases where the queried tensor τ has a tensor network decomposition. This is the case for energy tensors to members of exponential families, for which we have a decomposition into selection encodings $\sigma^{\mathcal{S}}\left[X_{[d]}, L\right]$ and canonical parameters $\theta\left[L\right]$. In most generality we assume a decomposition of τ by a tensor network on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $[d] \subset \mathcal{V}$ as

$$\tau\left[X_{[d]}\right] = \langle\{\tau^e\left[X_e\right] \ : \ e \in \mathcal{E}\}\rangle\left[X_{[d]}\right] \ .$$

Let us choose a subset $\tilde{\mathcal{E}} \subset \mathcal{E}$ and

$$\max_{x_{[d]}\in\mathcal{U}} \tau\left[X_{[d]} = x_{[d]}\right] = \max_{x_{[d]}\in\mathcal{U}} \left\langle \{\epsilon_{x_{[d]}}\left[X_{[d]}\right]\} \cup \{\tau^e\left[X_e\right] \ : \ e \in \mathcal{E}\}\right\rangle[\varnothing]$$

We now split the contractions (see Thm. 17.1) to contract the cores $\mathcal{E}/\tilde{\mathcal{E}}$ with the one-hot encoding first and keeping $\tilde{\mathcal{V}} = \bigcup_{e\in\tilde{\mathcal{E}}} e$ open. With this we get

$$\begin{aligned}
&\max_{x_{[d]}\in\mathcal{U}} \tau\left[X_{[d]} = x_{[d]}\right] \\
&= \max_{x_{[d]}\in\mathcal{U}} \left\langle \left\langle \{\epsilon_{x_{[d]}}\left[X_{[d]}\right]\} \cup \left\{\tau^e\left[X_e\right] \ : \ e \in \mathcal{E}/\tilde{\mathcal{E}}\right\}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right], \left\langle \left\{\tau^e\left[X_e\right] \ : \ e \in \tilde{\mathcal{E}}\right\}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\right\rangle[\varnothing] \ .
\end{aligned}$$

This optimization problem is the characterization of vectors, which convex hull is the face in the

polytope

$$\mathcal{M} = \left\{ \{ \left\langle \epsilon_{x_{[d]}} \left[X_{[d]} \right] \right\} \cup \left\{ \tau^e \left[X_e \right] \,:\, e \in \mathcal{E} / \tilde{\mathcal{E}} \right\} \right\rangle \left[X_{\tilde{\mathcal{V}}} \right] \,:\, x_{[d]} \in \mathcal{U} \right\}$$

with the face normal

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

$\Diamond$

3.6.2 Cones of Constant Modes

We have observed, that the modes of maximum entropy distributions are characterized by the corresponding face measures. Let us now investigate in more detail the sets of maximum entropy distributions, which coincide in their modes.

Definition 3.28 To each face $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ the normal cone is the set

$$C^{\mathcal{I}} = \left\{ \theta \left[L \right] \,:\, \theta \left[L \right] \in \mathbb{R}^p, \operatorname{argmax}_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \left\langle \theta \left[L \right], \mu \left[L \right] \right\rangle \left[\varnothing \right] = Q^{\mathcal{I}}_{\mathcal{S},\nu} \right\} .$$

Each $C^{\mathcal{I}}$ is a convex cone, as we show now.

Lemma 3.29 *For each face $\mathcal{I}$, $C^{\mathcal{I}}$ is a convex cone.*

Proof. $C^{\mathcal{I}}$ is a cone, since for any $\theta \in C^{\mathcal{I}}$ and $\lambda > 0$ we have

$$Q^{\mathcal{I}}_{\mathcal{S},\nu} = \operatorname{argmax}_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \left\langle \theta, \mu \right\rangle \left[\varnothing \right] = \operatorname{argmax}_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \left\langle \lambda \cdot \theta, \mu \right\rangle \left[\varnothing \right] .$$

$C^{\mathcal{I}}$ is convex, since for $\theta, \tilde{\theta} \in C^{\mathcal{I}}$ and $\lambda \in (0,1)$ we have

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

It follows

$$Q^{\mathcal{I}}_{\mathcal{S},\nu} = \operatorname{argmax}_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \left\langle \lambda \cdot \theta + (1-\lambda) \cdot \tilde{\theta}, \mu \right\rangle \left[\varnothing \right]$$

and $\lambda \cdot \theta + (1-\lambda) \cdot \tilde{\theta} \in C^{\mathcal{I}}$. ∎

The faces $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ of the mean parameter polytope build a partially ordered set, which is called the face lattice (see [Zie13]). We now investigate a corresponding partial order on the max cones to the faces.

Lemma 3.30 *For two non-empty faces $\mathcal{I}^0$ and $\mathcal{I}^1$ we have*

$$Q^{\mathcal{I}^0}_{\mathcal{S},\nu} \subset Q^{\mathcal{I}^1}_{\mathcal{S},\nu}$$

if and only if

$$C^{\mathcal{I}^1}_{\mathcal{S},\nu} \subset \overline{C^{\mathcal{I}^0}_{\mathcal{S},\nu}} .$$

Proof. "⇒"Let us assume $Q^{\mathcal{I}^0}_{\mathcal{S},\nu} \subset Q^{\mathcal{I}^1}_{\mathcal{S},\nu}$ and let us choose arbitrary $\theta_0 \in C^{\mathcal{I}^0}_{\mathcal{S},\nu}$, $\theta_1 \in C^{\mathcal{I}^1}_{\mathcal{S},\nu}$. It suffices to show, that for all $\lambda \in (0,1)$ we have $\lambda \cdot \theta_0 + (1-\lambda) \cdot \theta_1 \in C^{\mathcal{I}^0}_{\mathcal{S},\nu}$, since this implies $\theta_1 \in \overline{C^{\mathcal{I}^0}_{\mathcal{S},\nu}}$. By

assumption we have

$$\operatorname{argmax}_{\mu\in\mathcal{M}_{\mathcal{S},\nu}} \langle\theta_0,\mu\rangle\,[\varnothing] \subset \operatorname{argmax}_{\mu\in\mathcal{M}_{\mathcal{S},\nu}} \langle\theta_1,\mu\rangle\,[\varnothing]$$

and thus

$$\operatorname{argmax}_{\mu\in\mathcal{M}_{\mathcal{S},\nu}} \langle\lambda\cdot\theta_0+(1-\lambda)\cdot\theta_1,\mu\rangle\,[\varnothing] = \operatorname{argmax}_{\mu\in\mathcal{M}_{\mathcal{S},\nu}} \langle\theta_0,\mu\rangle\,[\varnothing]$$

which implies $\lambda\cdot\theta_0+(1-\lambda)\cdot\theta_1 \in C^{\mathcal{I}^0}_{\mathcal{S},\nu}$. "⇐"Conversely, let us assume $C^{\mathcal{I}^1}_{\mathcal{S},\nu} \subset \overline{C^{\mathcal{I}^0}_{\mathcal{S},\nu}}$. We then find a sequence $(\theta_n)_{n\in\mathbb{N}}$ with $\theta_n \in C^{\mathcal{I}^0}_{\mathcal{S},\nu}$ for $n\in\mathbb{N}$ and which limit θ exists in $C^{\mathcal{I}^1}_{\mathcal{S},\nu}$. For an arbitrary $x\in\mathcal{X}$ with $\sigma^{\mathcal{S}}(x) \in Q^{\mathcal{I}^0}_{\mathcal{S},\nu}$ we have

$$\begin{aligned}\left\langle\theta,\sigma^{\mathcal{S}}(x)\right\rangle[\varnothing] &= \lim_{n\to\infty}\left\langle\theta_n,\sigma^{\mathcal{S}}(x)\right\rangle[\varnothing]\\ &= \lim_{n\to\infty}\max_{y\in\mathcal{X}}\left\langle\theta_n,\sigma^{\mathcal{S}}(y)\right\rangle[\varnothing]\\ &= \max_{y\in\mathcal{X}}\left\langle\theta,\sigma^{\mathcal{S}}(y)\right\rangle[\varnothing]\end{aligned}$$

and therefore $\sigma^{\mathcal{S}}(x) \in Q^{\mathcal{I}^1}_{\mathcal{S},\nu}$. Note, that we used in the third equation, that $\mathcal{X}$ is finite.We use this property for all elements of the preimage of $\mathcal{I}^0$ and get

$$\begin{aligned}Q^{\mathcal{I}^0}_{\mathcal{S},\nu} &= \operatorname{conv}\left(\sigma^{\mathcal{S}}(x)\,:\,x\in\operatorname{argmax}_{y\in\mathcal{X}}\left\langle\theta_1,\sigma^{\mathcal{S}}(y)\right\rangle[\varnothing]\right)\\ &\subset \operatorname{conv}\left(\sigma^{\mathcal{S}}(x)\,:\,x\in\operatorname{argmax}_{y\in\mathcal{X}}\left\langle\theta,\sigma^{\mathcal{S}}(y)\right\rangle[\varnothing]\right)\\ &= Q^{\mathcal{I}^1}_{\mathcal{S},\nu}.\end{aligned}$$

■

Lem. 3.30 suggests that the partial order of faces by inclusion is mimicked by another partial order on the max cones, sketched in Figure 3.4. We now consider the set of cones with this partial order, and show that this set is homomorphic to the face lattice.

Theorem 3.31 *The max cone lattice of $\mathcal{M}_{\mathcal{S},\nu}$, partially order by the*

$$C^{\mathcal{I}^0}_{\mathcal{S},\nu} \prec C^{\mathcal{I}^1}_{\mathcal{S},\nu} \quad \textit{if and only if} \quad C^{\mathcal{I}^1}_{\mathcal{S},\nu} \subset \overline{C^{\mathcal{I}^0}_{\mathcal{S},\nu}}$$

is homomorphic to the face lattice of $\mathcal{M}_{\mathcal{S},\nu}$ with the homomorphism

$$\psi(Q^{\mathcal{I}}_{\mathcal{S},\nu}) = C^{\mathcal{I}}_{\mathcal{S},\nu}.$$

Proof. We have to show that for all pairs of faces $Q^{\mathcal{I}^0}_{\mathcal{S},\nu}, Q^{\mathcal{I}^1}_{\mathcal{S},\nu}$

$$\psi(Q^{\mathcal{I}^0}_{\mathcal{S},\nu}) \prec \psi(Q^{\mathcal{I}^1}_{\mathcal{S},\nu}) \quad\Leftrightarrow\quad Q^{\mathcal{I}^0}_{\mathcal{S},\nu} \prec Q^{\mathcal{I}^1}_{\mathcal{S},\nu}.$$

We show this first for the case, that $Q^{\mathcal{I}^0}_{\mathcal{S},\nu} = \varnothing$ or $Q^{\mathcal{I}^1}_{\mathcal{S},\nu} = \varnothing$. Note, that the empty face is contained in any other face, but contains no non-empty face. Conversely, the trivial max cone $C^{\mathcal{I}^\varnothing}_{\mathcal{S},\nu}$ is for minimal statistics $\{0\,[L]\}$ and contained in the boundary of any other non-empty cone. If the statistic is not minimal, the inclusion holds for the equivalence class of canonical parameters. Further, since the max cones are a disjoint partition of $\mathbb{R}^p$, $0\,[L]$ is not in any other max cone and thus $C^{\mathcal{I}}_{\mathcal{S},\nu} \prec C^{\mathcal{I}^\varnothing}_{\mathcal{S},\nu}$ does not hold for any non-empty $\mathcal{I}$.

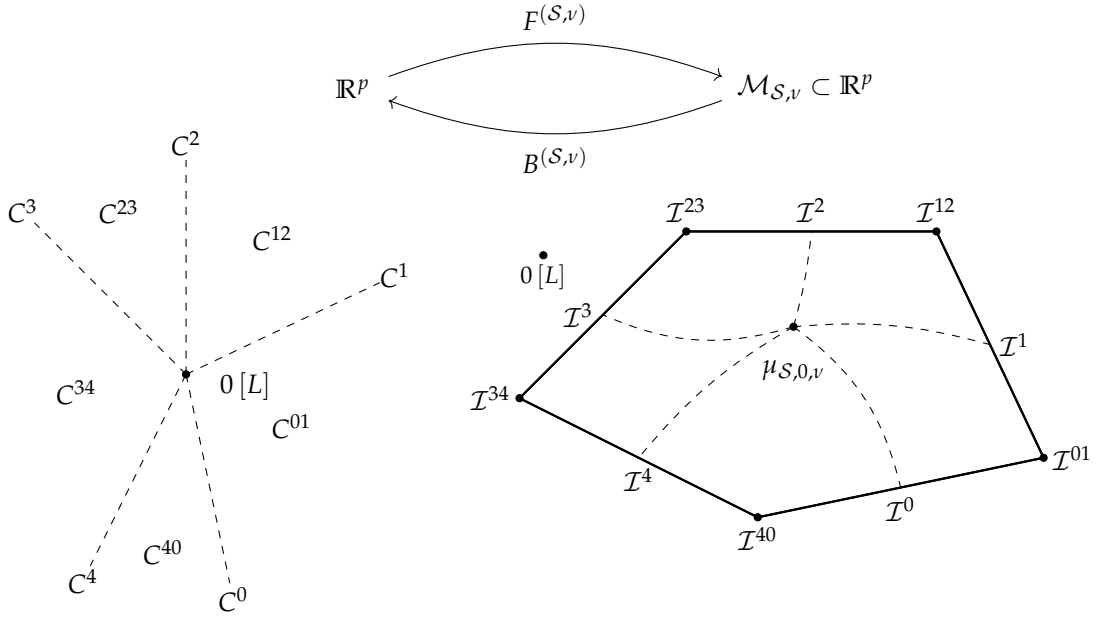


Figure 3.4: Partition of the canonical parameters in $\mathbb{R}^p$ into convex maximum cones $C^{\mathcal{I}}$ to faces $\mathcal{I}$ (left). The forward mapping maps the maximum cones into the mean parameter polytope (right).

For all pairs of non-empty faces $Q^{\mathcal{I}^0}_{\mathcal{S},\nu}, Q^{\mathcal{I}^1}_{\mathcal{S},\nu}$, Lem. 3.30 ensures that

$$\psi(Q^{\mathcal{I}^0}_{\mathcal{S},\nu}) \prec \psi(Q^{\mathcal{I}^1}_{\mathcal{S},\nu}) \quad \Leftrightarrow \quad Q^{\mathcal{I}^0}_{\mathcal{S},\nu} \prec Q^{\mathcal{I}^1}_{\mathcal{S},\nu}\,. \qquad \blacksquare$$

As a consequence of Thm. 3.31, the max cone lattice inherits all properties of the face lattice, for example those shown in Theorem 2.6 in [Zie13].

3.7 Mean Field Methods

Mean field methods are approximation schemes for forward mappings, designed for efficient inference. To introduce them we turn the maximization over the mean parameter polytope in the variational principle of forwards mappings into a maximization over the reproducing distributions, that is

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

where

$$\phi\left[X_{[d]}\right] = \left\langle \sigma^{\mathcal{S}}, \theta \right\rangle \left[X_{[d]}\right].$$

Mean field methods now provide lower bounds on this maximization by restricting the distribution optimized over to a tractable subset of distributions.

Let Γ be a subset of $\Lambda^{\delta,\mathrm{MAX},\mathbb{I}}$, we state the mean field problem as

$$\mathrm{argmax}_{\mathbb{P} \in \Gamma} \langle \phi, \mathbb{P} \rangle\,[\varnothing] + \mathbb{H}\,[\mathbb{P}] \qquad (\mathrm{P}^I_{\Gamma,\mathbb{P}^\phi})$$

We show next, that the mean field problem is an instance of an information projection, as indicated by the notation.

Theorem 3.32 *Let there be an energy tensor ϕ and consider the distribution*

$$\mathbb{P}^{\phi}\left[X_{[d]}\right] = \left\langle \exp\left[\phi\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right] .$$

For any set Γ of distributions we have

$$\operatorname{argmax}_{\mathbb{P}\in\Gamma}\langle\phi,\mathbb{P}\rangle[\varnothing] + \mathbb{H}[\mathbb{P}] = \operatorname{argmin}_{\mathbb{P}\in\Gamma} \mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}||\mathbb{P}^{\phi}\right] .$$

Problem ($\mathrm{P}^{I}_{\Gamma,\mathbb{P}^{\phi}}$) is thus the information projection of a $\mathbb{P}^{\phi}\left[X_{[d]}\right]$ onto Γ.

Proof. The cross entropy between a $\mathbb{P}\in\Gamma$ and $\mathbb{P}^{\phi}$ is

$$\begin{aligned}\mathbb{H}\left[\mathbb{P},\mathbb{P}^{\phi}\right] &= \left\langle \mathbb{P}\left[X_{[d]}\right], -\ln\left[\mathbb{P}^{\phi}\left[X_{[d]}\right]\right]\right\rangle[\varnothing] \\ &= \left\langle \mathbb{P}\left[X_{[d]}\right], -\phi\left[X_{[d]}\right]\right\rangle[\varnothing] + \left\langle \mathbb{P}\left[X_{[d]}\right]\right\rangle[\varnothing]\cdot\ln\left[\left\langle\exp\left[\phi\left[X_{[d]}\right]\right]\right\rangle[\varnothing]\right]\end{aligned}$$

Together we have, that

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}||\mathbb{P}^{\phi}\right] &= \mathbb{H}\left[\mathbb{P},\mathbb{P}^{\phi}\right] - \mathbb{H}[\mathbb{P}] \\ &= -\left\langle \mathbb{P}\left[X_{[d]}\right], \phi\left[X_{[d]}\right]\right\rangle[\varnothing] - \mathbb{H}[\mathbb{P}] + \ln\left[\left\langle\exp\left[\phi\left[X_{[d]}\right]\right]\right\rangle[\varnothing]\right]\end{aligned}$$

Since the last term is constant among $\mathbb{P}\in\Gamma$, it holds that

$$\operatorname{argmax}_{\mathbb{P}\in\Gamma}\langle\phi,\mathbb{P}\rangle[\varnothing] + \mathbb{H}[\mathbb{P}] = \operatorname{argmax}_{\mathbb{P}\in\Gamma} - \mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}||\mathbb{P}^{\phi}\right] = \operatorname{argmax}_{\mathbb{P}\in\Gamma}\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}||\mathbb{P}^{\phi}\right] . \qquad \blacksquare$$

3.7.1 Naive Mean Field Method

In the naive mean field method, we choose the approximating set Γ by the exponential family of Markov Networks $\Gamma^{\mathcal{S}^{\mathrm{EL}},\mathbb{I}}$ on the elementary graph

$$\mathrm{EL} = \left([d], \{\{k\} : k\in[d]\}\right) .$$

Markov Networks on this graph are represented by normalized leg cores

$$\mathbb{P}\left[X_{[d]}\right] = \bigotimes_{k\in[d]} \rho^{k}\left[X_{k}\right]$$

Problem ($\mathrm{P}^{I}_{\Gamma,\mathbb{P}^{\phi}}$) is for this instance of the form

$$\operatorname{argmax}_{\rho^{k}[X_{k}]\,:\,\langle\rho^{k}\rangle[\varnothing]=1,\,k\in[d]} \left\langle \{\phi\}\cup\{\rho^{k} : k\in[d]\}\right\rangle[\varnothing] + \sum_{k\in[d]}\mathbb{H}\left[\rho^{k}\right] .$$

We approximately solve this problem in Algorithm 3 by alternation through the leg cores and performing a locally optimal leg core update. The optimal update equations are derived as the next theorem.

Theorem 3.33 (Update equations for the naive mean field approximation) *For any $k\in[d]$ and*

leg cores $\{\rho^{\tilde{k}}\left[X_{\tilde{k}}\right] : \tilde{k} \in [d], \tilde{k} \neq k\}$ *the local problem*

$$\operatorname{argmax}_{\rho^k[X_k]:\left\langle\rho^k\right\rangle[\varnothing]=1} \left\langle \{\phi\} \cup \{\rho^{\tilde{k}}\left[X_{\tilde{k}}\right] : \tilde{k} \in [d]\} \right\rangle [\varnothing] + \sum_{k\in[d]} \mathbb{H}\left[\rho^k\right]$$

is solved at

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

Proof. We have

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

and by multilinearity of tensor contractions

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

Combining both, the condition

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

is equal to

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

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

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

■

Algorithm 3 is the alternation of legwise updates until a stopping criterion is met.

3.7.2 Structured Variational Approximation

We now generalize the naive mean field method towards generic families of Markov Networks. Let $\mathcal{G}$ be any hypergraph, the structured variational approximation method is Problem ($\mathrm{P}^I_{\Gamma,\mathbb{P}^\phi}$) with

$$\Gamma = \Gamma^{\mathcal{S}^{\mathcal{G}},\mathbb{I}} .$$

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

Theorem 3.34 (Update equations for the structured variational approximation) *The Markov Network* $\tau^{\mathcal{G}}$ *with hypercores* $\{\tau^e[X_e] : e \in \mathcal{E}\}$ *is a stationary point for structured variational*

Algorithm 3 Naive Mean Field Approximation

Input: Energy tensor $\phi\left[X_{[d]}\right]$

Output: Tensor Network $\{\rho^k\left[X_k\right] : k \in [d]\}$ approximating $\left\langle \exp\left[\phi\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right]$

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

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

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

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

  end for
end while
return $\{\rho^k\left[X_k\right] : k \in [d]\}$

approximation, if for all $e \in \mathcal{E}$ we find a $\lambda > 0$ with

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

Here, the quotient denotes the coordinatewise quotient.

Proof. We proof the theorem by the first order condition on the objective

$$O\left(\tau^{\mathcal{G}}\right) = \left\langle \phi, \left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]\right\rangle[\varnothing] + \mathbb{H}\left[\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]\right]$$

We further use Lem. 13.19, which shows a characterization of the derivative of functions dependent on tensors.

We have

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

Further we have

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

We define the tensor

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

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

The objective has then a representation as

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

Let us now differentiate all terms. With Lem. 13.19 we now get

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

Further we have

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

and (see Proof of 13.18)

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

Together, the first order condition

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

is equal to all y_e satisfying

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

We notice that by normalization

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

and that the scalar

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

is the constant for all y_e.

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

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

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

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

and reparametrization of λ_1 to

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

■

3.8 Backward Mapping in Exponential Families

Let us now continue with the discussion of the backward mapping, which calculates to a mean parameter $\mu\left[L\right]$ a canonical parameter $\theta\left[L\right]$, such that the corresponding member of the exponential family reproduced $\mu\left[L\right]$.

3.8.1 Variational Formulation

We now provide a variational characterization of the backward mapping.

Theorem 3.35 *Let there be a statistic $\mathcal{S}$, which is minimal with respect to a boolean base measure ν. The map $B^{(\mathcal{S},\nu)} : (\mathcal{M}_{\mathcal{S},\nu})^\circ \to \mathbb{R}^p$ defined as*

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

is a backward mapping.

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

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

and thus

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

The first order condition is therefore

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

and any θ satisfies this condition exactly when $\theta = B^{(\mathcal{S},\nu)}(\mu)$ for a backward mapping. We conclude the proof by noticing, that since $\mu\left[L\right]$ is in the interior $(\mathcal{M}_{\mathcal{S},\nu})^\circ$ we find by Thm. 3.12 a θ such that the first order condition is met. ■

3.8.2 Interpretation as a Moment Projection

Backward mappings coincide with the Maximum Likelihood Estimation Problem ($\mathrm{P}^{\mathrm{M}}_{\Gamma,\mathbb{P}^D}$), when we take the hypothesis to be exponential family $\Gamma^{\mathcal{S},\nu}$. We show this as a more general statement for moment projections onto exponential families.

Theorem 3.36 *Let there be any exponential family, a mean parameter vector* $\mu^* \in \operatorname{im}\left(F^{(\mathcal{S},\nu)}\right)$ *and a backward mapping* $B^{(\mathcal{S},\nu)}$*. Then* $\hat{\theta} = B^{(\mathcal{S},\nu)}(\mu^*)$ *is the canonical parameter to the solution of the moment projection Problem* $(\mathrm{P}^{\mathrm{M}}_{\Gamma,\mathbb{P}^*})$ *of any* $\mathbb{P}^*$ *with*

$$\left\langle \sigma^{\mathcal{S}}, \mathbb{P}^* \right\rangle [L] = \mu^*[L]$$

onto the exponential family, if

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

Proof. We exploit the variational characterization of the backward mapping by Thm. 3.35, 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 3.10

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

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

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

This shows, that the backward mapping coincides with the moment projection onto $\Gamma = \Gamma^{\mathcal{S},\nu}$. ∎

In particular, if we choose $\mu[L] = \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \mathbb{P}^D \right\rangle [L]$ for an empirical distribution $\mathbb{P}^D$, the backward mapping is a maximum likelihood estimator. This holds, since by Lem. 3.9 maximum likelihood estimation is a special instance of moment projection, when projecting $\mathbb{P}^D$. In that case, we choose $\mu[L] = \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \mathbb{P}^D \right\rangle [L]$.

3.8.3 Approximation by Alternating Algorithms

While the forward mapping always has a representation in closed form by contraction of the probability tensor, the backward mapping in general fails to have a closed form representation. Computation of the backward mapping 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 $l \in [p]$

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

This condition is equal to the collection of moment matching equations

$$\left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}, \sigma^{\mathcal{S}} \right\rangle [L = l] = \left\langle \mathbb{P}^D, \sigma^{\mathcal{S}} \right\rangle [L = l] .$$

Lemma 3.37 *For any sufficient statistic* $\mathcal{S}$ *a parameter vector* θ *and an index* $l \in [p]$ *we define*

$$\tau[Y_l] = \left\langle \{\beta^{\mathcal{S}}\left[Y_{[p]}, X_{[d]}\right]\} \cup \{\alpha^{\tilde{l},\theta}[Y_{\tilde{l}}] \,:\, \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\left[X_{[d]}\right]\} \right\rangle [Y_l] .$$

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

$$\left\langle \alpha^{l,\theta}\left[Y_l\right], I_l\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[\varnothing\right] = \left\langle \alpha^{l,\theta}\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[\varnothing\right] \cdot \mu\left[L=l\right] .$$

Proof. We have

$$\mathbb{P}^{(\mathcal{S},\theta,\nu)} = \frac{\left\langle \alpha^{l,\theta}\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[X_{[d]}\right]}{\left\langle \alpha^{l,\theta}\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[\varnothing\right]}$$

and

$$\left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}, s_l\right\rangle\left[\varnothing\right] = \frac{\left\langle \alpha^{l,\theta}\left[Y_l\right], I_l\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[X_{[d]}\right]}{\left\langle \alpha^{l,\theta}\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[\varnothing\right]} .$$

Here we used (see Cor. 14.14)

$$s_l = \left\langle \beta^{s_l}\left[Y_l, X_{[d]}\right], I_l\left[Y_l\right]\right\rangle\left[X_{[d]}\right]$$

and redundancies of copies of basis encodings. It follows that

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

is equal to

$$\left\langle \alpha^{l,\theta}\left[Y_l\right], I_l\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[\varnothing\right] = \left\langle \alpha^{l,\theta}\left[Y_l\right], \tau\left[Y_l\right]\right\rangle\left[\varnothing\right] \cdot \mu\left[L=l\right] . \qquad \blacksquare$$

The steps have to be alternated until sufficient convergence, since matching the moment to l by modifying $\theta\left[L=l\right]$ 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 4. Since the log likelihood is concave, the algorithm converges to a global minimum.

In general, if $\mathrm{im}\left(s_l\right)$ 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. 9.3.

The computation of τ^l in Algorithm 4 can be intractable and be replaced by an approximative procedure based on message passing schemes.

3.8.4 Second Order Methods

The Hesse matrix of $A^{(\mathcal{S},\nu)}$ at θ is the covariance of the features with respect to $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$, as we show next.

Lemma 3.38 *At any $\theta \in \mathbb{R}^p$ we have*

$$\begin{aligned}\nabla_{\tilde{\theta}\left[\tilde{L}\right]}\nabla_{\tilde{\theta}[L]}|_{\theta}A^{(\mathcal{S},\nu)}(\tilde{\theta}) = &\left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \sigma^{\mathcal{S}}\left[X_{[d]}, \tilde{L}\right]\right\rangle\left[\tilde{L}, L\right] \\ &- \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, \tilde{L}\right]\right\rangle\left[\tilde{L}\right] \otimes \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle\left[L\right] .\end{aligned}$$

Proof. By Lem. 3.13 we have

$$\nabla_{\tilde{\theta}[L]}|_{\theta}A^{(\mathcal{S},\nu)}(\tilde{\theta}) = \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle\left[L\right] .$$

Algorithm 4 Alternating Moment Matching for the Backward Mapping

Input: Empirical distribution $\mathbb{P}^D$, statistic $\mathcal{S}$ and base measure ν

Output: Canonical parameter $\theta\left[L\right]$, such that $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$ is the (approximative) moment projection of $\mathbb{P}^D$ onto $\Gamma^{\mathcal{S},\nu}$

Set $\theta\left[L\right] = 0\left[L\right]$
Compute $\mu_D\left[L\right] = \left\langle \mathbb{P}^D, \sigma^{\mathcal{S}} \right\rangle\left[L\right]$
while Convergence criterion not met **do**
  for all $l \in [p]$ **do**
    Compute

$$\tau^l\left[Y_l\right] \leftarrow \left\langle \{\beta^{\mathcal{S}}\} \cup \{\alpha^{\tilde{l},\theta[L=\tilde{l}]} \,:\, \tilde{l} \in [p],\, \tilde{l} \neq l\} \right\rangle\left[Y_l\right]$$

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

$$\left\langle \alpha^{l,\theta}\left[Y_l\right], I_l\left[Y_l\right], \tau^l \right\rangle\left[\varnothing\right] = \left\langle \alpha^{l,\theta}\left[Y_l\right], \tau^l \right\rangle\left[\varnothing\right] \cdot \mu_D\left[L = l\right] .$$

  end for
end while
return $\theta\left[L\right]$

It further holds

$$\begin{aligned}\nabla_{\tilde{\theta}\left[\tilde{L}\right]}|_{\theta}\mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right] = &\left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, \tilde{L}\right] \right\rangle\left[X_{[d]}, \tilde{L}\right] \\ &- \left\langle \mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right], \sigma^{\mathcal{S}}\left[X_{[d]}, \tilde{L}\right] \right\rangle\left[\tilde{L}\right] \otimes \mathbb{P}^{(\mathcal{S},\theta,\nu)}\left[X_{[d]}\right] .\end{aligned}$$

Combining both equations, we get the claim. ∎

With this characterization, we can perform second order optimization algorithms such as the Newton Method, see Algorithm 5 to solve the backward mapping.

3.9 Discussion

The forward and backward mapping have a correspondence with each other in terms of convex duality [Roc97]. As such, the cumulant function $A^{(\mathcal{S},\nu)}$ (see Def. 2.21) and its conjugate dual $\left(A^{(\mathcal{S},\nu)}\right)^*$ represent the forward and backward mapping by their gradient.

Further approximation schemes arise for the forward and backward mapping in exponential families arise from loopy message passing algorithms. For inference on Markov Network families, they are derived based on outer bounds on the mean polytope in terms of the local consistency polytope and approximations on the entropy term by local terms (see Chapter 3 in [WJ08]). In particular, different approximations of the polytope and the entropies are made in the Bethe and Kickuchi method. We will discuss message passing schemes in Chapter 17 with a focus on exact computation of generic contractions by local contractions.

Algorithm 5 Newton Method for the Backward Mapping

Input: Empirical distribution $\mathbb{P}^D$, statistic $\mathcal{S}$ and base measure ν

Output: Canonical parameter $\theta\,[L]$, such that $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$ is the (approximative) moment projection of $\mathbb{P}^D$ onto $\Gamma^{\mathcal{S},\nu}$

Set $\theta\,[L] = 0\,[L]$
Compute $\mu_D\,[L] = \left\langle \mathbb{P}^D, \sigma^{\mathcal{S}} \right\rangle [L]$
while Convergence criterion not met **do**
Calculate $\nabla_{\tilde{\theta}[\tilde{L}]} \nabla_{\tilde{\theta}[L]}|_\theta A^{(\mathcal{S},\nu)}(\tilde{\theta})$ and $\nabla_{\tilde{\theta}[L]}|_\theta A^{(\mathcal{S},\nu)}(\tilde{\theta})$ as in Lem. 3.38.
Solve the linear equation

$$\left(\nabla_{\tilde{\theta}[\tilde{L}]} \nabla_{\tilde{\theta}[L]}|_\theta A^{(\mathcal{S},\nu)}(\tilde{\theta})\right) \Delta\,[L] = \mu\,[L] - \nabla_{\tilde{\theta}[L]}|_\theta A^{(\mathcal{S},\nu)}(\tilde{\theta})$$

Update the canonical parameter

$$\theta\,[L] \leftarrow \theta\,[L] - \Delta\,[L]$$

end while
return $\theta\,[L]$

4

Propositional Logic

Propositional logics describes systems in factored representations with d boolean variables, which are called atoms and denoted by f^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 $x_{[d]} = x_0, \dots, x_{d-1}$ the indices f^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. Such boolean tensors have already appeared as base measures in the representation of probability distributions in Chapter 2.

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

4.1 Encoding of Booleans

Boolean variables are a basic data structure, valued in $\{\text{False}, \text{True}\}$.

4.1.1 Representation by Coordinates

To represent booleans by categorical variables X with two states we use the index interpretation function (see Def. 0.21 and for more details Def. 14.1 in Part III)

$$I : [2] \to \{\text{False}, \text{True}\}$$

defined as

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

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 4.1 *I is a homomorphism between the groups*

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

Proof. It suffices to notice, that for arbitrary $z_0, z_1 \in \{0,1\}$ we have

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

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

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

$$I^{-1}(\neg z) = 1 - I^{-1}(z)\,. \tag{4.1}$$

Since any logical connective can be represented as a composition of conjunctions and negations, any logical connective corresponds with an affine linear function on the interpreted truth values. Direct applications of this insight to execute logical calculus will be discussed later in Sect. 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 (4.1) the affine representation problem, which we aim to resolve in the following.

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

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

defined as

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

results is a homomorphism between the groups

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

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

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

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

4.1.2 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 ϵ and get an encoding

$$\epsilon \circ I^{-1} : \{\text{False}, \text{True}\} \to \{\epsilon_0\,[X]\,, \epsilon_1\,[X]\}\,,$$

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

$$\epsilon \circ I^{-1}(z) = \begin{bmatrix} I^{-1}(\neg z) \\ I^{-1}(z) \end{bmatrix}.$$

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

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

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

$$\epsilon \circ I^{-1}(\neg z)[Y_\neg] = \left\langle \beta^\neg\,[Y_\neg, X]\,, \epsilon \circ I^{-1}(z)[X] \right\rangle [Y_\neg]\,.$$

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.

4.1.3 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 $\{\epsilon_0[X], \epsilon_1[X]\}$. 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 Thm. 14.10.

4.2 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 basis encoding of formulas.

4.2.1 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 4.2 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.

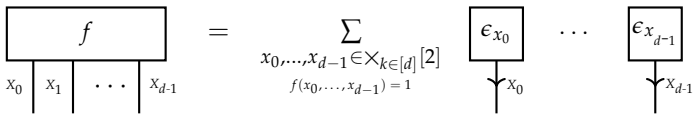
The propositional formulas coincide therefore with the boolean tensors (see Def. 0.5).

Since propositional formulas are binary valued tensors, the generic decomposition of Lem. 13.1 simplifies to

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

Thus, any propositional formula is the sum over the one-hot encodings of its models. This is equal to the encoding of the set of models, which will be introduced in Chapter 14 (see Def. 14.1).

We depict this decomposition in the diagrammatic notation by



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

4.2.2 Representation by Computation-Activation Networks

We apply the coordinate and the basis encoding schemes to represent propositional formulas as tensors. In the coordinate encoding scheme, which we use by default, we understand $[2]$ as subset of $\mathbb{R}$ and interpreting the formula directly as a tensor (as in Def. 4.2). To apply the basis encoding scheme, we understand $[2]$ as the states of a system with factored representation by a single categorical variable. Although it comes with an additional boolean variable compared with coordinate encoding scheme, we in this work use the basis encoding scheme to find sparse tensor network decompositions based on syntactical formula decompositions. Furthermore, the basis encoding connects with the formalism of Computation-Activation Networks developed in Chapter 2, as we show next.

Theorem 4.3 (Formula representation as Computation-Activation Network) *For each formula f we have*

$$f\left[X_{[d]}\right] = \left\langle \beta^f\left[Y_f, X_{[d]}\right], \epsilon_1\left[Y_f\right]\right\rangle\left[X_{[d]}\right]$$

and

$$\neg f\left[X_{[d]}\right] = \left\langle \beta^f\left[Y_f, X_{[d]}\right], \epsilon_0\left[Y_f\right]\right\rangle\left[X_{[d]}\right],$$

which are Computation-Activation Networks to the statistic $\{f\}$ and the hard activation tensor $\epsilon_1\left[Y_f\right]$, respectively $\epsilon_0\left[Y_f\right]$.

Proof. Given a factored system with d atoms $X_{[d]}$ and a propositional formula f, the basis encoding of f (see Def. 0.20 and Figure 4.1) is the tensor

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

We reorder the summation into

$$\begin{aligned}\beta^f\left[Y_f, X_{[d]}\right] =& \epsilon_0\left[Y_f\right] \otimes \left(\sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=0} \epsilon_{x_{[d]}}\left[X_{[d]}\right]\right) \\ &+ \epsilon_1\left[Y_f\right] \otimes \left(\sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=1} \epsilon_{x_{[d]}}\left[X_{[d]}\right]\right)\end{aligned}$$

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

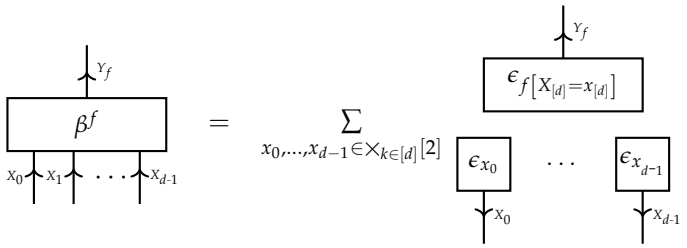


Figure 4.1: Basis encoding of a propositional formula. The encoding is a sum of the one hot encodings of all states of the factored system in a tensor product with basis vectors, which encode whether the state is a model of the formula. The tensor is directed, since any contraction with an encoded state results in the basis vector evaluating the formula, which we called basis calculus.

of contractions we get

$$\begin{aligned}\left\langle \beta^f \left[Y_f, X_{[d]}\right], \epsilon_1\left[Y_f\right]\right\rangle \left[X_{[d]}\right] &= \left\langle \epsilon_0\left[Y_f\right], \epsilon_1\left[Y_f\right]\right\rangle [\varnothing] \cdot \left(\sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=0} \epsilon_{x_{[d]}}\left[X_{[d]}\right]\right) \\ &\quad + \left\langle \epsilon_1\left[Y_f\right], \epsilon_1\left[Y_f\right]\right\rangle [\varnothing] \cdot \left(\sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=1} \epsilon_{x_{[d]}}\left[X_{[d]}\right]\right) \\ = \sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=1} \epsilon_{x_{[d]}}\left[X_{[d]}\right] &= f\left[X_{[d]}\right].\end{aligned}$$

Analogously we have

$$\left\langle \beta^f \left[Y_f, X_{[d]}\right], \epsilon_0\left[Y_f\right]\right\rangle \left[X_{[d]}\right] = \sum_{x_{[d]}\,:\, f\left[x_{[d]}\right]=0} \epsilon_{x_{[d]}}\left[X_{[d]}\right] = \neg f\left[X_{[d]}\right] \blacksquare$$

Thm. 4.3 shows that besides the formula itself, we can represent the negation of the formula as a Computation-Activation Network with the same statistic $\{f\}$. In general, any boolean tensor which has a representation as a Computation-Activation Network needs to have a boolean activation tensor. In the case of the single-dimensional statistic $\{f\}$ we have three boolean activation vectors $\{\mathbb{I}\left[Y_f\right], \epsilon_1\left[Y_f\right], \epsilon_0\left[Y_f\right], 0\left[Y_f\right]\}$. While the Computation-Activation Networks with the activation cores $\epsilon_1\left[Y_f\right], \epsilon_0\left[Y_f\right]$ are by Thm. 4.3 the formula and its negation, the activation core $\mathbb{I}\left[Y_f\right]$ and $0\left[Y_f\right]$ represent the trivial $\mathbb{I}\left[X_{[d]}\right]$ and the vanishing tensor $0\left[X_{[d]}\right]$.

The contraction of the basis encoding with one-hot encodings to worlds $x_{[d]} \in \times_{k\in[d]}[2]$ is

$$\left\langle \beta^f \left[Y_f, X_{[d]}\right], \epsilon_{x_{[d]}}\left[X_{[d]}\right]\right\rangle \left[Y_f\right] = \begin{cases} \epsilon_1\left[Y_f\right] & \text{if the world } x_{[d]} \text{ is a model of } f \\ \epsilon_0 Y_f & \text{else.}\end{cases}$$

We thus interpret the contractions as calculating in basis vectors, whether the world $x_{[d]}$ is a model of the formula f. This is the basis calculus scheme and will be investigated it in more detail in Chapter 14 (see Thm. 14.11).

4.3 Syntax of Propositional Formulas

Basis encodings of propositional formulas are especially useful when representing function compositions by the representation of their components (see Thm. 14.12). In propositional logics, the syntax of defining propositional formulas is oriented on compositions of formulas by connectives. In this section, we investigate the decomposition schemes of basis encodings into tensor networks of component encodings for binary tensors following propositional logic syntax.

4.3.1 Atomic Formulas

We call atomic formulas the most granular formulas, which are not split into compositions of other formulas. Our syntactic decomposition of propositional formulas will then investigate, how any propositional formula can be represented by these.

Definition 4.4 The tensors $f_k\left[X_{[d]}\right]$ defined for $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ as

$$f_k\left[X_{[d]} = x_{[d]}\right] = x_k$$

are called atomic formulas.

Atomic formulas and their basis encodings have an especially compelling representation.

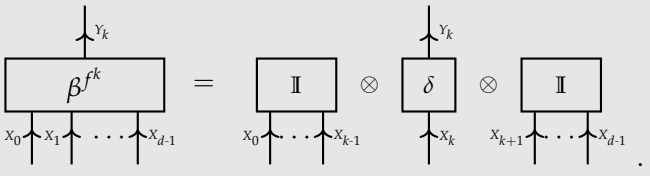
Lemma 4.5 *Any atomic formula* $f^k\left[X_{[d]}\right]$ *is represented as*

$$f^k\left[X_{[d]} = x_{[d]}\right] = \langle \epsilon_1\left[X_k\right]\rangle\left[X_{[d]}\right] = \epsilon_1\left[X_k\right] \otimes \mathbb{I}\left[X_{[d]/\{k\}}\right] .$$

and has a basis encoding

$$\beta^{f^k}\left[Y_d, X_{[d]}\right] = \langle \delta\left[Y_k, X_k\right]\rangle\left[X_{[d]}\right] = \delta\left[Y_k, X_k\right] \otimes \mathbb{I}\left[X_{[d]/\{k\}}\right] .$$

The decomposition is depicted in a network diagram as



Proof. We have by definition

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

■

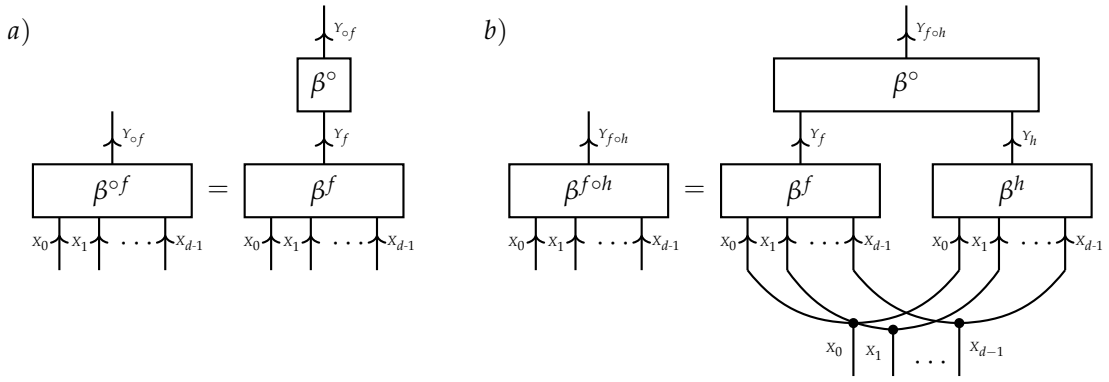


Figure 4.2: Representation of syntactical compositions of formulas by basis encodings. a) Unary connective $\circ$ acting on a formula f. b) Binary connective $\circ$ composing formulas f, h. In more generality, the basis encoding of arbitrary syntactical compositions is a contraction of the basis encoding of the components.

4.3.2 Syntactical Composition 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, typical logical formulas appearing in practical knowledge bases are sparse in the sense that they have short representations in a logical syntax. In logical syntax formulas are described by atomic formulas recursively combined via connectives, which are itself formulas of typically low order d. We now exploit such syntactical decompositions of formulas to derive tensor network decompositions.

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 have (see Figure 4.2b)

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

For any unary connective $\circ : [2] \to [2]$ we analogously have (see Figure 4.2a)

$$\beta^{\circ f}\left[Y_{\circ f}, X_{[d]}\right] = \left\langle \beta^{\circ}\left[Y_{\circ f}, Y_f\right], \beta^{f}\left[Y_f, X_{[d]}\right]\right\rangle \left[Y_{\circ f}, X_{[d]}\right]\,.$$

Lemma 4.6 *Let there be a r-ary connective*

$$\circ : \bigtimes_{l\in[r]} [2] \to [2]$$

and for each $l \in [r]$ a formula $f_l\left[X_{[d]}\right]$. For the composition formula $f\left[X_{[d]}\right]$ defined by

$$f\left[X_{[d]} = x_{[d]}\right] = \circ\left(f_0\left[X_{[d]} = x_{[d]}\right], \ldots, f_{r-1}\left[X_{[d]} = x_{[d]}\right]\right)$$

we have

$$\beta^{f\left[X_{[d]}\right]}\left[Y_f, X_{[d]}\right] = \left\langle \{\beta^{\circ}\left[Y_{f\left[X_{[d]}\right]}, Y_f, Y_{[r]}\right]\} \cup \{\beta^{f_l}\left[Y_l, X_{[d]}\right] \,:\, l \in [r]\}\right\rangle \left[Y_{f\left[X_{[d]}\right]}, X_{[d]}\right].$$

Proof. Here we used the main property of basis encodings, that basis encodings of composition are equal to contracted basis encodings of the components. We will show this property in more generality in Thm. 14.12. ∎

These properties motivate a hypergraph-based definition of syntactical compositions of a propositional formula.

Definition 4.7 A syntactical hypergraph is a directed acyclic hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see Def. 0.9) such that

- each hyperedge $e = (e^{\text{in}}, e^{\text{out}})$ has exactly one outgoing node, i.e. $\left|e^{\text{out}}\right| = 1$
- each node $v \in \mathcal{V}$ carries a boolean variable Y_v and appears at most once as the outgoing node of a hyperedge
- each hyperedge $(e^{\text{in}}, \{v\})$ with $e^{\text{in}} \neq \varnothing$ is decorated by a logical connective

$$\circ_v\left[Y_{e^{\text{in}}}\right] : \bigtimes_{v \in e^{\text{in}}} [2] \to [2]$$

- the node not appearing as an outgoing node are labeled by $[d]$

We say that the syntactical hypergraph is single-rooted, if exactly one node $\tilde{v}$ does not appear as an incoming node of a hyperedge. In this case this unique node is called the root node. We assign atomic formulas to the nodes $[d]$ and recursively assign to each further node v a node formula

$$f_v\left[X_{[d]} = x_{[d]}\right] = \circ_v\left[\left[f_{\bar{v}}\left[X_{[d]} = x_{[d]}\right] \,:\, \bar{v} \in e^{\text{in}}\right]\right] \quad \forall x_{[d]} \in \bigtimes_{k \in [d]} [m_k],$$

where e^{in} are the incoming nodes in the unique hyperedge with outgoing nodes $\{v\}$. We call the formula $f\left[X_{[d]}\right] := f_{\tilde{v}}\left[X_{[d]}\right]$ to the root note $\tilde{v}$ the syntactical composition of $\mathcal{G}$ and $\mathcal{G}$ is a syntactical decomposition of f.

We are now prepared to show the main result of this chapter, the representation of formulas as contracted basis encoding of their decomposition.

Theorem 4.8 *For any syntactical hypergraph $\mathcal{G}$ with composition f we have*

$$f\left[X_{[d]}\right] = \left\langle \left\{\beta^{\circ_v}\left[Y_v, Y_{e^{\text{in}}}\right] \,:\, (e^{\text{in}}, \{v\}) \in \mathcal{E}\right\} \cup \{\delta\left[Y_k, X_k\right] \,:\, k \in [d]\} \cup \{\epsilon_1\left[Y_{\tilde{v}}\right]\}\right\rangle \left[X_{[d]}\right].$$

Proof. We show inductively over the nodes, that

$$\beta^{f_v}\left[Y_v, X_{[d]}\right] = \left\langle \left\{\beta^{\circ_{\bar{v}}}\left[Y_{\bar{v}}, Y_{e^{\text{in}}}\right] \,:\, \bar{v} \prec v,\, (e^{\text{in}}, \{\bar{v}\}) \in \mathcal{E}\right\} \cup \{\beta^{\circ_v}\left[Y_v, Y_{e^{\text{in}}}\right]\} \cup \right. \tag{4.2}$$

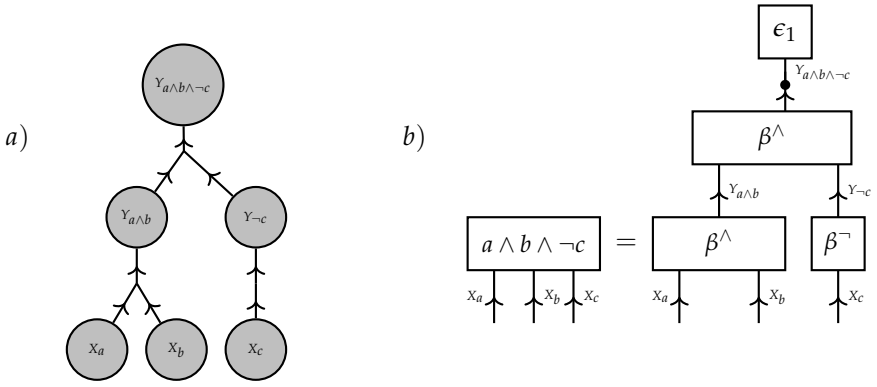


Figure 4.3: Decomposition of the formula tensor to $f = a \wedge b \wedge \neg c$ into unary (matrix) and binary (third order tensor) cores. a) Syntactical decomposition of f as a hypergraph $\mathcal{G}$ according to Def. 4.7. b) Tensor network of basis encodings of the connectives decorating the hyperedges, which represents by Thm. 4.8 the formula f.

$$\{\delta\left[Y_k, X_k\right] \,:\, k \in [d]\}\rangle\left[X_{[d]}\right].$$

By $\prec$ we denote the partial ordering of nodes by directed paths in $\mathcal{G}$, i.e. $\bar{v} \prec v$ holds if there is a directed path from $\bar{v}$ to v in the hypergraph $\mathcal{G}$. We start the induction at the nodes $[d]$ not appearing as outgoing nodes, where (4.2) holds by Lem. 4.5.

If for a node $v \in \mathcal{V}$ (4.2) holds at all $\bar{v} \in e^{\text{in}}$, we apply Lem. 4.6 to the connective $\circ_v$ and the formulas $f_{\bar{v}}\left[X_{[d]}\right]$ of the parents $\bar{v} \in e^{\text{in}}$ and get

$$\begin{aligned}
\beta^{f_v}\left[Y_v, X_{[d]}\right] &= \left\langle \{\beta^{\circ_v}\left[Y_{f_v}, Y_f, Y_{[r]}\right]\} \cup \{\beta^{f_l}\left[Y_l, X_{[d]}\right] \,:\, l \in [r]\}\right\rangle \left[Y_{f\left[X_{[d]}\right]}, X_{[d]}\right] \\
&= \langle \left\{\beta^{\circ_{\bar{v}}}\left[Y_{\bar{v}}, Y_{e^{\text{in}}}\right] \,:\, \bar{v} \prec v,\, (e^{\text{in}}, \{\bar{v}\}) \in \mathcal{E}\right\} \cup \{\beta^{\circ_v}\left[Y_v, Y_{e^{\text{in}}}\right]\} \cup \\
&\qquad \{\delta\left[Y_k, X_k\right] \,:\, k \in [d]\}\rangle\left[X_{[d]}\right].
\end{aligned}$$

Thus (4.2) also holds for v. Since the hypergraph is acyclic the induction reached all nodes and (4.2) holds in particular for the root node $\bar{v}$. The claim then follows with Thm. 4.3 applied on $f = f_{\bar{v}}$. ∎

An example of a syntactical decomposition is provided in Figure 4.3a, and the corresponding tensor network representation in Figure 4.3b. Examples of unary ($d = 1$) and binary ($d = 2$) connectives are further provided in Example 4.9.

Example 4.9 We use the following connectives:

- negation $\neg : [2] \to [2]$ by the vector

$$\neg[Y_f] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- conjunctions $\wedge : [2] \times [2] \to [2]$

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

- disjunctions $\vee : [2] \times [2] \to [2]$

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

- exact disjunction $\oplus : [2] \times [2] \to [2]$

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

- implications $\Rightarrow$: $[2] \times [2] \to [2]$

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

- biimplication $\Leftrightarrow$: $[2] \times [2] \to [2]$

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

$\Diamond$

Remark 4.10 (d-ary connectives such as $\wedge$ and $\vee$) Since the decomposition of basis encoding can be applied to generic function compositions (see Thm. 14.12), 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 basis encoding by any tree-structured composition of binary $\wedge$ and $\vee$ connectives. $\Diamond$

4.3.3 Syntactical Decomposition of Formulas

We have seen how syntactical decompositions of formulas into connectives acting can be exploited to find effective representations by tensor networks. We now provide a generic scheme to find syntactical decompositions to arbitrary propositional formulas.

Definition 4.11 (Terms and Clauses) Given two disjoint subsets $\mathcal{V}^0$ and $\mathcal{V}^1$ of $[d]$, the corresponding term is the formula defined on the indices $x_{[d]} \in \bigtimes_{k \in [d]} [2]$ by

$$Z^{\wedge}_{\mathcal{V}^0, \mathcal{V}^1}\left[X_{[d]}\right] = \left(\bigwedge_{k \in \mathcal{V}^0} \neg f^k \right) \wedge \left(\bigwedge_{k \in \mathcal{V}^1} f^k \right)$$

and the corresponding clause is the formula defined on the indices $x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [2]$ by

$$Z^{\vee}_{\mathcal{V}^0, \mathcal{V}^1}\left[X_{[d]}\right] = \left(\bigvee_{k \in \mathcal{V}^0} f^k \right) \vee \left(\bigvee_{k \in \mathcal{V}^1} \neg f^k \right) ,$$

where by $\wedge_{k \in \mathcal{V}}$ and $\vee_{k \in \mathcal{V}}$ we refer to the n-ary connectives $\wedge$ and $\vee$. We call the term a

minterm and the clause a maxterm, if $\mathcal{V}^0 \cup \mathcal{V}^1 = [d]$.

Terms and Clauses have for any index tuple $x_{[d]}$ a polynomial representation by

$$Z^{\wedge}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]} = x_{[d]}] = \left(\prod_{k\in\mathcal{V}^0}(1-x_k)\right)\left(\prod_{k\in\mathcal{V}^1} x_k\right)$$

and

$$Z^{\vee}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]} = x_{[d]}] = 1 - \left(\prod_{k\in\mathcal{V}^0}(1-x_k)\right)\left(\prod_{k\in\mathcal{V}^1} x_k\right).$$

Lemma 4.12 *Terms are contractions of one-hot encodings, that is for any disjoint subsets $\mathcal{V}^0, \mathcal{V}^1 \subset [d]$ we have*

$$Z^{\wedge}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]}] = \left\langle \epsilon_{\{x_k=0:k\in\mathcal{V}^0\}\cup\{x_k=1:k\in\mathcal{V}^1\}} \right\rangle \left[X_{[d]}\right].$$

Clauses are subtractions of one-hot encodings from the trivial tensor, that is for any disjoint subsets $\mathcal{V}^0, \mathcal{V}^1 \subset [d]$ we have

$$Z^{\vee}_{\mathcal{V}^0,\mathcal{V}^1}[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - \left\langle \epsilon_{\{x_k=0:k\in\mathcal{V}^0\}\cup\{x_k=1:k\in\mathcal{V}^1\}} \right\rangle \left[X_{[d]}\right].$$

The reference of the formulas in the case $\mathcal{V}^0 \dot\cup \mathcal{V}^1 = [d]$ as minterms and maxterms is due to the fact, that minterms are formulas with unique models and maxterms are formulas with a unique world not satisfying the formula. We use this insight and enumerate maxterms and minterms by the index $x \in \bigtimes_{k\in[d]}[2]$ of the unique world where the minterm is satisfied, respectively the maxterm is not satisfied. For any $\mathcal{V}^0 \dot\cup \mathcal{V}^1 = [d]$ we take the index tuple $x_0, \ldots, x_{d-1}$ where $x_k = 0$ if $k \in \mathcal{V}^0$ and $x_k = 1$ if $k \in \mathcal{V}^1$ and define

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

Corollary 4.13 *Minterms are basis elements of the tensor space, that is for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have*

$$Z^{\wedge}_{x_{[d]}} = \epsilon_{x_{[d]}}\left[X_{[d]}\right]$$

Maxterms are subtraction of basis elements from the trivial tensor, that is for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have

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

Proof. Follows from Lem. 4.12, 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 4.14 *For any boolean tensor* $\tau\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^2$ *we have*

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

and

$$\tau\left[X_{[d]}\right] = \left(\bigwedge_{x_{[d]} : \tau\left[X_{[d]}=x_{[d]}\right]=0} Z^{\vee}_{\{k:x_k=0\},\{k:x_k=1\}} \right) [X_{[d]}] \,.$$

Proof. To show the representation by minterms we use the decomposition

$$\tau\left[X_{[d]}\right] = \sum_{x_{[d]} : \tau\left[X_{[d]}=x_{[d]}\right]=1} \epsilon_{x_{[d]}}\left[X_{[d]}\right]$$

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

$$\tau\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] \quad - \sum_{x_{[d]} : \tau\left[X_{[d]}=x_{[d]}\right]=0} \epsilon_{x_{[d]}}\left[X_{[d]}\right]$$

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 τ and are thus identical to τ. ∎

The decompositions found in Thm. 4.14 are also called canonical normal forms to propositional formulas $\tau\left[X_{[d]}\right]$.

Remark 4.15 (Efficient Representation in Propositional Syntax) The decomposition in Thm. 4.14 is a basis CP decomposition of the binary tensor and will further be investigated in Chapter 15. The formulas constructed in the proof of Thm. 4.14 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 Thm. 4.14. 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. ◊

4.4 Outlook

While we in this chapter investigated representation schemes for single propositional formulas, we will further study the representation of knowledge bases consisting in multiple formulas in Sect. 8.2. Further, we will build hybrid models bridging the concepts of probability distributions and propositional logics in Sect. 8.3. Propositional formulas will therein serve as features and base measures for exponential families.

5

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.

5.1 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 5.1 (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.

5.1.1 Deciding Entailment by Contractions

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

Theorem 5.2 (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

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

"⇒": 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. ■

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 5.3 *Given propositional formulas $\mathcal{KB}$ and f we build*

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

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

Proof. This follows from Thm. 5.2 using that

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

and

$$\langle \mathcal{KB}, f \rangle [\varnothing] = \tau\left[Y_f = 1\right]. \qquad ■$$

5.1.2 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 Thm. 14.3. For two propositional formulas, we denote to this end $f \prec h$ (see Def. 14.2), if and only if for all $x_{[d]} \in \bigtimes_{k\in[d]}[2]$

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

Theorem 5.4 (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 $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ 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

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

$\mathcal{KB} \models \neg f$ \ $\mathcal{KB} \models f$		False	True
	False	*"Contingent"*	*"Entailed"*
	True	*"Contracticted"*	*"Inconsistent"*

Figure 5.1: Table of possible logical relations between a knowledge base $\mathcal{KB}$ and f, based on whether the knowledge base entails the formula ($\mathcal{KB} \models f$) and its negation ($\mathcal{KB} \models \neg f$).

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

5.1.3 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 5.5 (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

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

Now, by Thm. 5.2 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

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

5.1.4 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 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 Thm. 5.3 we decide these situations based on a single contraction.

The function TELL(f) incorporates an additional formula f into a Knowledge Base $\mathcal{KB}$. Here we exploit Thm. 5.5 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 with operations ASK and TELL

ASK($\mathcal{KB}$, f)

Input: Knowledge base $\mathcal{KB}$, query formula f
Output: Decision which relation between $\mathcal{KB}$ and f holds (see Figure 5.1)

$\tau\left[Y_f\right] \leftarrow \left\langle \{h\left[X_{[d]}\right] : h \in \mathcal{KB}\}, \beta^f\left[Y_f, X_{[d]}\right] \right\rangle \left[Y_f\right]$
if $\tau\left[Y_f = 0\right] = 0$ and $\tau\left[Y_f = 1\right] = 0$ **then**
  return *"Inconsistent"*
else if $\tau\left[Y_f = 0\right] = 0$ **then**
  return *"Entailed"*
else if $\tau\left[Y_f = 1\right] = 0$ **then**
  return *"Contradicted"*
else
  return *"Contingent"*
end if

TELL($\mathcal{KB}$, f)

Input: Knowledge base $\mathcal{KB}$, query formula f
Output: Decision whether a formula is added to the knowledge base (*"Added"*), or an exception in (*"Inconsistent"*, *"Redundant"* or *"Refused"*) is raised.

answer $\leftarrow$ ASK(f)
if answer is *"Inconsistent"*: **then**
  return *"Inconsistent"*
else if answer is *"Entailed"*: **then**
  return *"Redundant"*
else if answer is *"Contradicted"*: **then**
  return *"Refused"*
else if answer is *"Contingent"*: **then**
  $\mathcal{KB} \leftarrow \mathcal{KB} \cup \{f\}$
  return *"Added"*
end if

5.2 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 basis 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 β^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.

5.2.1 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. 3.1, 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 basis encoding as

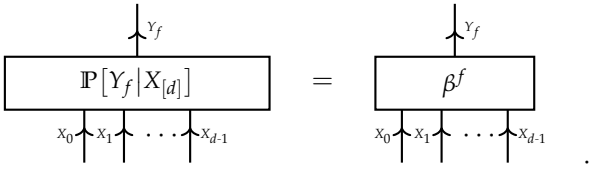
$$\mathbb{P}\left[Y_f, X_{[d]}\right] = \left\langle \beta^f \left[Y_f, X_{[d]}\right], \mathbb{P}\left[X_{[d]}\right] \right\rangle \left[X_{[d]}\right] .$$

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

Conditioning this probability distribution on the atoms, we get

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

We thus interpret the basis 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 $x_{[d]}$

$$\mathbb{P}\left[Y_f \middle| X_{[d]} = x_{[d]}\right] = \begin{cases} \epsilon_0 \left[Y_f\right] & \text{if} \quad f\left[X_{[d]} = x_{[d]}\right] = 0, \text{ i.e. } x_{[d]} \text{ is not a model of } f \\ \epsilon_1 \left[Y_f\right] & \text{if} \quad f\left[X_{[d]} = x_{[d]}\right] = 1, \text{ i.e. } x_{[d]} \text{ is a model of } f \end{cases} .$$

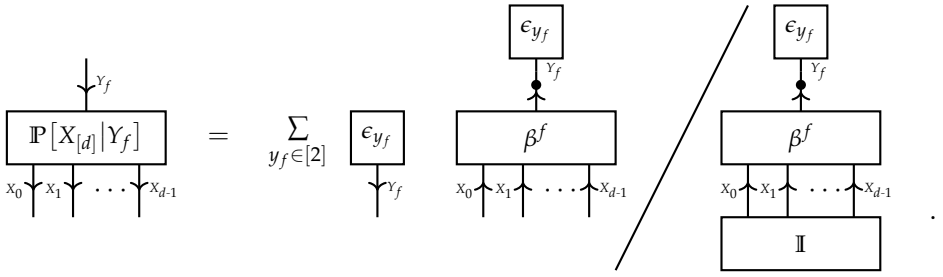
Since the conditional query $\mathbb{P}\left[Y_f \middle| X_{[d]}\right]$ provides an interpretation of β^f as a conditional probability, we interpret $\mathbb{P}\left[Y_f\right]$ as a marginal distribution inherited by $\mathbb{P}\left[X_{[d]}\right]$. This is also reflected in the fact that both $\mathbb{P}\left[Y_f \middle| X_{[d]}\right]$ and $\beta^f \left[Y_f, X_{[d]}\right]$ are directed, since the first is a normalization by Def. 3.1 and the second an basis encoding of a formula. Probabilistic queries (see Def. 3.1), which functions are propositional formulas are thus answered by the satisfaction rate of a propositional formula given a joint distribution of the corresponding atoms.

5.2.2 Uniform Distributions of the Models

Let us now converse the order of conditioning from $\mathbb{P}\left[Y_f \middle| X_{[d]}\right]$ to $\mathbb{P}\left[X_{[d]} \middle| Y_f\right]$. 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 $\{\beta^f \left[Y_f, X_{[d]}\right]\}$ a Markov Network $\mathbb{P}^{\{f\}\cup[d]} \left[Y_f, X_{[d]}\right]$. By definition we have

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

We depict this construction by:



Let us further investigate the slices of $\mathbb{P}\left[X_{[d]}\middle|f\right]$ 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

$$\mathbb{P}\left[X_{[d]}\middle|Y_f = 1\right] = \frac{1}{\langle f\rangle\,[\varnothing]} \sum_{x_{[d]}\in\times_{k\in[d]}[2]\,:\,f\left[X_{[d]}=x_{[d]}\right]=1} \epsilon_{x_{[d]}}\left[X_{[d]}\right]. \tag{5.1}$$

With $\langle f\rangle\,[\varnothing]$ being the number of models of f, this is the uniform distribution among the models of f. Conversely, when conditioning on the event $Y_f = 0$ we get a uniform distribution of the models of $\neg f$.

The probability distribution in Equation (5.1) 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 (5.1) 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 $2^{(2^d)}$.

5.2.3 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

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

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

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

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

Theorem 5.6 *Given a satisfiable formula $\mathcal{KB}$, we have $\mathcal{KB} \models f$, if and only if*

$$\mathbb{P}\left[Y_f = 0\middle|Y_{\mathcal{KB}} = 1\right] = 0.$$

Proof. Since $\mathcal{KB}$ is satisfiable, we have $\langle \mathcal{KB} \rangle [\varnothing] > 0$ and

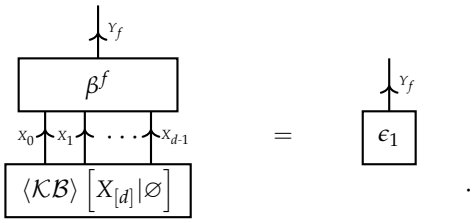
$$\mathbb{P}\left[Y_f = 0 \middle| Y_{\mathcal{KB}} = 1\right] = \frac{\langle \neg f, \mathcal{KB} \rangle [\varnothing]}{\langle \mathcal{KB} \rangle [\varnothing]}.$$

This term vanishes if and only if $\langle \neg f, \mathcal{KB} \rangle [\varnothing]$ vanish. Now, by Thm. 5.2 we have $\mathcal{KB} \models f$ if and only if $\langle \mathcal{KB}, \neg f \rangle [\varnothing] = 0$, which is therefore equal to $\mathbb{P}\left[Y_f = 0 \middle| Y_{\mathcal{KB}} = 1\right] = 0$. ∎

Since any conditional distribution is directed, we have

$$\begin{aligned} \mathbb{P}\left[Y_f \middle| Y_{\mathcal{KB}} = 1\right] = \epsilon_0 \left[Y_f\right] & \text{if} \quad \mathcal{KB} \models \neg f \\ \epsilon_1 \left[Y_f\right] & \text{if} \quad \mathcal{KB} \models f \\ \notin \{\epsilon_0 \left[Y_f\right], \epsilon_1 \left[Y_f\right]\} & \text{else}. \end{aligned}$$

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



.

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

Theorem 5.7 *Given a formula $\mathcal{KB}$, we have $\mathcal{KB} \models f$ (respectively $\mathcal{KB} \models \neg f$), if and only if*

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

Proof. This follows from Thm. 5.2 using that

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

Relating entailment to probability distributions motivates an extension of the entailment as provided by Def. 5.1 to arbitrary probability distributions.

Definition 5.8 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

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

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

We note, that when choosing for a formula $\mathcal{KB}$ the uniform distribution

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle X_{[d]} \right\rangle [Y_{\mathcal{KB}} = 1 | \varnothing]$$

among its models, then probabilistic entailment $\mathbb{P} \models f$ of a propositional formula f is by Thm. 5.6 equivalent to $\mathcal{KB} \models f$.

5.2.4 Knowledge Bases as Base Measures for Probability Distributions

Let us now further relate the probabilistic entailment provided by Def. 5.8 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 $\mathbb{I}_{\neq 0} : \mathbb{R} \to \mathbb{R}$ denotes the indicator function of the support defined as

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

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

Theorem 5.9 *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. 2.2) and positivity (see Def. 2.3) of distributions with respect to boolean base measures.

Theorem 5.10 *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. 2.2, $\mathbb{P}$ 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. 5.1 equal to $\nu \models \mathbb{I}_{\neq 0} \circ \mathbb{P}$.

To prove 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 $\nu \models \mathbb{I}_{\neq 0} \circ \mathbb{P}$. ∎

5.3 Entropy Optimization in Logics

Since normalizations of propositional formulas are probability distributions, we can apply the concepts of probabilistic reasoning. To develop results on entropy optimization, let us characterize the relative entropy between formulas using the entailment formalism.

Lemma 5.11 *Given two satisfiable formulas f, h we have*

$$\mathrm{D}_{\mathrm{KL}}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right] || \langle h\rangle\left[X_{[d]}|\varnothing\right]\right] = \begin{cases} \ln\left[\langle h\rangle[\varnothing]\right] - \ln\left[\langle f\rangle[\varnothing]\right] & \text{if} \quad f \models h \\ \infty & \text{if} \quad f \not\models h \end{cases}.$$

Proof. The relative entropy (see Def. 3.8) decomposed as

$$\mathrm{D}_{\mathrm{KL}}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right] || \langle h\rangle\left[X_{[d]}|\varnothing\right]\right] = \mathbb{H}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right], \langle h\rangle\left[X_{[d]}|\varnothing\right]\right] - \mathbb{H}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right]\right]$$

The entropy term is

$$\begin{aligned} \mathbb{H}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right]\right] &= \left\langle \frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}, -\ln\left[\frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}\right]\right\rangle[\varnothing] \\ &= \left\langle \frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}, -\ln\left[f\left[X_{[d]}\right]\right]\right\rangle[\varnothing] + \ln\left[\langle f\rangle[\varnothing]\right] \cdot \left\langle \frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}\right\rangle[\varnothing] \\ &= \ln\left[\langle f\rangle[\varnothing]\right]. \end{aligned}$$

In the last equation we use the convention $0 \cdot \ln[0] = 0$.

The cross entropy term is

$$\mathbb{H}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right], \langle h\rangle\left[X_{[d]}|\varnothing\right]\right] = \left\langle \frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}, -\ln\left[\frac{h\left[X_{[d]}\right]}{\langle h\rangle[\varnothing]}\right]\right\rangle[\varnothing]$$

If and only if $f \not\models h$ there is a state $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ such that $h\left[X_{[d]} = x_{[d]}\right] = 0$ and $f\left[X_{[d]} = x_{[d]}\right]$, which contributes to the contraction the summand

$$(-1)\cdot f\left[X_{[d]} = x_{[d]}\right]\cdot \ln\left[h\left[X_{[d]} = x_{[d]}\right]\right] = \infty\,.$$

If and only if $f \not\models h$ we therefore have

$$\mathrm{D}_{\mathrm{KL}}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right] \,||\, \langle h\rangle\left[X_{[d]}|\varnothing\right]\right] = \infty\,.$$

In the case $f \models h$ we get

$$\begin{aligned}\left\langle \frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}, -\ln\left[\frac{h\left[X_{[d]}\right]}{\langle h\rangle[\varnothing]}\right]\right\rangle[\varnothing] &= \left\langle \frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}, -\ln\left[h\left[X_{[d]}\right]\right]\right\rangle[\varnothing] + \ln\left[\langle h\rangle[\varnothing]\right]\cdot\left\langle \frac{f\left[X_{[d]}\right]}{\langle f\rangle[\varnothing]}\right\rangle[\varnothing] \\ &= \ln\left[\langle h\rangle[\varnothing]\right]\end{aligned}$$

and with the above

$$\mathrm{D}_{\mathrm{KL}}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right] \,||\, \langle h\rangle\left[X_{[d]}|\varnothing\right]\right] = \ln\left[\langle h\rangle[\varnothing]\right] - \ln\left[\langle f\rangle[\varnothing]\right]\,. \qquad \blacksquare$$

We use Lem. 5.11 to characterize the logical analogue of the M-projection and the I-projection.

Theorem 5.12 *Given a set of formulas $\mathcal{H}$ we get for the logical M-projection*

$$\min_{f\in\mathcal{H}} \mathrm{D}_{\mathrm{KL}}\left[\langle h\rangle\left[X_{[d]}|\varnothing\right] \,||\, \langle f\rangle\left[X_{[d]}|\varnothing\right]\right] = \min_{f\in\mathcal{H}\,:\,h\models f} \ln\left[\langle f\rangle[\varnothing]\right] - \ln\left[\langle h\rangle[\varnothing]\right]$$

and for the logical I-projection

$$\min_{f\in\mathcal{H}} \mathrm{D}_{\mathrm{KL}}\left[\langle f\rangle\left[X_{[d]}|\varnothing\right] \,||\, \langle h\rangle\left[X_{[d]}|\varnothing\right]\right] = \min_{f\in\mathcal{H}\,:\,f\models h} \ln\left[\langle h\rangle[\varnothing]\right] - \ln\left[\langle f\rangle[\varnothing]\right]\,.$$

Proof. We use the characterization of the relative entropy from Lem. 5.11 and restrict the feasible set of the minimizations to the respective cases, where the cost is finite. Note that if $h \models f$ (respectively $f \models h$) is not satisfied for $f \in \mathcal{H}$, then the feasible sets of the right sight are empty and we use the convention that the minimum is ∞. $\blacksquare$

5.4 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 [RN21], we now define Constraint Satisfaction Problems.

Definition 5.13 Let there be a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\tau^{\mathcal{G}}$ be a tensor network of boolean constraint tensors $\tau^e[X_e]$ to each $e \in \mathcal{E}$, that is

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

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

such that

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

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

5.4.1 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 5.14 *Let $\mathbb{P}^{\mathcal{G}}$ be a Markov Network to the Tensor Network $\tau^{\mathcal{G}} = \{\tau^e [X_e] : e \in \mathcal{E}\}$ on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For each $e \in \mathcal{E}$ we build the factor constraint cores*

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

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

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

is unsatisfiable.

Proof. We first show, that

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

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 $\tau^e [X_e = x_e]$. But this is equivalent to

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

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

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

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

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

which is equal to

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

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

■

For any positive tensor τ we have

$$\mathbb{I}_{\neq 0} \circ \tau [X_e] = \mathbb{I} [X_e] \,,$$

which does not influence the distribution and can be omitted from the Markov Network. By Thm. 5.14, 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 $\mathbb{I}_{\neq 0} \circ \tau\left[X_e\right]$ are boolean, each is a propositional formula and the Markov Network is turned into a Knowledge Base of their conjunctions. Deciding probabilistic entailment is thus traced back to logical entailment.

Exponential families have a tensor network representation by a Markov Network (see Thm. 2.24). However, all factors corresponding with a coordinate of the statistic $\mathcal{S}$ 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.

5.4.2 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 5.15 (Categorical Constraint and Atomization Variables) Given a list $X_0, \ldots, X_{m-1}$ of boolean variables and a categorical variable X with dimension m a categorical constraint is a tensor $\beta^e\left[X_{[m]}, X\right]$ with coordinates

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

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

We notice that the categorical constraint tensor is the basis encoding of the one-hot map on $[m]$. With Thm. 15.20 the basis encoding β^e decomposes in a basis CP format (see Figure 5.2b) of if its coordinate maps e^k, where $k \in [m]$, defined as

$$e^k\left(l\right) = \begin{cases} 1 & \text{if} \quad \tilde{k} = k \\ 0 & \text{else}\,. \end{cases}$$

Their basis encoding are decomposed as

$$\beta^{e^k}\left[X_k, X\right] = \epsilon_1\left[X_k\right] \otimes \epsilon_k\left[X\right] + \epsilon_0\left[X_k\right] \otimes \left(\mathbb{I}\left[X\right] - \epsilon_k\left[X\right]\right).$$

We thus have by Thm. 15.20

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

In the next theorem we show how a categorical constraint can be enforced in a tensor network by adding the tensor e to a contraction.

Theorem 5.16 *For any tensor $\tau\left[X_{[d]}\right]$ and a categorical constraint defined by an ordered subset $X_A \subset X_{[d]}$, a variable $X \in X_{[d]}$ we have*

$$\left\langle \tau\left[X_{[d]}\right], \beta^e\left[X_A, X\right] \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]$$

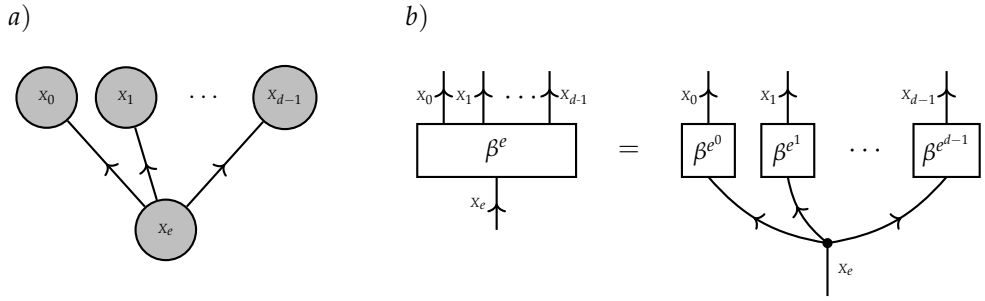


Figure 5.2: Representation of a categorical constraint in a CP Format tensor network. a) Representation of the dependency of the graphical model. b) Tensor Representation with further network decomposition.

$$= \begin{cases} \tau\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] & \text{if} \quad \exists x : x_A = \epsilon_x \\ 0 & \text{else}\,. \end{cases}$$

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

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

$$\left\langle \left\{ \tau\left[X_{[d]}\right], \beta^e\left[X_A, X\right] \right\} \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = \tau\left[x_{[d]}\right] \cdot \beta^e\left[X_A = x_A, X = x\right]\,.$$

If $x_A = \epsilon_x$ we have $e[X_A = x_A, X = x] = 1$ and thus

$$\left\langle \tau\left[X_{[d]}\right], \beta^e\left[X_A, X\right] \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = \tau\left[x_{[d]}\right]\,.$$

If this is not the case then $\beta^e\left[X_A = x_A, X = x\right] = 0$ and

$$\left\langle \tau\left[X_{[d]}\right], \beta^e\left[X_A, X\right] \right\rangle \left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = 0\,. \qquad \blacksquare$$

Remark 5.17 (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

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

$\Diamond$

Example 5.18 (Sudoku) An interesting example, where categorical constraints are combined is Sudoku, the game of assigning numbers to a grid (see for example Section 5.2.6 in [RN21]). The basic variables therein are $X_{i,j}$, with $m_{i,j} = n^2$ and $i, j \in [n^2]$. By understanding i as a line index and j as a column index, they are ordered in a grid as sketched in Figure 5.3 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 categorical constraints on the sets

$$\{X_{i,j,k} \,:\, k \in [n^2]\}\,.$$

We further have $3 \cdot n^2$ constraints by the

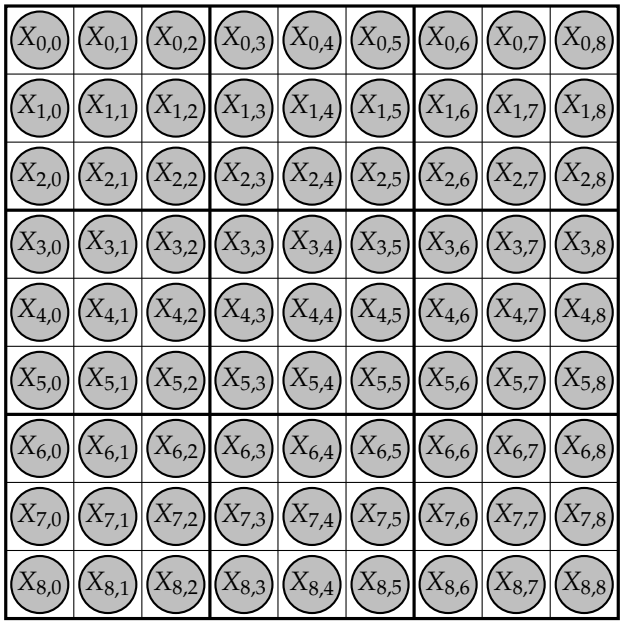


Figure 5.3: Sudoku grid of basic categorical variables $X_{i,j}$, here drawn in the standard case of $n = 3$, each with dimension $m = n^2 = 9$. Each basic categorical variables has n^2 corresponding atomization variables, which are further atomization variables to the row, column and squares constraints. Instead of depicting those constraints by hyperedges in a variable dependency graph, we here just indicate their existence through row, column and squares blocks.

- 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 [Sim05], which is NP-hard [AH08]. 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. 5.5.

$\Diamond$

5.5 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 Thm. 5.2 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.

5.5.1 Monotonicity of Entailment

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

Theorem 5.19 (Monotonicity of Entailment) *For any Markov Network $\mathbb{P}^{\mathcal{G}}$ 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 5.20 *For any non-negative tensor network $\tau^{\mathcal{G}}$ 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 $\tau^{\tilde{\mathcal{G}}}$ with tensors coinciding on $\tilde{\mathcal{E}}$ with those in $\tau^{\mathcal{G}}$ we have*

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

if $\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [\varnothing] = 0$.

Proof. Since the tensor network $\tau^{\tilde{\mathcal{G}}}$ is non-negative, we have whenever $\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [\varnothing] = 0$ that

$$\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] = 0\, [X_{\tilde{\mathcal{V}}}]\ .$$

It follows with the commutation of contractions (see Thm. 17.1 in Chapter 17), that

$$\begin{aligned} \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}] &= \left\langle \{\tau^e \,:\, e \in \mathcal{E}/\tilde{\mathcal{E}}\} \cup \{\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}]\} \right\rangle [X_{\mathcal{V}}] \\ &= \left\langle \{\tau^e \,:\, e \in \mathcal{E}/\tilde{\mathcal{E}}\} \cup \{0\, [X_{\tilde{\mathcal{V}}}]\} \right\rangle [X_{\mathcal{V}}] \\ &= 0 \end{aligned}$$

Thus, also the contraction of $\tau^{\mathcal{G}}$ vanishes in this case. ∎

Proof of Thm. 5.19. We use Lem. 5.20 on the subset $\tau^{\tilde{\mathcal{G}}}$ of the cores $\tau^{\mathcal{G}}$ 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 Thm. 5.2

$$\left\langle \tau^{\tilde{\mathcal{G}}} \cup \{\neg f\} \right\rangle [\varnothing] = 0\,.$$

It follows with Lem. 5.20 that also

$$\left\langle \tau^{\mathcal{G}} \cup \{\neg f\} \right\rangle [\varnothing] = 0\,.$$

and therefore $\mathbb{P}^{\mathcal{G}} \models f$. ∎

Remark 5.21 To make use of Thm. 5.19 we can exploit any entailment criterion. However, there is no general statement about entailment possible, when the local entailment does not hold. Thm. 5.19 therefore just provides a sufficient but not necessary criterion of entailment with respect to $\mathbb{P}^{\mathcal{G}}$. ♢

5.5.2 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

$$\kappa^e\left[X_e\right] = \bigvee_{x_e \,:\, \kappa^e[X_e = x_e]} \bigwedge_{v \in e} \left(X_e = x_e\right) .$$

Definition 5.22 Let $\tau^{\mathcal{G}}$ be a constraint satisfaction problem. We say that a knowledge core $\kappa^e\left[X_e\right]$ is sound for $\tau^{\mathcal{G}}$, if

$$\mathbb{I}_{\neq 0} \circ \left\langle \tau^{\mathcal{G}} \right\rangle \left[X_e\right] \prec \kappa^e\left[X_e\right]$$

and complete for $\tau^{\mathcal{G}}$ if in addition

$$\mathbb{I}_{\neq 0} \circ \left\langle \tau^{\mathcal{G}} \right\rangle \left[X_e\right] = \kappa^e\left[X_e\right] .$$

5.5.3 Knowledge Propagation

We now provide a solution algorithm for constraint satisfaction problems by propagating local contractions. The dynamic programming paradigm is implemented by the storage of partial entailment results in Knowledge Cores. We then iterate over local entailment checks, where we recursively add further entailment checks to be redone due to additional knowledge. This local entailment scheme is called Knowledge Propagation and described in a generic way in Algorithm 7.

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 [Mac77]) is a specific instance, where knowledge cores are assigned to single variables and propagation is performed on single constraint cores.

Theorem 5.23 *At any state of the Knowledge Propagation Algorithm 7, we have that each knowledge core $\kappa^{\tilde{e}}$ is sound for $\tau^{\mathcal{G}}$. After each update in Algorithm 7, $\kappa^{\tilde{e}}$ is further monotonically decreasing with respect to the partial ordering.*

Proof. We show the first claim by induction over the update steps in Algorithm 7. At the start, where $\kappa^e\left[X_e\right] = \mathbb{I}\left[X_e\right]$, we trivially have

$$\left\langle \tau^{\mathcal{G}} \cup \left\{\kappa^e\left[X_e\right] \,:\, e \in \mathcal{E}^k\right\} \right\rangle \left[X_{\mathcal{V}}\right] = \left\langle \tau^{\mathcal{G}} \cup \left\{\mathbb{I}\left[X_e\right] \,:\, e \in \mathcal{E}^k\right\} \right\rangle \left[X_{\mathcal{V}}\right] = \left\langle \tau^{\mathcal{G}} \right\rangle \left[X_{\mathcal{V}}\right] .$$

Let us now assume, that for a state of cores $\left\{\kappa^e \,:\, e \in \mathcal{E}^k\right\}$ the first claim holds and let $\tilde{\mathcal{E}} \subset \mathcal{E}$ be chosen for the update of $\kappa^{\tilde{e}}$. By the invariance under adding the support of subcontractions, which we will proof in more detail as Thm. 17.8 in Chapter 17, we have for the update

$$\tilde{\kappa}^{\tilde{e}}[X_{\tilde{e}}] = \mathbb{I}_{\neq 0} \circ \left\langle \left\{\tau^e \,:\, e \in \tilde{\mathcal{E}}\right\} \cup \left\{\kappa^e \,:\, e \in \mathcal{E}^k,\, e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing\right\} \right\rangle [X_{\tilde{e}}]$$

Algorithm 7 Knowledge Propagation

Input: Boolean Tensor Network $\tau^{\mathcal{G}}$ on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, domain edges $\mathcal{E}^k$ and a set $\mathcal{U}$ of subsets of $\mathcal{E}$ for local propagation
Output: Knowledge cores $\kappa^e[X_e]$ for $e \in \mathcal{E}^k$ with $\left\langle \tau^{\mathcal{G}} \right\rangle [X_e] \prec \kappa^e[X_e]$

Initialize for all $e \in \mathcal{E}^k$:

$$\kappa^e[X_e] = \mathbb{I}[X_e]$$

Initialize a queue

$$\mathcal{Q} = \mathcal{U}$$

while $\mathcal{Q}$ is not empty **do**
Choose a set of edges from the queue

$$\tilde{\mathcal{E}} \leftarrow \mathcal{Q}.\text{pop}()$$

for all $e \in \mathcal{E}^k$ with $e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing$ **do**
Contract

$$\tau[X_e] = \mathbb{I}_{\neq 0} \circ \left\langle \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \in \tilde{\mathcal{E}}\} \cup \{\kappa^e[X_e] : e \in \mathcal{E}^k, e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing\} \right\rangle [X_e]$$

if $\tau[X_e] \neq \kappa^e[X_e]$ **then**

$$\kappa^e[X_e] \leftarrow \tau[X_e]$$

for all $\tilde{\mathcal{E}} \in \mathcal{U}$ with $e \cap \bigcup_{\tilde{e} \in \tilde{\mathcal{E}}} \tilde{e} \neq \varnothing$ **do**

$$\mathcal{Q}.\text{push}(\tilde{\mathcal{E}})$$

end for
end if
end for
end while
return $\{\kappa^e[X_e] : e \in \mathcal{E}^k\}$

that

$$\left\langle \tau^{\mathcal{G}} \cup \{\kappa^e[X_e] : e \in \mathcal{E}^k\} \right\rangle [X_{\mathcal{V}}] = \left\langle \tau^{\mathcal{G}} \cup \{\kappa^e[X_e] : e \in \mathcal{E}^k\} \cup \{\tilde{\kappa}^{\tilde{e}}[X_{\tilde{e}}]\} \right\rangle [X_{\mathcal{V}}] .$$

Thus, the first claim holds also after the update of the core to $\tilde{e}$.

We further have with the monotonicity of boolean contraction (see Thm. 17.8) that for any update of $\kappa^{\tilde{e}}$ by $\tilde{\kappa}^{\tilde{e}}$

$$\tilde{\kappa}^{\tilde{e}}[X_{\tilde{e}}] = \mathbb{I}_{\neq 0} \circ \left\langle \{\tau^e : e \in \tilde{\mathcal{E}}\} \cup \{\kappa^e : e \in \mathcal{E}^k, e \cap \bigcup_{e \in \tilde{\mathcal{E}}} e \neq \varnothing\} \right\rangle [X_{\tilde{e}}] \prec \kappa^{\tilde{e}}[X_{\tilde{e}}] .$$

Thus, each Knowledge Core is monotonously decreasing at each update, with respect to the partial tensor ordering.

From the first claim we further have for any $\tilde{e} \in \tilde{\mathcal{E}}$

$$\left\langle \{\kappa^{\tilde{e}}\,[X_{\tilde{e}}]\} \cup \tau^{\mathcal{G}} \cup \left\{\kappa^{e} \,:\, e \in \mathcal{E}^k / \{\tilde{e}\}\right\} \right\rangle [X_{\tilde{e}}] = \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{e}}]$$

And thus in combination with the monotonicity of boolean contraction (see Thm. 17.8) that

$$\mathbb{I}_{\neq 0}\left(\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{e}}]\right) \prec \kappa^{\tilde{e}}\,[X_{\tilde{e}}]\ . \qquad \blacksquare$$

Let us now show that Knowledge Propagation always terminates. We can further characterize the knowledge cores at termination.

Definition 5.24 We say that a set of knowledge cores $\{\kappa^e \,:\, e \in \mathcal{E}^k\}$ is consistent with a set $\{\tau^e \,:\, e \in \tilde{\mathcal{E}}\}$, if for any $e \in \mathcal{E}^k$

$$\kappa^e\,[X_e] = \mathbb{I}_{\neq 0} \circ \left\langle \{\tau^e \,:\, e \in \tilde{\mathcal{E}}\} \cup \{\kappa^e \,:\, e \in \mathcal{E}^k\} \right\rangle [X_e]\ .$$

This property is similar to the completeness of a knowledge core, when interpreting the other knowledge cores and the constraints $\{\tau^e \,:\, e \in \tilde{\mathcal{E}}\}$ as posing a Constraint Satisfaction Problem.

Theorem 5.25 *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 $\{\kappa^e \,:\, e \in \mathcal{E}^k,\, e \cap \bigcup_{e \in \tilde{\mathcal{E}}} e \neq \varnothing\}$ are consistent with $\{\tau^e \,:\, e \in \tilde{\mathcal{E}}\}$.*

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

$$\kappa^e\,[X_e] = \mathbb{I}_{\neq 0} \circ \left\langle \{\tau^e \,:\, e \in \tilde{\mathcal{E}}\} \cup \{\kappa^e \,:\, e \in \mathcal{E}^k,\, e \cap \bigcup_{e \in \tilde{\mathcal{E}}} e \neq \varnothing\} \right\rangle [X_e]\ .$$

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. $\blacksquare$

We can exploit the Knowledge Propagation Algorithm 7 for the solution of Constraint Satisfaction Problems, by taking $\tau^{\mathcal{G}}$ 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 5.26 *Let us for a Constraint Satisfaction Problem encoded by $\tau^{\mathcal{G}}$ run Knowledge Propagation Algorithm 7. Whenever for any $\tilde{e} \in \mathcal{E}$ we have $\kappa^{\tilde{e}}\,[X_{\tilde{e}}] = 0\,[X_{\tilde{e}}]$, then the Constraint Satisfaction Problem is not satisfiable.*

Proof. Whenever $\kappa^{\tilde{e}}\,[X_{\tilde{e}}] = 0\,[X_{\tilde{e}}]$, then we have by Thm. 5.23

$$\mathbb{I}_{\neq 0}\left(\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{e}}]\right) \prec 0\,[X_{\tilde{e}}]$$

and therefore

$$\left\langle \tau^{\mathcal{G}} \right\rangle [\varnothing] = 0 \,. \qquad \blacksquare$$

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 $\tau^{\mathcal{G}}$, we have the necessary condition

$$\forall \tilde{e} \in \tilde{\mathcal{E}} : \kappa^{e} \left[X_{\tilde{e}} = x_{\mathcal{V}}|_{\tilde{\mathcal{E}}} \right] = 1 \,,$$

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. .

$$\forall \tilde{e} \in \tilde{\mathcal{E}} : \left\langle \kappa^{e} \left[X_{\tilde{e} \cap \tilde{\mathcal{V}}} = x_{\mathcal{V}}|_{\tilde{e} \cap \tilde{\mathcal{V}}}, X_{\tilde{e}/\tilde{\mathcal{V}}} \right] \right\rangle [\varnothing] \neq 0 \,.$$

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

$$\kappa^{\{v\}} \left[X_v \right] = \epsilon_{x_v} \left[X_v \right] \,.$$

Therefore, Knowledge Propagation Algorithm 7 can be integrated with backtracking search, with iterations between propagations of knowledge and guessing of additional variables.

5.5.4 Applications

Let us exemplify the usage of Knowledge Propagation on Constraint Satisfaction Problems posed by entailment queries on Markov Networks.

Corollary 5.27 *Let Algorithm 7 be run on the cores $\tau^{\mathcal{G}} \cup \{\beta^{f}\}$ with an arbitrary design of $\tilde{\mathcal{E}}$. Whenever for a formula $f\left[X_{\tilde{\mathcal{V}}}\right]$ and a κ^{e} we have*

$$\left\langle \kappa^{e}, \beta^{f} \right\rangle \left[Y_f = 0 \right] = 0$$

then the Markov Network $\tau^{\mathcal{G}}$ probabilistically entails f. If on the contrary

$$\left\langle \kappa^{e}, \beta^{f} \right\rangle \left[Y_f = 1 \right] = 0$$

then the Markov Network $\tau^{\mathcal{G}}$ probabilistically entails $\neg f$, that is probabilistically contradicts f.

Proof. This follows from Thm. 5.23 ensuring the soundness of Knowledge Propagation and the sufficiency of local entailment. $\blacksquare$

Example 5.28 (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 $\{\beta^{f} \left[Y_f, X_{[d]} \right] : f \in \mathcal{F}\}$ and prepare the knowledge cores

$$\kappa^{\{f\}} \left[Y_f \right] \,.$$

To decide entailment batchwise, Knowledge Propagation Algorithm 7 can be run. Whenever

during the algorithm we have that for a f, then Cor. 5.27 implies that if

$$\kappa^{\{f\}}\left[Y_f\right] = \begin{cases} \epsilon_1\left[Y_f\right] & \text{then, the formula is entailed by } \mathbb{P}^{\mathcal{G}}. \\ \epsilon_0\left[Y_f\right] & \text{then, the formula is contradicted by } \mathbb{P}^{\mathcal{G}}. \\ \mathbb{I}\left[Y_f\right] & \text{then no conclusion can be drawn.} \end{cases}$$

Note, that $\kappa^{\{f\}}\left[Y_f\right] = 0\left[Y_f\right]$ 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. ◊

5.5.5 Mimicking 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

$$\epsilon_1\left[Y_h\right] = \left\langle \{f\left[X_{[d]}\right] : f \in \mathcal{F}\} \cup \{\beta^h\left[Y_h, X_{[d]}\right]\}\right\rangle\left[Y_h\right],$$

that is the inference rule can be performed in when $\mathcal{F}$ are in $\mathcal{U}$.

Example 5.29 Modus Ponens For example, when for two formulas $f, h \in \mathcal{F}$ we have $f \models h$, then when $\kappa^{\{f\}}\left[Y_f\right] = \epsilon_1\left[Y_f\right]$ we have

$$\epsilon_1\left[Y_h\right] = \left\langle (f \Rightarrow h)[Y_f, Y_h], \kappa^{\{f\}}\left[Y_f\right]\right\rangle\left[Y_h\right],$$

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 [RN21]). 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. ◊

Part II

Hybrid Logic Networks

6

Introduction to Part II

Research [...] should bring together logic's aptitude for handling the visible and probability's ability to summarize the invisible. - Judea Pearl [Pea88]

After having explored the tensor formalism in the logical and probabilistic foundations of artificial intelligence in Part I, we now investigate Hybrid Logic Networks, which unify the two approaches.

6.1 Hybrid Logic Networks as Computation-Activation Networks

The unification is along the framework of Computation-Activation Networks (see Def. 2.20), which is common to the tensor network representations in probabilistic and logical models:

- Probability distributions, which have a sufficient statistics can be represented by a computation network of the statistic and an activation tensor. We are especially interested in distributions with elementary activation tensor, which is the case for exponential families.
- Propositional formulas can be computed based on their syntactical decompositions (see Chapter 4). The uniform distributions over their models has the formula itself as a sufficient statistic and can be instantiated by a boolean activation cores.

By allowing for arbitrary elementary activation tensors, we can unify both approaches, when focusing on boolean valued statistics. The resulting Computation-Activation Networks are called Hybrid Logic Networks, since they unify logical and probabilistic models. In Chapter 8 we focus on the representation of such networks and introduce the following (see Figure 6.1):

- **Markov Logic Networks:** By demanding positive and elementary activation tensors, we parametrize positive distributions interpreted as uncertainty-tolerant soft logical knowledge bases.
- **Hard Logic Networks:** By demanding boolean and elementary activation tensors, we parametrize propositional formulas interpreted by hard logical knowledge bases.
- **Hybrid Logic Networks:** Both the soft and the hard parametrizations are unified when admitting generic elementary activation tensors.

In Chapter 9 we characterize these networks as maximum entropy distributions and then focus on the inference properties of such networks.

Regarding the representation of Hybrid Logic Networks, we investigate sparsity mechanisms to result in efficient tensor network representations:

- **Decomposition sparsity:** When the sufficient statistics is decomposable into logical connectives, we find tensor network representation by basis encodings of the component functions. This mechanism has been exploited in Chapter 4 for a single propositional formula and will in Chapter 8 be extended to sets of propositional formulas.
- **Selection sparsity:** We will investigate representation schemes for sets of formulas, which share a common structure, in Chapter 7.

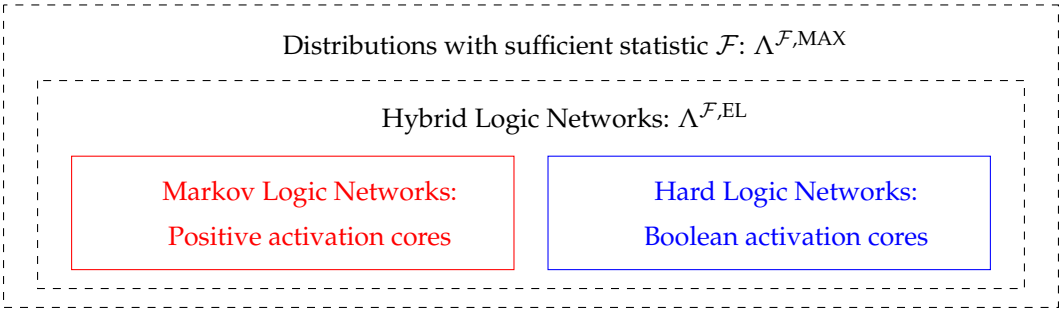


Figure 6.1: Sketch of distributions with sufficient statistics by a set of propositional formulas $\mathcal{F}$. In most generality, any distribution with a sufficient statistics can be represented by an arbitrary activation tensor and is therefore in $\Lambda^{\mathcal{F},\mathrm{MAX}}$. The Hybrid Logic Networks are those with elementary activation tensor, that is the elements of $\Lambda^{\mathcal{F},\mathrm{EL}}$. By further demanding positive activation tensors we characterize the Markov Logic Networks $\mathcal{F}$ and by demanding boolean activation tensors we the Hard Logic Networks $\mathcal{F}$.

- **Polynomial sparsity:** Tensors are decomposed into sums of restricted elementary tensors, which are interpreted by monomials. We will describe such decompositions in Sect. 8.4.3 and relate it later in Chapter 15 to restricted CP decompositions, which we call basis+.

6.2 Extensions towards First-Order Logics

While so-far the focus had been on propositional logic as an explainable framework in machine learning, we show in Chapter 11 extensions towards more expressive first-order logic. We therein encounter two mechanisms in logic introducing tensor structures:

- **Substitution structure:** Assigning objects to variables results in a vector of possible substitutions. When there are formulas with multiple variables, these lists of substitutions carry a tensor structure.
- **Semantic structure:** Mapping all possible worlds defined by d atoms requires 2^d dimensions, which are at best represented by a space of order d tensors. This is analogous to the factored representations of a system.

6.3 Probabilistic Guarantees

Besides that, probabilistic guarantees on the success of the learning problems are derived in Chapter 10. Here we also focus on boolean statistics, which coordinates are Bernoulli variables. Due to their boundedness, they and their averages are sub-Gaussian variables with favorable concentration properties.

6.4 Outline

We first investigate in Chapter 7 efficient representation schemes for propositional formulas. In Chapter 8 and Chapter 9 we develop the hybrid reasoning scheme for logical and probabilistic approaches, which we call Hybrid Logic Networks. We extend the scheme towards first-order logic in Chapter 11 and derive probabilistic guarantees on the success of the learning problems in Chapter 10.

7

Formula Selecting Networks

In this chapter we will investigate efficient schemes to represent collections of propositional formulas with similar structure by a single tensor network. We introduce the mechanism of selection sparsity, which utilizes selection variables to represent common structure of formulas as a tensor network.

7.1 Formula Selecting Maps

In Chapter 4 we introduced with decomposition sparsity a scheme of tensor network representation of a single propositional formula. Let us now extend this scheme to selection sparsity, which provides a sparse representation of sets of propositional formulas. To this end, we first define formula selecting maps and then derive tensor network representation for them.

Definition 7.1 Given a set of p propositional formulas $\{f_l : l \in [p]\}$, the formula selecting map is the map

$$\mathcal{F} : \bigtimes_{k\in[d]} [2] \to \bigtimes_{l\in p} [2]$$

defined for $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ as

$$\mathcal{F}\left(x_{[d]}\right) = \bigtimes_{l\in p} f_l\left[x_{[d]}\right] .$$

A tensor representation of a formula selecting map is provided by the selection encoding (see Def. 0.22)

$$\sigma^{\mathcal{F}}\left[X_{[d]}, L\right]$$

where the selection variable L takes values in $[p]$ and selects specific formulas in the set $\{f_l : l \in [p]\}$. By definition, we have for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ and $l \in [p]$

$$\sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L = l\right] = f_l\left[X_{[d]} = x_0, \ldots, x_{d-1}\right] .$$

This selection encoding is thus the sum

$$\sigma^{\mathcal{F}}\left[X_{[d]}, L\right] = \sum_{l\in[p]} f_l\left[X_{[d]}\right] \otimes \epsilon_l\left[L\right] .$$

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. Central to this sparse representation scheme are basis encodings of the selection encodings $\sigma^{\mathcal{F}}$ (see Sect. 4.3). To this end we understand the tensor $\sigma^{\mathcal{F}}$ as a boolean

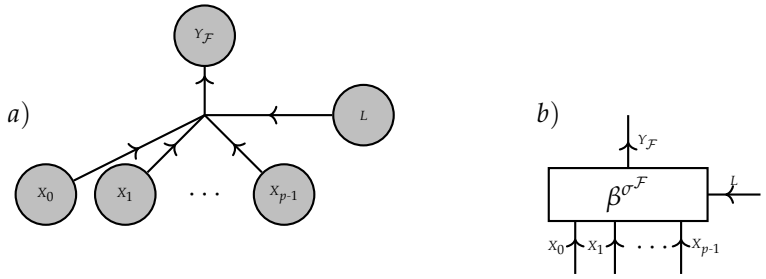


Figure 7.1: Representation of the basis encoding $\beta^{\sigma^{\mathcal{F}}}$ to a selection encoded formula selecting map as an a) Factor of a Graphical Model with a selection variable L and a computed variable $Y_{\mathcal{F}}$. b) Decorating Tensor Core with selection variable corresponding with an additional axis.

function by its coordinate evaluation map and define

$$\beta^{\sigma^{\mathcal{F}}}\left[Y, X_{[d]}, L\right] = \sum_{x_{[d]} \in \times_{k\in[d]}[m_k]} \sum_{l\in[p]} \epsilon_{\sigma^{\mathcal{F}}\left[X_{[d]}=x_{[d]}, L=l\right]}\left[Y\right] \otimes \epsilon_{x_{[d]}}\left[X_{[d]}\right] \otimes \epsilon_l\left[L\right] .$$

These basis encodings represent the selection encoding by

$$\sigma^{\mathcal{F}}\left[X_{[d]}, L\right] = \left\langle \epsilon_1\left[Y\right], \beta^{\sigma^{\mathcal{F}}}\left[Y, X_{[d]}, L\right]\right\rangle \left[X_{[d]}, L\right] .$$

The mechanism of *selection sparsity* now applies this equations to find tensor network decompositions of $\sigma^{\mathcal{F}}$ based on decompositions of $\beta^{\sigma^{\mathcal{F}}}$. We depict the basis encoding $\beta^{\sigma^{\mathcal{F}}}$ of formula selecting maps in Figure 7.1.

7.2 Construction Schemes

Let us now investigate efficient schemes to define sets of formulas to be used in the definition of $\mathcal{F}$. We will motivate the folding of the selection variable into multiple selection variables by compositions of selection maps.

7.2.1 Connective Selecting Maps

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 7.2 Let $\{\circ_0, \ldots, \circ_{p_C-1}\}$ be a set of connectives with d arguments. The associated connective selection map is

$$\mathcal{F}_C : \bigtimes_{k\in[d]} [2] \to \bigtimes_{l\in[p_C]} [2]$$

defined for $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ as

$$\mathcal{F}_C\left(x_{[d]}\right) = \bigtimes_{l\in[p_C]} \circ_l \left[X_{[d]} = x_{[d]}\right] .$$

7.2.2 Variable Selecting Maps

Choices of connectives can be combined with selections of variables assigned building the arguments of a connective. To this end, we introduce variable selecting maps.

Definition 7.3 The selection of one out of p variables in a list $X_{[p]}$ is done by variable selecting maps

$$\mathcal{F}_V : \bigtimes_{l\in[p]} [2] \to \bigtimes_{l\in[p]} [2] .$$

defined as the identity map

$$\mathcal{F}_V \left(X_{[p]} \right) = X_{[p]} .$$

The selection encoding of the variable selecting map is the tensor $\sigma^{\mathcal{F}_V} \left[X_{[p]}, L_V \right]$

$$\sigma^{\mathcal{F}_V} \left[X_{[p]} = x_{[p]}, L_V = l_V \right] = x_{l_V} .$$

Selection encodings of variable selecting maps appear in the literature as multiplex gates (see e.g. Definition 5.3 in [KF09]). The basis encoding of the variable selection map has a decomposition

$$\beta^{\sigma^{\mathcal{F}_V}} \left[Y_V, X_{[p_V]}, L_{\mathcal{F}_V} \right] = \sum_{l_V\in[p_V]} \delta \left[Y_V, X_{l_V} \right] \otimes \mathbb{I} \left[X_{[p_V/\{l_V\}]} \right] \otimes \epsilon_{l_V} \left[L_V \right] .$$

This structure is exploited in the next theorem to derive a tensor network decomposition of $\beta^{\sigma^{\mathcal{F}_V}}$.

Theorem 7.4 (Decomposition of Variable Selecting Maps) *Given a list $X_{[p_V]}$ of variables, we define for each $l_V \in [p_V]$ the tensors*

$$\tau^{l_V} \left[X_{l_V}, L_V \right] = \delta \left[Y_V, X_{l_V} \right] \otimes \epsilon_{l_V} \left[L_V \right] + \mathbb{I} \left[Y_V, X_{l_V} \right] \otimes \left(\mathbb{I} \left[L_V \right] - \epsilon_{l_V} \left[L_V \right] \right) .$$

Then we have (see Figure 7.2)

$$\beta^{\sigma^{\mathcal{F}_V}} \left[Y_V, X_{[p]}, L_V \right] = \left\langle \{ \tau^{l_V} \left[Y_V, X_{l_V}, L_V \right] : l_V \in [p_V] \} \right\rangle \left[Y_V, X_{[p]}, L_V \right] .$$

Proof. We show the equivalence of the tensors on an arbitrary coordinates. For $\tilde{l}_V \in [p_V]$, $Y_V \in [2]$ and $x_{[p_V]} \in \bigtimes_{k\in[p_V]} [2]$ we have

$$\begin{aligned}
&\left\langle \{ \tau^{l_V} \left[Y_V, X_{l_V}, L_V \right] : l_V \in [p_V] \} \right\rangle \left[Y_V = y_V, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V \right] \\
&\quad = \prod_{l_V\in[p_V]} \tau^{l_V} \left[Y_V = y_V, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V \right] \\
&\quad = \tau^{\tilde{l}_V} \left[Y_V = y_V, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V \right] \\
&\quad = \begin{cases} 1 & \text{if} \quad y_V = x_{l_V} \\ 0 & \text{else} \end{cases} \\
&= \beta^{\sigma^{\mathcal{F}_V}} \left[Y_V = y_V, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V \right]
\end{aligned}$$

In the second equality, we used that the tensor τ^{l_V} have coordinates 1 whenever $\tilde{l}_V \neq l_V$. ∎

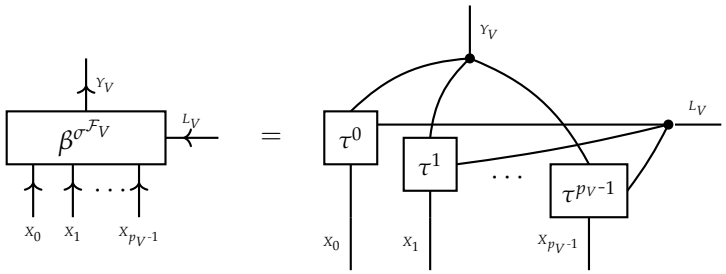


Figure 7.2: Decomposition of the basis encoding of a variable selecting tensor into a network of tensors defined in Thm. 7.4. The decomposition is in a CP format (see Chapter 15).

The decomposition provided by Thm. 7.4 is in a CP format (see Chapter 15). The introduced tensors τ^{l_V} are boolean, but not directed and therefore basis encodings of relations but not functions (see Chapter 14).

7.3 State Selecting Maps

When the variables to be selected are the atomization variables to the same categorical variable (see Sect. 5.4.2), one can avoid the instantiation of all atomization cores and instead represent the variable selecting map using the categorical variable only. To show this we introduce the state selecting map to a categorical variable X.

Definition 7.5 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 map is the map

$$\mathcal{F}_S : [m_S] \to \bigtimes_{k \in [m_S]} [2]$$

defined for $x_S \in [m_S]$ as

$$\mathcal{F}_S(x_S) = \epsilon_{x_S}[L_S] .$$

The selection encoding of the state selecting map coincides with the dirac delta tensor, that is for $x_S \in [m_S]$ and $l_S \in [p_S]$ we have

$$\sigma^{\mathcal{F}_S}[X_S, L_S] = \begin{cases} 1 & \text{if} \quad x = l_S \\ 0 & \text{else} \end{cases} .$$

The relation of the variable selecting map and the state selecting map is shown in the next lemma.

Lemma 7.6 *Let $X_{[p]}$ be a collection of atomization variables to a categorical variable X taking values in $[p]$. Then we have for*

$$\left\langle \beta^e \left[X_{[p]}, X \right], \sigma^{\mathcal{F}_V} \left[X_{[p]}, L \right] \right\rangle [X, L] = \sigma^{\mathcal{F}_S}[X, L] .$$

Proof. For each $x, l \in [p]$ we have that

$$\left\langle \beta^e \left[X_{[p]}, X \right], \sigma^{\mathcal{F}_V} \left[X_{[p]}, L \right] \right\rangle [X = x, L = l] = \left\langle \beta^e \left[X_{[p]}, X = x \right], \sigma^{\mathcal{F}_V} \left[X_{[p]}, L = l \right] \right\rangle [\varnothing]$$

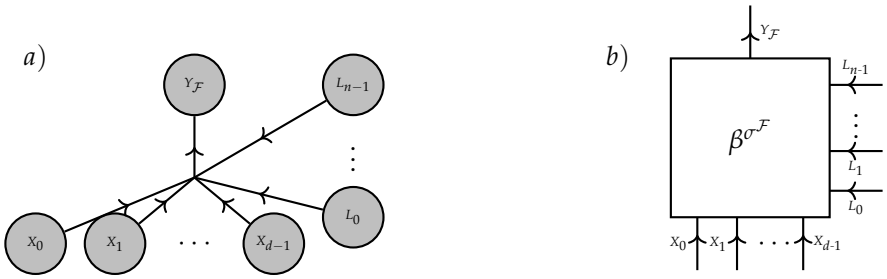


Figure 7.3: Basis encoding of the folded map $\mathcal{F}$.

$$= \begin{cases} 1 & \text{if} \quad x = l \\ 0 & \text{else} \end{cases}$$

which is thus equal to $\sigma^{\mathcal{F}_S}\left[X = x, L = l\right]$. ■

Lem. 7.6 shows that when the variables to be selected are the atomization variables to a categorical variable, the state selecting map can thus be used instead of the variable selecting map. The state selecting map has the advantage, that the instantiation of the tensor $\beta^e\left[X_{[p]}, X\right]$ enforcing the categorical constraint can be avoided.

7.4 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 7.7 Let there be a formula $f_{l_0,\ldots,l_{n-1}}$ for each index tuple in $l_0, \ldots, l_{n-1} \in \bigtimes_{s\in[n]}[p_s]$, where $n, p_0, \ldots, p_{n-1} \in \mathbb{N}$. The folded formula selection map (see Figure 7.3) is the map

$$\mathcal{F} : \bigtimes_{k\in[d]}[2] \to \bigtimes_{l_0\in[p_0]} \cdots \bigtimes_{l_{n-1}\in[p_{n-1}]}[2]$$

defined as

$$\mathcal{F}\left(x_{[d]}\right) = \left(f_{l_{[n]}}\left[X_{[d]} = x_{[d]}\right]\right)_{l_{[n]}\in\bigtimes_{s\in[n]}[p_s]}.$$

Folding the selection variable into multiple selection variables is especially useful to find efficient decomposition schemes of the formula selecting maps. In the reminder of this section we will provide an example, where each selection variable is constructed to parameterize a local change to the formula with the result that the basis encoding of the global formula selecting map decomposes into local formula selecting maps.

7.4.1 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 (see Def. 7.2) with variable selection maps (see Def. 7.3) for each argument a formula selecting neuron.

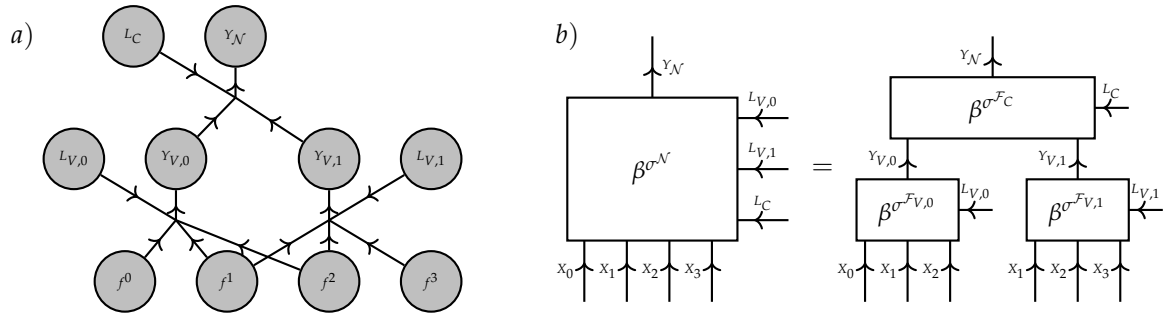


Figure 7.4: Example of a logical neuron $\mathcal{N}$ of order $n = 2$. a) Selection and categorical variables and their interdependencies visualized in a hypergraph. b) Basis encoding of the logical neuron and tensor network decomposition into variable selecting and connective selecting tensors.

Definition 7.8 Given an order $n \in \mathbb{N}$ let there be a connective selector $L_\circ$ selecting connectives of order n and let $\mathcal{F}_{V,0}, \ldots, \mathcal{F}_{V,n-1}$ be a collection of variable selectors. The corresponding logical neuron is the map

$$\mathcal{N} : \bigtimes_{k\in[d]} [2] \to \bigtimes_{l_C\in[p_C]} \bigtimes_{l_0\in[p_0]} \cdots \bigtimes_{l_{n-1}\in[p_{n-1}]} [2]$$

defined for $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ by

$$\mathcal{N}\left(x_{[d]}\right) = \left(\circ_{l_C}(X_{l_0},\ldots,X_{l_{n-1}})\right)_{l_C\in[p_C],l_0\in[p_0],\ldots,l_{n-1}\in[p_{n-1}]}$$

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 7.9 *Decomposition of formula selecting neurons Let $\mathcal{N}$ a logical neuron, defined for a connective selector $L_\circ$ and variable selectors $\mathcal{F}_{V,0}, \ldots, \mathcal{F}_{V,n-1}$. Then we have (see Figure 7.4 for the example of $n = 2$):*

$$\begin{aligned}
&\beta^{\sigma^{\mathcal{N}}}\left[Y_{\mathcal{N}}, X_{[d]}, L_C, L_{V,0}, \ldots, L_{V,n-1}\right] \\
&\quad = \langle \{\beta^{\sigma^{\mathcal{F}_C}}\left[Y_{\mathcal{N}}, Y_{V,0}, \ldots, Y_{V,n-1}\right], \\
&\qquad \beta^{\sigma^{\mathcal{F}_{V,0}}}\left[Y_{V,0}, X_{[d]}, L_{V,0}\right], \ldots, \beta^{\sigma^{\mathcal{F}_{V,n-1}}}\left[Y_{V,n-1}, X_{[d]}, L_{V,n-1}\right]\}\rangle \left[Y_{\mathcal{N}}, X_{[d]}, L_C, L_{V,0}, \ldots, L_{V,n-1}\right].
\end{aligned}$$

Proof. By composition Thm. 14.12. ∎

7.4.2 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 7.10 An architecture graph $\mathcal{G}^{\mathcal{A}} = (\mathcal{V}^{\mathcal{A}}, \mathcal{E}^{\mathcal{A}})$ is an acyclic directed hypergraph

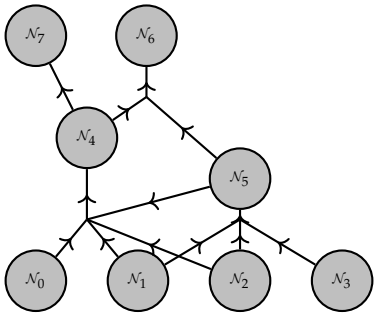


Figure 7.5: Example of an architecture graph $\mathcal{G}^{\mathcal{A}}$ with input neurons $\mathcal{A}^{\text{in}} = \{\mathcal{N}_0, \mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3\}$ and output neurons $\mathcal{A}^{\text{out}} = \{\mathcal{N}_6, \mathcal{N}_7\}$

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 7.5 for an example).

Given an architecture graph $\mathcal{G}^{\mathcal{A}} = (\mathcal{V}^{\mathcal{A}}, \mathcal{E}^{\mathcal{A}})$, a *formula selecting neural network* $\sigma^{\mathcal{A}}$ is a tensor network of logical neurons at each $\mathcal{N} \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}}$, such that each neuron depends on variables $X_{\text{Pa}(\mathcal{N})}$ and on selection variables $L_{\mathcal{N}}$. The collection of all selection variable is notated by $L_{\mathcal{A}}$.

The activation tensor of each neuron $\mathcal{N} \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}}$ is

$$\sigma^{\mathcal{N}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \left\{ \beta^{\sigma^{\tilde{\mathcal{N}}}} \,:\, \tilde{\mathcal{N}} \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}} \right\} \cup \left\{ \epsilon_1\left[Y_{\mathcal{N}}\right] \right\} \right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] .$$

The activation tensor of the formula selecting neural network is the contraction

$$\sigma^{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \left\{ \beta^{\sigma^{\mathcal{N}}} \left[Y_{\mathcal{N}}, X_{\text{Pa}(\mathcal{N})}, L_{\mathcal{A}}\right] \,:\, \mathcal{N} \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}} \right\} \cup \left\{ \epsilon_1\left[Y_{\mathcal{N}}\right] \,:\, \mathcal{N} \in \mathcal{A}^{\text{out}} \right\} \right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] .$$

The expressivity of a formula selecting neural network $\sigma^{\mathcal{A}}$ is the formula set

$$\mathcal{F}_{\mathcal{A}} = \left\{ \sigma^{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} = l_{\mathcal{A}}\right] \,:\, l_{\mathcal{A}} \in \bigtimes_{s \in [n]} [p_s] \right\} .$$

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 $\mathcal{N} \in \mathcal{V}^{\mathcal{A}} / \mathcal{A}^{\text{in}}$ by an output variable $Y_{\mathcal{N}}$ and further add selection variables $L_{\mathcal{N}}$ to the directed edges, that is to each directed hyperedge $(\{\mathcal{N}\}, \text{Pa}(\mathcal{N})) \in \mathcal{E}^{\mathcal{A}}$ we construct a directed hyperedge $(\{Y_{\mathcal{N}}\}, X_{\text{Pa}(\mathcal{N})} \cup L_{\mathcal{N}})$.

Theorem 7.11 *Given fixed selection variables $L_{\mathcal{A}}$, the formula selecting neural network is the conjunction of output neurons, that is*

$$\sigma^{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \left\{ \sigma^{\mathcal{N}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] \,:\, \mathcal{N} \in \mathcal{A}^{\text{out}} \right\} \right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] .$$

Proof. By effective calculus (see Thm. 14.36), we have

$$\left\langle \beta^{\sigma^{\wedge}} \left[X_{\wedge}, X_{[d]}\right], \epsilon_1\left[X_{\wedge}\right] \right\rangle \left[X_{[d]}\right] = \bigotimes_{k \in [d]} \epsilon_1\left[X_k\right]$$

and thus

$$\sigma^{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \left\{\beta^{\sigma^{\mathcal{N}}} : \mathcal{N} \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}\right\} \cup \left\{\beta^{\sigma^{\wedge}}\left[X_{\wedge}, Y_{\mathcal{N}} : \mathcal{N} \in \mathcal{A}^{\text{out}}\right], \epsilon_1\left[X_{\wedge}\right]\right\}\right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] .$$

■

By the commutation of contractions, we can further use Thm. 7.9 to decompose each tensor $\beta^{\sigma^{\mathcal{N}}}$ into connective and variable selecting components to get a sparse representation of a formula selecting neural network $\sigma^{\mathcal{A}}$.

7.5 Application of Formula Selecting Networks

Formula selecting networks are efficient in the simultaneous representation of multiple formulas with similar structure.

7.5.1 Efficient Representation of Formulas

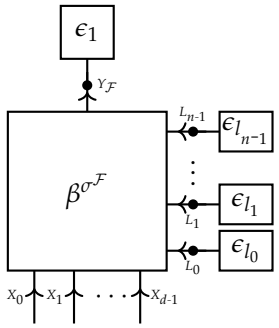
Formula Selecting Neural Networks are means to represent exponentially many formulas with linear (in sum of candidates list lengths) storage. Their contraction with probability tensor networks, is thus a batchwise evaluation of exponentially many formulas. This is possible due to redundancies in logical 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

$$f_{l_0,\ldots,l_{n-1}}[X_{[d]}] = \sigma^{\mathcal{F}}\left[X_{[d]}, L = l_0, \ldots, l_{n-1}\right] .$$

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 contraction of parameter cores θ with efficient network decompositions along the selection variables.

In Gibbs Sampling (Algorithm 2), one can use the energy-based approach to queries Thm. 3.3, and contract basis vectors on all but one selection variables.

7.5.2 Batch Contraction of Parametrized Formulas

Given a set $\mathcal{F}$ of formulas, we build a formula selecting network parametrizing the formulas. The contraction

$$\left\langle \tau^{\mathcal{G}}, \mathcal{F} \right\rangle \left[L_{[n]}\right]$$

is a tensor containing the contractions of the formulas $f_{l_{[n]}}$ with an arbitrary tensor network $\tau^{\mathcal{G}}$ as

$$\left\langle \tau^{\mathcal{G}}, f_{l_{[n]}} \right\rangle [\varnothing] = \left\langle \tau^{\mathcal{G}}, \mathcal{F} \right\rangle \left[L_{[n]} = l_{[n]} \right] .$$

7.5.3 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 7.12 (Average satisfaction of formulas) The average of the formula satisfactions in $\mathcal{F}$ given a probability tensor $\mathbb{P}$ is

$$\frac{1}{\prod_{s\in[n]} p_s} \cdot \left\langle \mathbb{P}, \sigma^{\mathcal{F}} \right\rangle [\varnothing] .$$

$\Diamond$

Example 7.13 (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

$$\left\langle \mathcal{KB}, \sigma^{\mathcal{F}} \right\rangle [\varnothing] = 0 .$$

When the formulas are representable in a folded scheme, we find tensor network decompositions of $\mathcal{F}$ and exploit them along efficient representations of $\mathcal{KB}$ in an efficient calculation of $\left\langle \mathcal{KB}, \sigma^{\mathcal{F}} \right\rangle [\varnothing]$. This is further equal to

$$\mathcal{KB} \models \neg \left(\bigvee_{f\in\mathcal{F}} f \right) .$$

$\Diamond$

7.6 Examples of Formula Selecting Neural Networks

7.6.1 Correlation

For example (see Figure 7.6) 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

$$\tau\left[L_{V,0}, L_{V,1}\right] = \left\langle \sigma^{\mathcal{A}} \right\rangle \left[L_{V,0}, L_{V,1}\right]$$

to read of covariances as

$$\mathrm{Cov}(X_k, X_l) = \tau\left[L_{V,0} = k, L_{V,1} = l\right] - \tau\left[L_{V,0} = k, L_{V,1} = k\right] \cdot \tau\left[L_{V,0} = l, L_{V,1} = l\right] .$$

7.6.2 Conjunctive and Disjunctive Normal Forms

We can represent any propositional knowledge base by the following scheme: Literal selecting neurons are logical neurons with connective identity/negation (selecting positive/negative literal) and selecting neurons select for each an atom. The single output neuron represents the disjunction, respectively the conjunction, combining the literal selecting neurons. The number

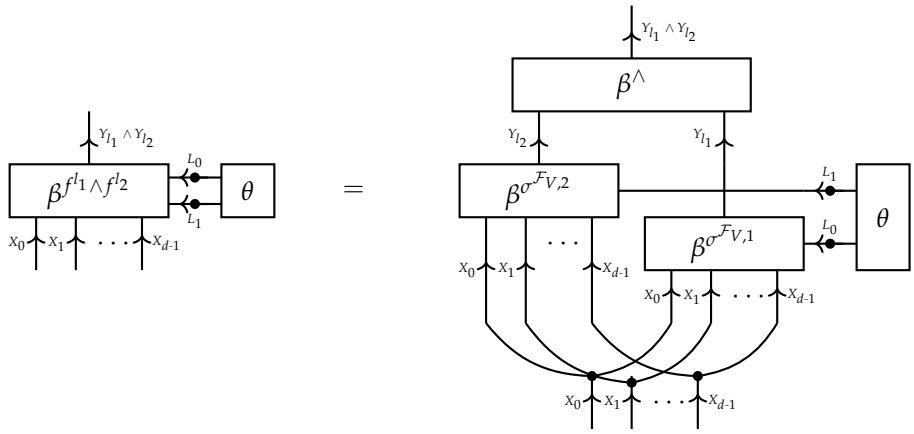


Figure 7.6: Superposition of the encoded formulas $f^{l_1} \wedge f^{l_2}$ with weight $\theta\,[L_1 = l_1, L_2 = l_2]$

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). This architecture will be discussed in more detail in Chapter 16 as CP selecting networks. The parameter core is in the basis CP format and each slice selects a clause of the knowledge base. In combination with polynomial decompositions, which will be provided in Chapter 8, one can exploit this architecture to find sparse formula decompositions.

Remark 7.14 (Minterms and Maxterms) All minterms and maxterms can be represented by a two layer selection tensor networks without variable selection in two layers. The bottom layer has an $\neg$/Id connective selection neuron to each atom and the upper layer consists of a single $d$ary conjunction. ♢

7.7 Extension to Variables of Larger Dimension

While we here restricted on boolean variables, formula selecting networks can be extended to variables of larger cardinality.

- Connective selecting tensors: Can encode arbitrary functions h_l of discrete variables, but need $X_{\mathcal{F}_C}$ to be an enumeration of the states, in particular to be of dimension

$$m_{\mathcal{F}_C} = \left| \cup_{l \in [p]} \mathrm{im}\,(h_l) \right| .$$

- Variable selecting tensors can be understood as specific cases of connective selecting tensors and can thus also be generalized in a straight forward manner by

$$m_{\mathcal{F}_C} = \left| \cup_{l \in [p]} \mathrm{im}\,(h_l) \right| .$$

- State selecting tensors are directly defined for larger dimensions

An example of such a more generic usage is a discretization scheme for continuous neurons.

Example 7.15 (Discretization of a continuous neuron) Let there be a neuron by a map of weight vectors and input vectors to $\mathbb{R}$, that is

$$\mathcal{N}(w, x) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} .$$

We restrict the weights to a subset $\mathcal{U}^{weight} \subset \mathbb{R}^d$ and the input vectors to $\mathcal{U}^x \subset \mathbb{R}^d$, If follows that

$$\left|\text{im}\left(\mathcal{N}|_{\mathcal{U}^{weight}\times\mathcal{U}^x}\right)\right| \leq \left|\mathcal{U}^{weight}\right| \cdot |\mathcal{U}^x| \ .$$

To discretize the neuron, we use the subset encoding scheme of Def. 14.1 and define enumeration variables O_{weight}, O_x and $O_{\mathcal{N}}$ enumerating $\mathcal{U}^{weight}$, $\mathcal{U}^x$ and im $\left(\mathcal{N}|_{\mathcal{U}^{weight}\times\mathcal{U}^x}\right)$, which are accompanied by respective index interpretation functions. Then the basis encoding of the discretized neuron is

$$\beta^{\sigma^{\mathcal{N}}}\left[O_{\mathcal{N}}, O_{weight}, O_x\right] \ . = \sum_{o_{weight}\in[r_{weight}], o_x\in[r_x]} \epsilon_{I^{-1}_{\mathcal{N}(I_{weight}(o_{weight}),I_x(o_x))}(\mathcal{N})} O_{\mathcal{N}} \otimes \epsilon_{o_{weight}}\left[O_{weight}\right] \otimes \epsilon_{o_x}\left[O_x\right] \ .$$

If the neuron is of the form

$$\mathcal{N}(w,x) = \psi(\sum_i w_i \cdot x_i)$$

a decomposition into multiplication at each coordinate and summation of the results, with basis encodings for each, can be done. Here the index interpretation variables are split into a selection enumerated by i and each variable gets assigned to single cores in the decomposition. $\lozenge$

8

Hybrid Logic Representation

Hybrid Logic Networks are graphical models with an interpretation by propositional logics. Since they unify probabilistic and logical reasoning, we call them hybrid. In this chapter we investigate their representation based on tensor networks, while the reasoning mechanisms are discussed in Chapter 9. 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. We then investigate the corresponding mean parameter polytopes of Hybrid Logic Networks and show expressivity results.

8.1 Markov Logic Networks

Markov Logic Networks exploit the efficiency and interpretability of logical calculus as well as the expressivity of graphical models.

8.1.1 Markov Logic Networks as Exponential Families

We introduce Markov Logic Networks in the formalism of exponential families (see Sect. 2.3).

Definition 8.1 (Markov Logic Network) 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 coordinate s_l is contained in $\{0,1\}$, each is a propositional formulas (see Def. 4.2). Thus, any exponential family of distributions of $\bigtimes_{k\in[d]}[2]$, such that im $(s_l) \subset \{0,1\}$ for all $l \in [p]$ is a set of Markov Logic Networks with fixed formulas.

Remark 8.2 (Alternative Definitions) We here defined Markov Logic Networks on propositional logic, while originally they are defined in the more expressive first-order logic [RD06]. The relation of both frameworks will be discussed further in Chapter 11. ♢

8.1.2 Tensor Network Representation

Based on the previous discussion on the representation of exponential families by tensor networks in Sect. 2.3 we now derive a representation for Markov Logic Networks.

8.1.2.1 Basis Encodings for Distributions

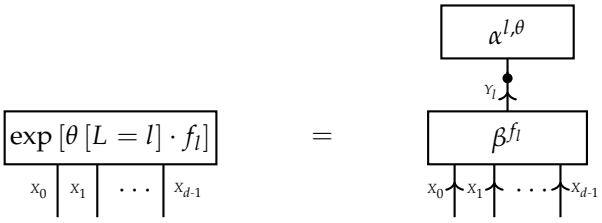


Figure 8.1: Factor of a Markov Logic Network to a formula f_l, represented as the contraction of a computation core β^{f_l} and an activation core $\alpha^{l,\theta}$. While the computation core β^{f_l} prepares based on basis calculus a categorical variable representing the value of the statistic formula f_l dependent on assignments to the distributed variables, the activation core multiplies an exponential weight to coordinates satisfying the formula.

Theorem 8.3 (Basis encodings for Markov Logic Networks) *A Markov Logic Network to a set of formulas $\mathcal{F} = \{f_l : l \in [p]\}$ is represented as*

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \{\beta^{f_l}\left[Y_l, X_{[d]}\right] : l \in [p]\} \cup \{\alpha^{l,\theta}[Y_l] : l \in [p]\} \right\rangle \left[X_{[d]}|\varnothing\right]$$

where we denote for each $l \in [p]$ an activation core

$$\alpha^{l,\theta}[Y_l = y_l] = \begin{cases} 1 & \text{if} \quad y_l = 0 \\ \exp[\theta[L = l]] & \text{if} \quad y_l = 1 \end{cases}.$$

Proof. Markov Logic Networks are exponential families, which base measure is trivial and which statistic consist of boolean features. We apply the tensor network decomposition of more generic exponential families Thm. 2.24 to this case and get

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \{\mathbb{I}\left[X_{[d]}\right]\} \cup \{\beta^{s_l}\left[Y_l, X_{[d]}\right] : l \in [p]\} \cup \{\alpha^{l,\theta}[Y_l] : l \in [p]\} \right\rangle \left[X_{[d]}|\varnothing\right].$$

While the base measure tensor is trivial, it can be ignored in the contraction. Since the image of each feature f_l is in $[2]$, we choose the index interpretation function by the identity $I : [2] \to \{0,1\}$ and get

$$\begin{aligned} \alpha^{l,\theta}[Y_l = y_l] &= \exp[\theta[L = l] \cdot I_l(y_l)] = \exp[\theta[L = l] \cdot y_l] \\ &= \begin{cases} 1 & \text{if} \quad y_l = 0 \\ \exp[\theta[L = l]] & \text{if} \quad y_l = 1 \end{cases}. \end{aligned}$$

■

Thm. 8.3 provides a decomposition of Markov Logic Networks by a tensor network of computation cores β^{f_l} and accompanying activation cores $\alpha^{l,\theta}$. Since the head variable Y_l appears exclusively in these pairs, we can contract each computation core with the corresponding activation core to get a factor, see Figure 8.1. With this we get the decomposition

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \{\exp\left[\theta[L = l] \cdot f_l\left[X_{[d]}\right]\right] : l \in [p]\} \right\rangle \left[X_{[d]}|\varnothing\right].$$

More precisely, this transformation of the decomposition holds by Thm. 17.1 to be shown in Chapter 17, stating that the contraction of computation and activation cores can be performed before the global contraction of the result.

While in the decomposition of Thm. 8.3 the basis encodings of the features carry all distributed variables $X_{[d]}$, we now seek towards sparser decompositions. To each $l \in [p]$ we denote by $\mathcal{V}^l$ the

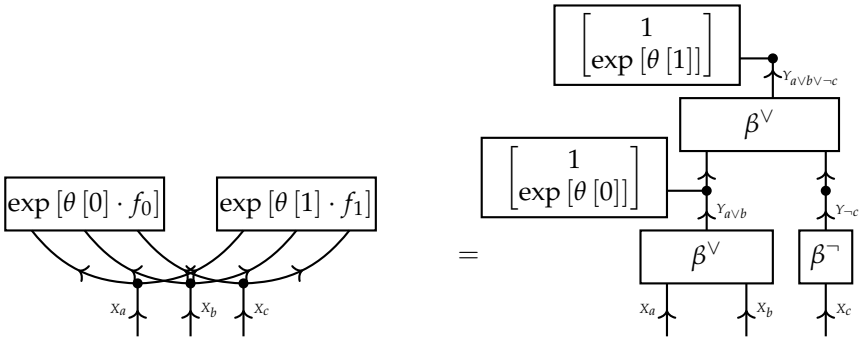


Figure 8.2: Example of a decomposed Markov Network representation of a Markov Logic Network with formulas $\{f_0 = a \vee b, f_1 = a \vee b \vee \neg c\}$. Since both formulas share the subformula $a \vee b$, their contracted factors have a representation by a connected tensor network.

maximal subset of $[d]$ such that there is a reduced function $\tilde{f}_l : \bigtimes_{k \in \mathcal{V}^l} [m_k] \to [2]$ with

$$f_l \left[X_{[d]} \right] = \langle \tilde{f}_l [X_{\mathcal{V}^l}] \rangle \left[X_{[d]} \right] .$$

We often account for such situations of sparse formulas, when f_l has a syntactical decomposition involving only the atomic variables $\mathcal{V}^l$. As a consequence we have

$$\beta^{f_l} \left[X_{[d]} \right] = \beta^{\tilde{f}_l} \left[X_{\mathcal{V}^l} \right]$$

and

$$\mathbb{P}^{\mathcal{F},\theta} [X_{[d]}] = \left\langle \{ \exp \left[\theta \left[L = l \right] \cdot \tilde{f}_l [X_{\mathcal{V}^l}] \right] \, : \, l \in [p] \} \right\rangle \left[X_{[d]} | \varnothing \right] .$$

Thus, any Markov Logic Network has a sparse representation by a markov network on the graph

$$\mathcal{G}^{\mathcal{F}} = ([d], \{ \mathcal{V}^l \, : \, l \in [p] \}) .$$

This sparsity inducing mechanism is analogous to the decomposition of probability distributions based on conditional independence assumptions, when understanding each formula in the Markov Logic Network as an introduced dependency among the affected variables $\mathcal{V}^l$.

A further sparsity introducing mechanism is through exploiting redundancy in the computation of f_l, when a decomposition of the feature is known. For the propositional formulas f_l this amounts to a syntactic representation of the formula as a composition of logical connectives is available (see Figure 8.2). In this case, we exploit the representation by tensor networks of the basis encodings (shown as Thm. 14.12 in Chapter 14) Note, that this decomposition scheme introduces further auxiliary variables Y with deterministic dependence on the distributed variables $X_{[d]}$. Such variables are often referred to as hidden.

We can further exploit common syntactical structure in the formulas $f_l \in \mathcal{F}$ to reduce the number of basis encodings of connectives. This is the case, when the syntax graph of two or more formulas share a subgraph. In that case, the respective syntax graph needs to be represented only once an can be incorporated into the decomposition of all formulas, which share this subgraph. For an example see Figure 8.2, where the syntactical representation of the formula f_0 is a subgraph of the syntactical representation of f_1.

To summarize, our sparse representations of Markov Logic Networks implement the two sparsity mechanisms described in Part I, originating from graphical models and propositional syntax:

- **Independence mechanism** (see Sect. 2.2.1): Formulas often depend only on subsets of atoms, and therefore the variable Y_f is independent on some atomic variables X_k. This exploits the main sparsity mechanism in graphical models, where factors in sparse representations depend only on a subset of variables.

- **Computation mechanism** (see Sect. 2.2.2): When the features of an exponential family are compositions of smaller formulas, the computation core is decomposed into a tensor network of their basis encodings. This can be regarded as the main sparsity mechanism of propositional logics, where syntactical decompositions of formulas are exploited. Further, when the structure of the smaller formulas is shared among different features, the respective basis encodings need to be instantiated only once.

8.1.2.2 Selection Encodings for Energy Tensors

As for generic exponential families, we can represent Markov Logic Networks in terms of their energy tensors

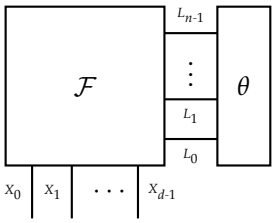
$$\phi^{\mathcal{F},\theta}\left[X_{[d]}\right] = \sum_{l\in[p]} \theta\left[L=l\right]\cdot f_l\left[X_{[d]}\right] = \left\langle \sigma^{\mathcal{F}}\left[X_{[d]},L\right],\theta\left[L\right]\right\rangle\left[X_{[d]}\right] .$$

The energy tensor provides an exponential representation of the distribution by

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \exp\left[\phi^{\mathcal{F},\theta}\right]\right\rangle\left[X_{[d]}|\varnothing\right] .$$

In case of a common structure of the formulas in a Markov Logic Network, formula selecting networks (see Chapter 7) can be applied to represent their energies. We represent the superposition of formulas as a contraction with a 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:

$$\sum_{l_{[n]}\in\bigtimes_{s\in[n]}[p_s]} \theta\left[L_{[n]}=l_{[n]}\right] f_{l_{[n]}} \quad =$$



If the number of atoms and parameters gets large, it is important to represent the tensor $f_{l_0,\ldots,l_{n-1}}$ efficiently in tensor network format and avoid contractions. Such efficient decomposition schemes can be found using decomposition sparsity, see Sect. 8.4.2. 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 10) 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 normalize the energy tensor, a task for which basis 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.

8.1.3 Expressivity

Based on Markov Logic Networks containing only maxterms and minterms (see Def. 4.11), we now show that any positive probability distribution has a representation by a Markov Logic Network.

Theorem 8.4 *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. 4.11)

$$\mathcal{F}_\wedge = \{Z^\wedge_{x_0,\dots,x_{d-1}} : x_0,\dots,x_{d-1} \in \bigtimes_{k\in[d]} [2]\}$$

with parameters

$$\theta\left[L_0 = x_0,\dots,L_{d-1} = x_{d-1}\right] = \ln \mathbb{P}\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}\right]$$

coincides with $\mathbb{P}\left[X_{[d]}\right]$.

Further, the Markov Logic Network of maxterms

$$\mathcal{F}_\vee = \{Z^\vee_{x_0,\dots,x_{d-1}} : x_0,\dots,x_{d-1} \in \bigtimes_{k\in[d]} [2]\}$$

with parameters

$$\theta\left[L_0 = x_0,\dots,L_{d-1} = x_{d-1}\right] = -\ln \mathbb{P}\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}\right]$$

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

$$\phi^{\mathcal{F},\theta} = \ln \mathbb{P}\left[X_{[d]}\right]$$

and therefore

$$\mathbb{P}^{\mathcal{F},\theta}[X_{[d]}] = \left\langle \exp\left[\phi^{\mathcal{F},\theta}\right]\right\rangle \left[X_{[d]}|\varnothing\right] = \left\langle \exp\left[\phi^{\mathcal{F},\theta}\right]\right\rangle \left[X_{[d]}\right] = \mathbb{P}\left[X_{[d]}\right] .$$

In the case of minterms, we notice that for any $x_0,\dots,x_{d-1} \in \bigtimes_{k\in[d]}[2]$

$$Z^\wedge_{x_0,\dots,x_{d-1}}[X_{[d]}] = \epsilon_{x_0,\dots,x_{d-1}}\left[X_{[d]}\right]$$

and thus with the weights in the claim

$$\sum_{x_0,\dots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} \left(\ln \mathbb{P}\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}\right]\right) \cdot Z^\wedge_{x_0,\dots,x_{d-1}}[X_{[d]}] = \ln \mathbb{P}\left[X_{[d]}\right] .$$

For the maxterms we have analogously

$$Z^\vee_{x_0,\dots,x_{d-1}}[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - \epsilon_{x_0,\dots,x_{d-1}}\left[X_{[d]}\right]$$

and thus that the maximal clauses coincide with the one-hot encodings of respective states. We thus have

$$\sum_{x_0,\dots,x_{d-1}\in\bigtimes_{k\in[d]}[2]} \left(-\ln \mathbb{P}\left[X_0 = x_0,\dots,X_{d-1} = x_{d-1}\right]\right) \cdot Z^\vee_{x_0,\dots,x_{d-1}}[X_{[d]}]$$

$$
\begin{aligned}
&= \left(\sum_{\mathcal{V}_0 \subset [d]} (-\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]) \cdot \mathbb{I}\left[X_{[d]}\right] \right) \\
&\quad + \left(\sum_{\mathcal{V}_0 \subset [d]} (\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]) \cdot \epsilon_{x_0,\ldots,x_{d-1}} \left[X_{[d]}\right] \right) \\
&= \ln \mathbb{P}\left[X_{[d]}\right] + \lambda \cdot \mathbb{I}\left[X_{[d]}\right] ,
\end{aligned}
$$

where λ is a constant. ∎

We note, that there are 2^d maxterms and 2^d minterms, which would have to be instantiated by basis encodings to get a tensor network decomposition. This large number of features originates from the generality of the representation scheme. As a fundamental tradeoff, efficient representations come at the expense of a smaller expressivity of the representation scheme.

Thm. 8.4 is the analogue in Markov Logic to Thm. 4.14, 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.

Sparser representation formats based on the same principle as used in Thm. 8.4 can be constructed to represent markov networks by Markov Logic Networks. Here, we can separately instantiate the factors by combinations of terms and clauses only involving the variables containted in the factor. The reason for this is, that minterms form a partition statistics, which will be discussed in more detail in Chapter 14, see Def. 14.32.

8.1.4 Examples

Let us now provide examples of Markov Logic Networks.

8.1.4.1 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 8.5 *Let $\mathbb{P}\left[X_{[d]}\right]$ be a positive probability distribution, such that atomic formulas are independent from each other. Then $\mathbb{P}\left[X_{[d]}\right]$ is the Markov Logic Network of atomic formulas*

$$\mathcal{F}_{[d]} = \{f^k : k \in [d]\}$$

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 Thm. 2.12 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

$$\mathbb{P}^{(\{f^k\},\theta[L=k])}[X_k] = \mathbb{P}^k[X_k]$$

and therefore with the parameter vector of dimension $p = d$ defined as

$$\theta[L] = \sum_{k\in[d]} \theta[L=k] \cdot \epsilon_k[L]$$

we have

$$\begin{aligned}\mathbb{P}^{(\{f^k:k\in[d]\},\theta)}[X_{[d]}] &= \bigotimes_{k\in[d]} \mathbb{P}^{(\{f^k\},\theta[L=k])}[X_k] \\ &= \bigotimes_{k\in[d]} \mathbb{P}^k[X_k] \\ &= \mathbb{P}\left[X_{[d]}\right] .\end{aligned}$$

■

In Thm. 8.5 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. 8.2).

All atomic formulas can be selected by a single variable selecting tensor, that is

$$\phi^{(\{f^k:k\in[d]\},\theta)}[X_{[d]}] = \left\langle \mathcal{F}_V\left(X_{[d]}, L\right), \theta[L]\right\rangle \left[X_{[d]}\right] .$$

In case of negative coordinates $\theta[L=k]$ it is convenient to replace f^k by $\neg f^k$, in order to facilitate the interpretation. The probability distribution is left invariant, when also replacing $\theta[L=k]$ by $-\theta[L=k]$.

8.1.4.2 Boltzmann Machines

A Boltzmann machine is a distribution of boolean variables $X_{[d]}$ depending on a weight matrix

$$W[L_{V,0}, L_{V,1}] \in \mathbb{R}^d \otimes \mathbb{R}^d \quad \text{and a bias vector} \quad b[L_{V,0}] \in \mathbb{R}^d .$$

Its distribution is

$$\mathbb{P}^{W,b}\left[X_{[d]}\right] = \left\langle \exp\left[\phi^{W,b}[X_{[d]}]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$$

where its energy tensor is

$$\phi^{W,b}[X_{[d]} = x_{[d]}] = \sum_{k,l\in[d]} W[L_{V,0}=k, L_{V,1}=l] \cdot x_k \cdot x_l + \sum_{k\in[d]} b[L_{V,0}=k] \cdot x_k .$$

We notice that this tensor 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 $d^2 + d$. For an sparse representation of this energy tensor using a formula selecting network, see Example 8.14. Therefore, Boltzmann machines are specific Markov Logic Networks with the statistic being biimplications between atoms and atoms itself. Generic Markov Logic Networks are more expressive than Boltzmann machines, by the flexibility to create further features by propositional formulas.

8.2 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 now discuss their hard logic counterparts, where worlds not satisfying activated formulas have zero probability.

The probability function of Markov Logic Networks with positive weights mimics 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 normalized 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 normalized knowledge bases.

Lemma 8.6 *Given a formula f we have in the limit of $\beta \to \infty$ the coordinatewise convergence*

$$\left\langle \exp\left[\beta \cdot \theta \cdot f\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right] \to \begin{cases} \langle f\rangle \left[X_{[d]}|\varnothing\right] & \text{if} \quad f \text{ satisfiable and } \theta > 0 \\ \langle \mathbb{I}\rangle \left[X_{[d]}|\varnothing\right] & \text{if} \quad f \text{ or } \neg f \text{ not satisfiable or } \theta = 0 \\ \langle \neg f\rangle \left[X_{[d]}|\varnothing\right] & \text{if} \quad \neg f \text{ satisfiable and } \theta < 0 \end{cases} .$$

Proof. The statement is trivial for the case of f or $\neg f$ being satisfiable, or $\theta = 0$, since then for all $\beta \in \mathbb{R}$

$$\left\langle \exp\left[\beta \cdot \theta \cdot f\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right] = \langle \mathbb{I}\rangle \left[X_{[d]}|\varnothing\right] .$$

In the case of f being satisfiable and $\theta > 0$ we have

$$\left\langle \exp\left[\beta \cdot \theta \cdot f\left[X_{[d]}\right]\right]\right\rangle [\varnothing] = \left(\prod_{k\in[d]} m_k\right) - \langle f\rangle [\varnothing] + \langle f\rangle [\varnothing] \cdot \exp[\beta \cdot \theta]$$

and therefore for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ with $f\left[X_{[d]} = x_{[d]}\right] = 1$

$$\begin{aligned} \langle \exp[\beta \cdot \theta \cdot f]\rangle \left[X_{[d]} = x_{[d]}|\varnothing\right] &= \frac{\exp[\beta \cdot \theta]}{\left(\prod_{k\in[d]} m_k\right) - \langle f\rangle [\varnothing] + \langle f\rangle [\varnothing] \cdot \exp[\beta \cdot \theta]} \\ &\to \frac{1}{\langle f\rangle [\varnothing]} = \langle f\rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1}|\varnothing] . \end{aligned}$$

For any $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ with $f\left[X_{[d]} = x_{[d]}\right] = 0$ we have on the other side

$$\begin{aligned} \langle \exp[\beta \cdot \theta \cdot f]\rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1}|\varnothing] &= \frac{1}{\left(\prod_{k\in[d]} m_k\right) - \langle f\rangle [\varnothing] + \langle f\rangle [\varnothing] \cdot \exp[\beta \cdot \theta]} \\ &\to 0 = \langle f\rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1}|\varnothing] . \end{aligned}$$

This shows the statement in the case of f being satisfiable and $\theta > 0$. The case of $\neg f$ being satisfiable and $\theta < 0$ follows from a similar argument. ∎

Lem. 8.6 only characterizes the annealing limits in exponential families with statistics by single

formulas. If there are multiple formulas the limiting distribution gets more involved. We will characterize the generic limiting distributions later in Sect. 8.5.6 based on face measures in corresponding mean parameter polytopes.

We can represent the local limiting distribution in the three cases by an activation core contracted to the computation core $\beta^f\left[Y_f, X_{[d]}\right]$. When choosing

$$\kappa\left[Y_f\right] = \begin{cases} \epsilon_1\left[Y_f\right] & \text{if} \quad f \text{ satisfiable and } \theta > 0 \\ \mathbb{I}\left[Y_f\right] & \text{if} \quad f \text{ or } \neg f \text{ not satisfiable or } \theta = 0 \\ \epsilon_0\left[Y_f\right] & \text{if} \quad \neg f \text{ satisfiable and } \theta < 0 \end{cases}$$

we have

$$\left\langle \exp\left[\theta \cdot f\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right] \to \left\langle \beta^f\left[Y_f, X_{[d]}\right], \kappa\left[Y_f\right]\right\rangle\left[X_{[d]}|\varnothing\right] .$$

We notice that these three activation cores are all non-vanishing boolean cores in dimension 2. This motivates the definition of Hard Logic Networks as follows.

Definition 8.7 (Hard Logic Network) Given a boolean statistic $\mathcal{F}$, a subset $A \subset [p]$ and a tuple $y_A \in \bigtimes_{l\in A}[2]$ the Hard Logic Network is the distribution

$$\mathbb{P}^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{(A,y_A)}\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}|\varnothing\right]$$

where $\kappa^{(A,y_A)}\left[Y_{[p]}\right] = \bigotimes_{l\in[p]} \kappa^{l,(A,y_A)}\left[Y_l\right]$ and for $l \in [p]$

$$\kappa^{l,(A,y_A)}\left[Y_l\right] = \begin{cases} \epsilon_{y_l}\left[Y_l\right] & \text{if} \quad l \in A \\ \mathbb{I}\left[Y_l\right] & \text{if} \quad l \notin A \end{cases},$$

provided that the normalization exists.

We notice that with the parametrization by the tuple (A, y_A), we can represent and non-vanishing elementary boolean tensor in $\bigotimes_{l\in[p]} \mathbb{R}^2$. Any activation leg core $\mathbb{I}\left[Y_l\right]$ furthermore results in a trivial contraction with the corresponding computation core. Based on these insights we can characterize Hard Logic Networks by propositional formulas.

Theorem 8.8 *For a boolean statistics $\mathcal{F}$ and parameters (A, y_A) the Hard Logic Network exists, if and only if the formula*

$$f^{\mathcal{F},(A,y_A)} = \left(\bigwedge_{l\in A\,:\,y_l=1} f_l\left[X_{[d]}\right]\right) \wedge \left(\bigwedge_{l\in A\,:\,y_l=0} \neg f_l\left[X_{[d]}\right]\right)$$

is satisfiable. In that case we have

$$\mathbb{P}^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right] = \left\langle f^{\mathcal{F},(A,y_A)}\right\rangle\left[X_{[d]}|\varnothing\right] .$$

Proof. We apply that

$$\left\langle \beta^{f_l}\left[Y_l, X_{[d]}\right], \kappa^{l,(A,y_A)}\left[Y_l\right]\right\rangle\left[X_{[d]}\right] = \begin{cases} f_l\left[X_{[d]}\right] & \text{if} \quad \kappa^{l,(A,y_A)}\left[Y_l\right] = \epsilon_1\left[Y_l\right] \\ \neg f_l\left[X_{[d]}\right] & \text{if} \quad \kappa^{l,(A,y_A)}\left[Y_l\right] = \epsilon_0\left[Y_l\right] \\ \mathbb{I}\left[X_{[d]}\right] & \text{if} \quad \kappa^{l,(A,y_A)}\left[Y_l\right] = \mathbb{I}\left[Y_l\right] \end{cases}$$

With the definition of $\kappa^{(A,y_A)}\left[Y_{[p]}\right]$ in Def. 8.7 we get

$$\begin{aligned} &\left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{(A,y_A)}\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}\right] \\ &= \left\langle \bigcup_{l\in[p]} \{\beta^{f_l}\left[Y_l, X_{[d]}\right], \kappa^{l,(A,y_A)}\left[Y_l\right]\}\right\rangle\left[X_{[d]}\right] \\ &= \left\langle \left\{\left\langle \beta^{f_l}\left[Y_l, X_{[d]}\right], \kappa^{l,(A,y_A)}\left[Y_l\right]\right\rangle\left[X_{[d]}\right] : l \in [p]\right\}\right\rangle\left[X_{[d]}\right] \\ &= \left\langle \{f_l\left[X_{[d]}\right] : \kappa^{l,(A,y_A)}\left[Y_l\right] = \epsilon_1\left[Y_l\right]\} \cup \{f_l\left[X_{[d]}\right] : \kappa^{l,(A,y_A)}\left[Y_l\right] = \epsilon_0\left[Y_l\right]\}\right\rangle\left[X_{[d]}\right] \\ &= f^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right]. \end{aligned}$$

The normalization of this contraction exists if and only if $\left\langle f^{\mathcal{F},(A,y_A)}\right\rangle[\varnothing] > 0$, that is if and only if $f^{\mathcal{F},(A,y_A)}$ is satisfiable. In that case we have

$$\mathbb{P}^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right] = \frac{f^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right]}{\left\langle f^{\mathcal{F},(A,y_A)}\right\rangle[\varnothing]} = \left\langle f^{\mathcal{F},(A,y_A)}\right\rangle\left[X_{[d]}|\varnothing\right]. \qquad \blacksquare$$

8.3 Hybrid Logic Networks

Markov Logic Networks are by definition positive distributions and are represented by positive activation tensors in $\Lambda^{\mathcal{F},\mathrm{EL}}$ (see Figure 6.1). In contrary, Hard Logic Networks model uniform distributions over model sets of the respective Knowledge Base and therefore have vanishing coordinates. They are represented by boolean activation tensors in $\Lambda^{\mathcal{F},\mathrm{EL}}$. Having discussed these representation approaches, let us now investigate the general case of distributions in $\Lambda^{\mathcal{F},\mathrm{EL}}$, which we call Hybrid Logic Networks.

8.3.1 Definition

Towards defining Hybrid Logic Networks, we represent any distribution computable by $\mathcal{F}$ and EL, as a normalization of a Hard Logic Network and a Markov Logic Network.

Theorem 8.9 *Any distribution $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{F},\mathrm{EL}}$ is a normalized contraction of a Hard Logic Network and a Markov Logic Network, that is there are for $l \in [p]$ boolean cores $\kappa^{l,(A,y_A)}\left[Y_l\right]$ and a weight vector $\theta\left[L\right]$ such that*

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \{\beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right]\} \cup \left(\bigcup_{l\in[p]} \{\kappa^{l,(A,y_A)}\left[Y_l\right], \alpha^{l,\theta}\left[Y_l\right]\}\right)\right\rangle\left[X_{[d]}|\varnothing\right].$$

Proof. We show the claim by splitting the activation cores into a contraction of hard (i.e. logical) and soft (i.e. probabilistic) activation cores. Since $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{F},\mathrm{EL}}$ we find for $l \in [p]$ non-

negative vectors $\xi^l\left[Y_l\right]$ such that

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \{\beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right]\} \cup \{\xi^l\left[Y_l\right] : l \in [p]\}\right\rangle \left[X_{[d]}|\varnothing\right] .$$

We now define for $l \in [p]$ boolean cores by

$$\kappa^{l,(A,y_A)}\left[Y_l\right] = \mathbb{I}_{>0}\left(\xi^l\left[Y_l\right]\right)$$

and coordinates of the canonical parameter vector $\theta\left[L\right]$ by

$$\theta\left[L = l\right] = \begin{cases} 0 & \text{if} \quad \kappa^{l,(A,y_A)}\left[Y_l\right] \neq \mathbb{I}\left[Y_l\right] \\ \ln\left[\frac{\xi^l[Y_l=1]}{\xi^l[Y_l=0]}\right] & \text{else} \end{cases} .$$

For all $l \in [p]$ we have that $\theta\left[L = l\right]$ is well-defined, since by assumption $\xi^l\left[Y_l\right]$ is positive, if $\kappa^{l,(A,y_A)}\left[Y_l\right] = \mathbb{I}\left[Y_l\right]$. If for $l \in [p]$ we have $\kappa^{l,(A,y_A)}\left[Y_l\right] \neq \mathbb{I}\left[Y_l\right]$, then $\xi^l\left[Y_l\right] \in \{\epsilon_1\left[Y_l\right], \epsilon_0\left[Y_l\right]\}$, since $\xi^l\left[Y_l\right] = 0\left[Y_l\right]$ would lead to a vanishing contraction with the basis encoding of $\mathcal{F}$. Then is a real number $\lambda_l > 0$ such that

$$\xi^l\left[Y_l\right] = \lambda_l \cdot \kappa^{l,(A,y_A)}\left[Y_l\right]$$

and, since $\alpha^{l,0}\left[Y_l\right] = \mathbb{I}\left[Y_l\right]$ also

$$\xi^l\left[Y_l\right] = \lambda_l \cdot \left\langle \kappa^{l,(A,y_A)}\left[Y_l\right], \alpha^{l,\theta}\left[Y_l\right]\right\rangle \left[Y_l\right] .$$

Conversely, if for $l \in [p]$ we have $\kappa^{l,(A,y_A)}\left[Y_l\right] = \mathbb{I}\left[Y_l\right]$, then there is a scalar $\lambda_l > 0$ such that

$$\begin{aligned} \xi^l\left[Y_l\right] &= \lambda_l \cdot \alpha^{l,\theta}\left[Y_l\right] \\ &= \lambda_l \cdot \left\langle \kappa^{l,(A,y_A)}\left[Y_l\right], \alpha^{l,\theta}\left[Y_l\right]\right\rangle \left[Y_l\right] . \end{aligned}$$

We therefore have

$$\begin{aligned} &\left\langle \{\beta^*\left[\mathcal{F}\right] Y_{[p]}, X_{[d]}\} \cup \{\xi^l\left[Y_l\right] : l \in [p]\}\right\rangle \left[X_{[d]}\right] \\ &= \left(\prod_{l\in[p]} \lambda_l\right) \cdot \left\langle \{\beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right]\} \cup \left(\bigcup_{l\in[p]} \{\kappa^{l,(A,y_A)}\left[Y_l\right], \alpha^{l,\theta}\left[Y_l\right]\}\right)\right\rangle \left[X_{[d]}\right] \end{aligned}$$

and the claim follows, since the normalization of a tensor is invariant under multiplication with the positive scalar $\left(\prod_{l\in[p]} \lambda_l\right)$. ∎

Motivated by this decomposition, we now define and parametrize Hybrid Logic Networks.

Definition 8.10 (Hybrid Logic Network) Given a boolean statistic $\mathcal{F}$ we call any element of $\Lambda^{\mathcal{F},\mathrm{EL}}$ a Hybrid Logic Network (HLN). The extended canonical parameter set to $\mathcal{F}$ is the set

$$\mathcal{P}_p \coloneqq \{(A, y_A)) : A \subset [p], y_A \in \bigtimes_{l\in A} [2]\} \times \mathbb{R}^p .$$

To each Hybrid Logic Network $\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right]$ we find a tuple (A, y_A, θ) consistent of a

subset $A \subset [p]$, a tuple $y_A \in \bigtimes_{l \in A}[2]$ and $\theta\,[L] \in \mathbb{R}^p$ such that

$$\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \xi^{(A,y_A,\theta)}\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}|\varnothing\right]$$

where the activation core is

$$\xi^{(A,y_A,\theta)}\left[Y_{[p]}\right] = \left\langle \alpha^{\theta}\left[Y_{[p]}\right], \kappa^{(A,y_A)}\left[Y_{[p]}\right]\right\rangle\left[Y_{[p]}\right].$$

Note that by Thm. 8.9 we find for any Hybrid Logic Network a parametrization by a tuple (A, y_A, θ). We further relate with the formalism of exponential families and determine the base measures which are used to build Hybrid Logic Networks.

Theorem 8.11 *Given a boolean statistic $\mathcal{F}$, we build for any $A \subset [p]$ and $y_A \in \bigtimes_{l\in A}[2]$ the formula*

$$f^{\mathcal{F},(A,y_A)} = \left(\bigwedge_{l\in A\,:\,y_l=1} f_l\left[X_{[d]}\right]\right) \wedge \left(\bigwedge_{l\in A\,:\,y_l=1} \neg f_l\left[X_{[d]}\right]\right).$$

Then we have

$$\Lambda^{\mathcal{F},\mathrm{EL}} = \bigcup_{A\subset[p]} \bigcup_{y_A\in\bigtimes_{l\in A}[2]\,:\,\left\langle f^{\mathcal{F},(A,y_A)}\right\rangle[\varnothing]>0} \Gamma^{\mathcal{F},f^{\mathcal{F},(A,y_A)}}.$$

Proof. Thm. 8.9 implies that $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{F},\mathrm{EL}}$ is equivalent to the existence of parameters (A, y_A, θ) with $\mathbb{P}\left[X_{[d]}\right] = \mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right]$. We further have that $\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right]$ is the member of the exponential family $\Gamma^{\mathcal{F},f^{\mathcal{F},(A,y_A)}}$ with canonical parameter θ, since

$$\begin{aligned}
\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right] &= \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \alpha^{\theta}\left[Y_{[p]}\right], \kappa^{(A,y_A)}\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}|\varnothing\right] \\
&= \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \alpha^{\theta}\left[Y_{[p]}\right], \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{(A,y_A)}\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}\right]\right\rangle\left[X_{[d]}|\varnothing\right] \\
&= \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \alpha^{\theta}\left[Y_{[p]}\right], f^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right]\right\rangle\left[X_{[d]}|\varnothing\right],
\end{aligned}$$

where we used Thm. 8.8 in the last equation. We therefore have $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{F},\mathrm{EL}}$ if and only if there is a tuple (A, y_A) with $\mathbb{P}\left[X_{[d]}\right] \in \Gamma^{\mathcal{F},f^{\mathcal{F},(A,y_A)}}$. ∎

We can thus understand Hybrid Logic Networks, as a union of exponential families, where the union is over all boolean activation cores understood as generating a base measure. This trick is known to the field of variational inference, see for Example 3.6 in [WJ08]. For an example of a tensor network representation of a Hybrid Logic Network see Figure 8.3.

8.3.2 Logical Reasoning Properties

Deciding probabilistic entailment (see Def. 5.8) with respect to Hybrid Logic Networks can be reduced to the hard logic parts of the network.

Theorem 8.12 *Let $\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right]$ be a Hybrid Logic Network. Given a query formula f we have*

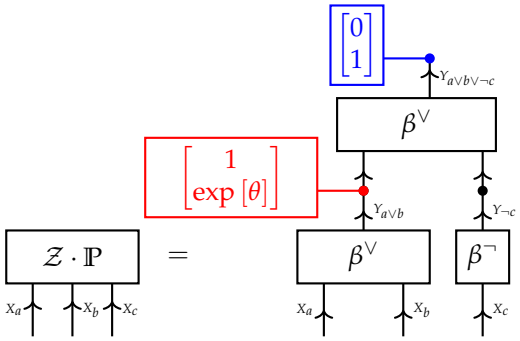


Figure 8.3: Tensor network representation of a Hybrid Logic Network with statistics $\mathcal{F} = (X_a \vee X_b, X_a \vee X_b \vee \neg c)$, following Thm. 8.9. Besides the computation cores required to compute the statistics, we have one non-trivial probabilistic soft activation core $\alpha^{0,\theta[L=0]}\,[Y_{a\vee b}]$ and one non-trivial constraint hard activation core $\kappa^{1}\,[Y_{a\vee b\vee\neg c}]$. The trivial activation cores $\kappa^{0}\,[Y_{a\vee b}]$ and $\alpha^{1,0}\,[Y_{a\vee b\vee\neg c}]$ are omitted from the diagram, since leaving the contraction invariant.

that

$$\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right] \models f$$

if and only if

$$f^{\mathcal{F},(A,y_A)} \models f\,.$$

Proof. This follows from Thm. 5.14 and

$$\mathbb{I}_{>0}\left(\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right]\right) = f^{\mathcal{F},(A,y_A)}$$

using the representation of Hybrid Logic Networks as exponential distributions in Thm. 8.9, and the fact that the support of any member of an exponential family is the base measure. ∎

Formulas in $\mathcal{F}$, which are entailed or contradicted by $\mathcal{KB}$ are redundant, as we show next.

Theorem 8.13 *If for a Hybrid Logic Network* $\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right]$ *and any formula* $f_l \in \mathcal{F}$ *we have*

$$f^{\mathcal{F},(A,y_A)} \models f_l \quad \textit{or} \quad f^{\mathcal{F},(A,y_A)} \models \neg f_l$$

then

$$\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\left[X_{[d]}\right] = \mathbb{P}^{\mathcal{F}/\{f_l\},(A/\{l\},y_{A/\{l\}},\tilde{\theta})}\left[X_{[d]}\right]$$

where $\tilde{\theta}$ *denotes the tensor* θ*, where the coordinate to* f_l *is dropped.*

Proof. The theorem can be shown by isolation of the factor to the hard formula, which is constant for all situations. ∎

A similar statement holds for the hard formulas itself, as shown in Thm. 5.5.

These results are especially interesting for the efficient implementation of Algorithm 6, which has been introduced in Chapter 5. By Thm. 8.12 only the hard logic parts of a Hybrid Logic Network are required in the ASK operation.

8.4 Sparse Representation Mechanisms

Let us now investigate sparsity mechanisms, which can be exploited for efficient representations of Hybrid Logic Networks.

8.4.1 Decomposition Sparsity

Decomposition sparsity aims at a tensor network representation of the computation network. It is a generalization of the syntactic decomposition of single propositional formulas as introduced in Chapter 4. When each formula coincides with some node formula of a node in a decomposition graph (see Def. 4.7), then the tensor network representation of the decomposition graph serves as computation network. Denoting the nodes to f_l by v_l we get

$$\begin{aligned}\beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right] = \langle \left\{\beta^{\circ_v}\left[Y_v, Y_{e^{\text{in}}}\right] : (e^{\text{in}}, \{v\}) \in \mathcal{E}\right\} \cup \\ \{\delta\left[Y_k, X_k\right] : k \in [d]\} \cup \\ \left\{\delta\left[Y_l, Y_{v_l}\right] : l \in [p]\right\} \rangle \left[X_{[d]}\right].\end{aligned}$$

In Figure 8.3 is an example of a Hybrid Logic Network, which computation network is represented using decomposition sparsity.

8.4.2 Selection Sparsity

Selection sparsity exploits formula selecting networks, which expressivity is the statistic $\mathcal{F}$ to a Hybrid Logic Network. In this sense selection sparsity is more flexible than Decomposition sparsity, since it does not demand identical subformulas, but only an identical decomposition graph. The limitation of this scheme is that it can only represent weighted sums of the formulas by contractions of the formula selecting tensor with a weight tensor. Such weighted sums appear in the energy tensor of Markov Logic Networks, but note that the energy tensor of generic Hybrid Logic Networks is not defined. In Example 8.14 we now demonstrate this representation mechanism on the energy tensors of Boltzmann machines.

Example 8.14 (Energy tensors of Boltzmann machines) In Sect. 8.1.4.2 we have described Boltzmann machines as an example of Markov Logic Networks, with formulas by

$$\mathcal{F} = \{X_k \Leftrightarrow X_l : k, l \in [d]\} \cup \{X_k : k \in [d]\}$$

To demonstrate a representation of the energy tensor of a Boltzmann machine using Selection sparsity, we now define a formula selecting network to $\mathcal{F}$ (see Figure 8.4). 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

$$\triangleleft[X_0 = x_0, X_1 = x_1] = \mathcal{F}_V\left(X_0 = x_0, X_1 = x_1, L_V = 0\right).$$

Given a weight matrix $W[L_{V,0}, L_{V,1}] \in \mathbb{R}^d \otimes \mathbb{R}^d$ and a bias vector $b[L_{V,0}] \in \mathbb{R}^d$ we choose a canonical parameter as

$$\theta\left[L_C, L_{V,0}, L_{V,1}\right] = \epsilon_0\left[L_C\right] \otimes W[L_{V,0}, L_{V,1}] + \epsilon_1\left[L_C\right] \otimes b[L_{V,0}] \otimes \epsilon_0\left[L_{V,1}\right].$$

We then have (see Figure 8.4)

$$\phi^{W,b}[X_{[d]}] = \left\langle \sigma^{\mathcal{A}}\left[X_{[d]}, L_C, L_{V,0}, L_{V,1}\right], \theta\left[L_C, L_{V,0}, L_{V,1}\right]\right\rangle \left[X_{[d]}\right].$$

$\Diamond$

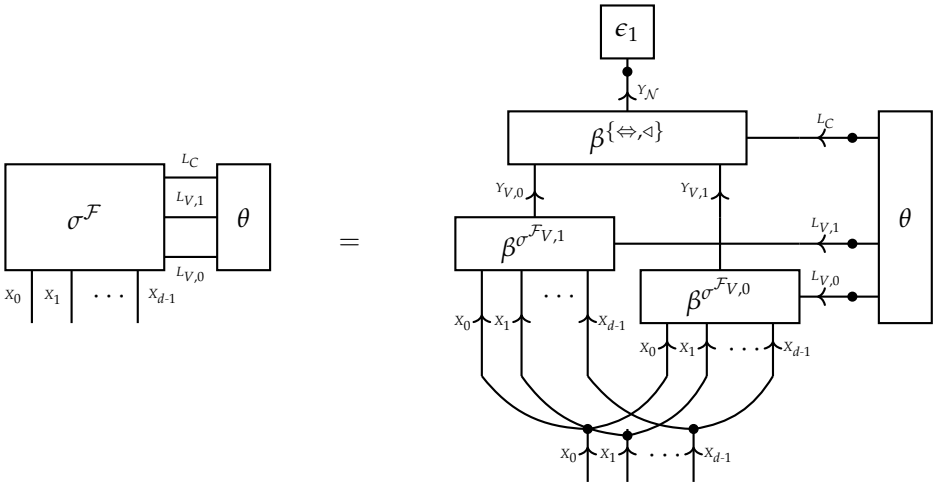


Figure 8.4: Tensor network representation of the energy of a Boltzmann machine

8.4.3 Polynomial Sparsity

We now introduce a further mechanism to find sparse representation formats for the introduced Hybrid Logic Networks, which we call polynomial sparsity. The motivation of this mechanism is the observation, that terms and clauses can be represented storing the signs of the literals only.

This is an application of the basis+ CP format introduced in Chapter 15, which is connected to polynomial decompositions of boolean functions. First of all, we establish a sparsity result for terms and clauses (see Def. 4.11).

Lemma 8.15 *Any term is representable by a single monomial and any clause is representable by a sum of at most two monomials.*

Proof. Let $\mathcal{V}_0$ and $\mathcal{V}_1$ be disjoint subsets of $\mathcal{V}$, then we have

$$Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1} = \epsilon_{\{x_k=0\,:\,k\in\mathcal{V}_0\}\cup\{x_k=1\,:\,k\in\mathcal{V}_1\}}\left[X_{\mathcal{V}_0\cup\mathcal{V}_1}\right]\otimes\mathbb{I}\left[X_{\mathcal{V}/(\mathcal{V}_0\cup\mathcal{V}_1)}\right]$$

and

$$Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1} = \mathbb{I}\left[X_{\mathcal{V}}\right] - \epsilon_{\{x_k=0\,:\,k\in\mathcal{V}_0\}\cup\{x_k=1\,:\,k\in\mathcal{V}_1\}}\left[X_{\mathcal{V}_0\cup\mathcal{V}_1}\right]\otimes\mathbb{I}\left[X_{\mathcal{V}/(\mathcal{V}_0\cup\mathcal{V}_1)}\right].$$

We notice that any tensors $\mathbb{I}$ and $\epsilon_x\otimes\mathbb{I}$ 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. ∎

A formula in conjunctive normal form is a conjunction of clauses, where clauses are disjunctions of literals being atoms (positive literals) or negated atoms (negative literals). Based on these normal forms, we show representations of formulas as sparse polynomials. We apply Lem. 8.15 to show the following sparsity bound.

Theorem 8.16 *Any formula f with a conjunctive normal form of n clauses satisfies*

$$\operatorname{rank}^{\text{bas+},d}(f) \leq 2^n.$$

Proof. Let f have a conjunctive normal form 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 Thm. 15.14 and Lem. 8.15 to get

$$\mathrm{rank}^{\mathrm{bas}+,d}(f) \leq \prod_{l \in [n]} \mathrm{rank}^{\mathrm{bas}+,d}\left(Z^{\vee}_{\mathcal{V}_0^l, \mathcal{V}_1^l}\right) \leq 2^n .$$

■

We apply this result on the sparse representation of a single formula to derive sparse representations for Hard Logic Networks and the energy tensor of Hybrid Logic Networks. Both results use in addition to Thm. 8.16 sparsity bounds, which are shown by explicit representation construction in Chapter 15.

Corollary 8.17 *Any Hard Logic Network $\mathcal{KB}$ obeys*

$$\mathrm{rank}^{\mathrm{bas}+,d}(\mathcal{KB}) \leq \prod_{f \in \mathcal{KB}} 2^{n_f}$$

Proof. We apply the contraction bound Thm. 15.14 for the decomposition

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

and get

$$\mathrm{rank}^{\mathrm{bas}+,d}(\mathcal{KB}) \leq \prod_{f \in \mathcal{KB}} \mathrm{rank}^{\mathrm{bas}+,d}(f) .$$

The claimed bound follows with Thm. 8.16. ■

Corollary 8.18 *The energy tensor of a Hybrid Logic Network with statistic $\mathcal{F}$*

$$\mathrm{rank}^{\mathrm{bas}+,d}\left(\left\langle \sigma^{\mathcal{F}}\left[X_{[d]}, L\right], \theta[L] \right\rangle \left[X_{[d]}\right]\right) \leq \sum_{l \in [p] \,:\, \theta[L=l] \neq 0} 2^{n_{f_l}} .$$

where n_{f_l} denotes the number of clauses in a conjunctive normal form of f_l.

Proof. We decompse the energy into the sum

$$\left\langle \sigma^{\mathcal{F}}\left[X_{[d]}, L\right], \theta[L] \right\rangle \left[X_{[d]}\right] = \sum_{l \in [p] \,:\, \theta[L=l] \neq 0} \theta[L = l] \cdot f_l\left[X_{[d]}\right]$$

and apply Thm. 15.13 to get

$$\mathrm{rank}^{\mathrm{bas}+,d}\left(\left\langle \sigma^{\mathcal{F}}\left[X_{[d]}, L\right], \theta[L] \right\rangle \left[X_{[d]}\right]\right) \leq \sum_{l \in [p] \,:\, \theta[L=l] \neq 0} \mathrm{rank}^{\mathrm{bas}+,d}\left(f_l\left[X_{[d]}\right]\right) \leq \sum_{l \in [p] \,:\, \theta[L=l] \neq 0} 2^{n_{f_l}} .$$

■

8.5 Mean Parameters of Hybrid Logic Networks

While mean parameter polytopes $\mathcal{M}_{\mathcal{S},\nu}$ to generic exponential families have been subject to Chapter 3, we in this section restrict to the mean polytopes of Hybrid Logic Networks, which we characterize using propositional logics. Hybrid Logic Networks are exponential families, which statistic $\mathcal{S}$ consists of coordinates with $\mathrm{im}\,(s_l) \subset \{0,1\}$ and are therefore propositional formulas. The convex polytope of mean parameters (see Def. 2.26) is for a statistic $\mathcal{F}$ of propositional formulas and a base measure ν the set

$$\mathcal{M}_{\mathcal{F},\mathbb{I}} = \left\{ \left\langle \mathbb{P}\left[X_{[d]}\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right] \right\rangle [L] \,:\, \mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\delta,\mathrm{MAX},\nu} \right\},$$

where by $\Lambda^{\delta,\mathrm{MAX},\nu}$ we denote the set of all by ν representable probability distributions. By Thm. 2.27 the convex polytope has a characterization as a convex hull

$$\mathcal{M}_{\mathcal{F},\mathbb{I}} = \mathrm{conv}\left(\sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right] \,:\, x_{[d]} \in \bigtimes_{k\in[d]} [2],\, \nu\left[X_{[d]} = x_{[d]}\right] = 1 \right).$$

We notice that for boolean statistics $\mathcal{F}$ all $\sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]$ are boolean vectors in $\mathbb{R}^p$. The mean parameter polytopes are thus 0/1-polytopes [Zie00; Gil07], from which a few obvious properties follow. Since those are convex subsets of the cube $[0,1]^p$, which vertices are all boolean vectors in $\mathbb{R}^p$, also each $\sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]$ (with $\nu\left[X_{[d]} = x_{[d]}\right] = 1$) is a vertex. Further, if for any $l \in [p]$ we have $\mu\,[L = l] \in \{0,1\}$, then $\mu\,[L]$ is on a proper face of the cube and thus also of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$. In the following, we characterize the faces of the mean parameter polytope.

8.5.1 Vertices by Hard Logic Networks

First of all, we show that the vertices of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ are the boolean vectors contained in $\mathcal{M}_{\mathcal{F},\mathbb{I}}$.

Theorem 8.19 *Given a boolean statistic $\mathcal{F}$, a base measure $\nu\left[X_{[d]}\right]$ and let $\mu\,[L] \in \mathcal{M}_{\mathcal{F},\mathbb{I}}$. Then the following are equivalent:*

(i) The set $\{\mu\,[L]\}$ is a vertex of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$

(ii) $\mu\,[L]$ is boolean

Proof. (i) $\Rightarrow$(ii): We use that any set of vectors, which convex hull forms a polytope, contains all vertices of that polytope (see Proposition 2.2 in [Zie13]). Since by definition $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ is the convex hull of boolean vectors $\sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]$, all vertices are boolean vectors.

(ii)$\Rightarrow$(i): Since $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ is contained in $[0,1]^p$ and all boolean vectors in $[0,1]^p$ are vertices, whenever a boolean vector is contained in $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ it is a vertex. ∎

We can further provide an equivalent criterion for a vertex by the satisfaction of a corresponding formula.

Theorem 8.20 *Let there be a boolean statistic $\mathcal{F}$ and let $\mu\,[L]$ be a boolean vector. Then the following are equivalent:*

(i) The set $\{\mu\,[L]\}$ is a vertex of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$.

(ii) The formula

$$f^{\mathcal{F},([p],\mu)} = \bigwedge_{l\in[p]} \neg^{(1-\mu[L=l])} f_l\left[X_{[d]}\right]$$

is satisfiable.

(iii) The Hard Logic Network $\mathbb{P}^{\mathcal{F},([p],\mu)}\left[X_{[d]}\right]$ *exists.*

Here we denote by $\neg^0$ *the identity connective and by* $\neg^1 = \neg$ *the logical negation.*

Proof. (i) ⇒(ii): If $\{\mu[L]\}$ is a vertex of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$, then there is $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ such that

$$\mu[L] = \sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right].$$

For all $l \in [p]$ we have

$$\neg^{(1-\mu[L=l])} f_l\left[X_{[d]} = x_{[d]}\right].$$

The index $x_{[d]}$ is therefore a model of $f^{\mathcal{F},([p],\mu)}$ and $f^{\mathcal{F},([p],\mu)}$ is satisfiable.

(ii)⇒(iii): This follows from Thm. 8.8.

(iii)⇒(i): The mean parameter is reproduced by $\mathbb{P}^{\mathcal{F},([p],\mu)}\left[X_{[d]}\right]$ and therefore $\mu[L] \in \mathcal{M}_{\mathcal{F},\mathbb{I}}$. Since $\mu[L]$ is further boolean, Thm. 8.19 implies that $\{\mu[L]\}$ is a vertex. ■

8.5.2 Faces represented by Hard Logic Networks

With Thm. 8.19 we showed that all vertices are reproduced by Hard Logic Networks. In general, as we show next, Hard Logic Networks are face measures to a set of faces which we characterize in the next theorem.

Theorem 8.21 *Let* $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$ *be a face of* $\mathcal{M}_{\mathcal{F},\mathbb{I}}$*, then the following are equivalent:*

(i) There is a face Q *of* $[0,1]^p$ *such that* $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} = \mathcal{M}_{\mathcal{F},\mathbb{I}} \cap Q$*.*

(ii) There is a Hard Logic Network which coincides with the normalized face measure $\left\langle \nu^{\mathcal{F},\mathcal{I}} \right\rangle \left[X_{[d]}|\varnothing\right]$*.*

(iii) The normalized face measure $\left\langle \nu^{\mathcal{F},\mathcal{I}} \right\rangle \left[X_{[d]}|\varnothing\right]$ *is in* $\Lambda^{\mathcal{S},\mathrm{EL},\nu}$*.*

Proof. (i) ⇒(ii): Given the face Q such that $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} = \mathcal{M}_{\mathcal{F},\mathbb{I}} \cap Q$, we construct a boolean and basis+ elementary tensor (see Chapter 15), such that

$$\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{\mathcal{I}}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right].$$

For each $l \in [p]$ we build the vectors

$$\kappa^{l,(A,y_A)}[Y_l] = \begin{cases} \epsilon_1[Y_l] & \text{if} \quad \forall \mu \in Q : \mu[L=l] = 1 \\ \epsilon_0[Y_l] & \text{if} \quad \forall \mu \in Q : \mu[L=l] = 0 \\ \mathbb{I}[Y_l] & \text{else} \end{cases}$$

and get the activation tensor as

$$\kappa^{\mathcal{I}}\left[Y_{[p]}\right] = \bigotimes_{l\in[p]} \kappa^{l,(A,y_A)}\left[Y_l\right] .$$

(ii)⇒(iii): By definition any Hard Logic Network is in $\Lambda^{\mathcal{S},\mathrm{EL},\nu}$.

(iii)⇒(i): If the normalized face measure $\left\langle \nu^{\mathcal{F},\mathcal{I}}\right\rangle \left[X_{[d]}|\varnothing\right]$ is in $\Lambda^{\mathcal{S},\mathrm{EL},\nu}$, Then there is a boolean and elementary tensor $\kappa^{\mathcal{I}}\left[Y_{[p]}\right]$ such that the face measure is

$$\nu^{\mathcal{S},\mathcal{I}}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{\mathcal{I}}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]}\right] .$$

Boolean and elementary tensors in leg-dimension two are basis+ elementary tensors. Given a basis+ elementary tensor decomposition of $\kappa^{\mathcal{I}}$ we construct a face of the cube $[0,1]^p$ by sets

$$\mathcal{U}^l = \begin{cases} \{0\} & \text{if} \quad \kappa^{l,(A,y_A)}\left[Y_l\right] = \epsilon_0\left[Y_l\right] \\ \{1\} & \text{if} \quad \kappa^{l,(A,y_A)}\left[Y_l\right] = \epsilon_1\left[Y_l\right] \\ [0,1] & \text{else} \end{cases}$$

and define a face by the cartesian product of these sets as

$$Q = \bigtimes_{l\in[p]} \mathcal{U}^l .$$

We then notice, that

$$\left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \kappa^{\mathcal{I}}\left[Y_{[p]}\right]\right\rangle \left[X_{[d]} = x_{[d]}\right] = 1$$

if and only if $\sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right] \in Q$. Therefore, the parametrized face is the intersection of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ with Q. ■

Thm. 8.21 characterizes the faces, which measures are reproducible by Hard Logic Networks. The corresponding mean parameters are the face centers.

Remark 8.22 For non-boolean statistics $\mathcal{S}$, we can derive limited versions of Thm. 8.21. Any face, which is the intersection with faces of the distorted cube

$$\bigtimes_{l\in[p]} [\min_x \mathcal{S}_l\left[X = x\right], \max_x \mathcal{S}_l\left[X = x\right]]$$

has a representation of its encoded pre-image with an activation tensor of basis+ rank 1. Since for leg dimensions larger than 2 the boolean tensors with elementary decomposition contain more tensors than the basis+ tensors of rank 1, we can represent further faces. ♢

8.5.3 Generic Faces

When the normalized face measure to a face is not reproduceable by a Hard Logic Network, it does not lie in $\Lambda^{\mathcal{S},\mathrm{EL},\nu}$. On the other side, Thm. 2.35 implies, that in general $\mathcal{S}$ is a sufficient statistic and it thus lies in $\Lambda^{\mathcal{S},\mathrm{MAX},\nu}$. The core $\kappa^{\mathcal{I}}\left[Y_{[p]}\right]$ has then CP rank of at least two. In general, we can express any face measure to faces of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ by a propositional formula.

Theorem 8.23 *Let $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$ be a face of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$. Then the face measure is given by the formula*

$$f^{\mathcal{F},(\mathcal{I})} = \bigvee_{\mu \in Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} \cap \{0,1\}^p} f^{\mathcal{F},([p],\mu)} = \bigvee_{\mu \in Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} \cap \{0,1\}^p} \bigwedge_{l \in [p]} \neg^{(1-\mu[L=l])} f_l \left[X_{[d]} \right] .$$

Proof. By Thm. 2.35 we have

$$f^{\mathcal{F},(\mathcal{I})} \left[X_{[d]} \right] = \left\langle \beta^{\mathcal{F}} \left[Y_{[p]}, X_{[d]} \right], \kappa^{\mathcal{I}} \left[Y_{[p]} \right] \right\rangle \left[X_{[d]} \right]$$

where

$$\kappa^{\mathcal{I}} \left[Y_{[p]} \right] = \sum_{\mu \in Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} \cap \mathrm{im}\left(\sigma^{\mathcal{S}}\right)} \epsilon_{y^{\mu}_{[p]}} \left[Y_{[p]} \right] .$$

We notice that for each mean parameter $\mu \in Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} \cap \mathrm{im}\left(\sigma^{\mathcal{S}}\right)$ the boolean tensors

$$\left\langle \beta^{\mathcal{F}} \left[Y_{[p]}, X_{[d]} \right], \epsilon_{y^{\mu}_{[p]}} \left[Y_{[p]} \right] \right\rangle \left[X_{[d]} \right]$$

coincide with $f^{\mathcal{F},([p],\mu)}$. Further, the formulas $f^{\mathcal{F},([p],\mu)}$ have disjoint model sets and their disjunction is therefore represented by their sum. We therefore have

$$\begin{aligned} f^{\mathcal{F},(\mathcal{I})} \left[X_{[d]} \right] &= \sum_{\mu \in Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} \cap \mathrm{im}\left(\sigma^{\mathcal{S}}\right)} f^{\mathcal{F},([p],\mu)} \left[X_{[d]} \right] \\ &= \bigvee_{\mu \in Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} \cap \{0,1\}^p} \bigwedge_{l \in [p]} \neg^{(1-\mu[L=l])} f_l \left[X_{[d]} \right] . \end{aligned}$$

■

8.5.4 Mean Parameters reproducible by Hybrid Logic Networks

Based on the representation results for face measures, we now study the mean parameters of Hybrid Logic Networks. Let us find conditions on a mean parameter $\mu\,[L]$ for the existence of a Hybrid Logic Network reproducing it.

Theorem 8.24 *Hybrid Logic Networks reproduce exactly those mean parameters $\mu\,[L]$, which are in the effective interior of cube faces $[0,1]^p$ (see Def. 2.38), i.e. for which there is a face $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$ of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ and a face Q of the cube $[0,1]^p$ such that*

$$\mu\,[L] \in \left(Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} \right)^{\circ} \quad \textit{and} \quad Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}} = \mathcal{M}_{\mathcal{F},\mathbb{I}} \cap Q\,.$$

Proof. Thm. 8.21 implies that the normalization of the face measure $\nu^{\mathcal{S},\mathcal{I}}$ to $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$ is a Hard Logic Network and in $\Lambda^{\mathcal{F},\mathrm{EL}}$. By assumption, $\mu\,[L]$ is in the effective interior of $\mathcal{M}_{\mathcal{F},\nu^{\mathcal{S},\mathcal{I}}}$ and therefore reproduced by a distribution $\mathbb{P}\left[X_{[d]} \right]$ in $\Gamma^{\mathcal{F},\nu^{\mathcal{S},\mathcal{I}}}$. Since $\mathbb{P}\left[X_{[d]} \right] \in \Lambda^{\mathcal{F},\mathrm{EL}}$ we found a Hybrid Logic Network reproducing $\mu\,[L]$. ■

We now provide a procedure to investigate, whether a mean parameter $\mu\,[L] \in \mathcal{M}_{\mathcal{F},\mathbb{I}}$ is reproduceable by a Hybrid Logic Network in $\Lambda^{\mathcal{F},\mathrm{EL}}$. To this end, we first determine the minimal face with respect to face inclusion (the partial order of the face lattice, see [Zie13]), which includes $\mu\,[L]$.

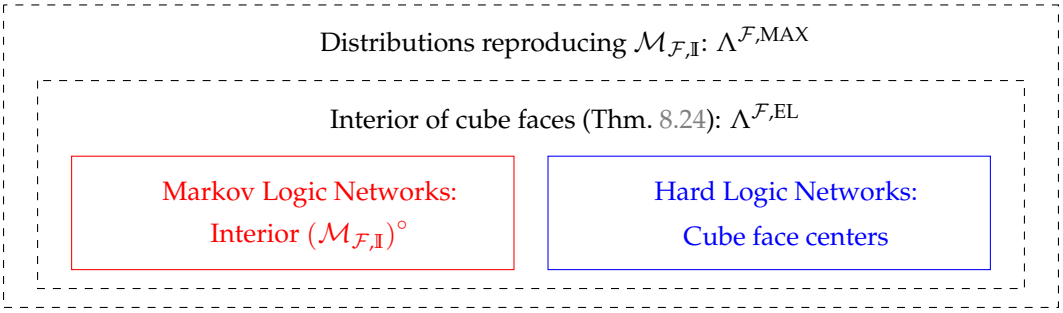


Figure 8.5: Sketch of distributions and their mean parameters with respect to . The Hybrid Logic Networks are also maximum entropy distribution, as we will show in Chapter 9.

Lemma 8.25 *For each mean parameter $\mu[L] \in \mathcal{M}_{\mathcal{F},\mathbb{I}}$ the minimal face of the cube $[0,1]^p$ containing $\mu[L]$ is*

$$Q^\mu = \bigtimes_{l \in [p]} \mathcal{U}^l .$$

where for each $l \in [p]$

$$\mathcal{U}^l = \begin{cases} \{\mu[L = l]\} & \text{if} \quad \mu[L = l] \in \{0,1\} \\ [0,1] & \text{else} \end{cases} .$$

Proof. Since $\mathcal{M}_{\mathcal{F},\mathbb{I}} \subset [0,1]^p$, $\mu[L]$ is always contained in the maximal face $[0,1]^p = \bigtimes_{l\in[p]}[0,1]$ of the cube, and the smallest face containing $\mu[L]$ exists. Any face of the cube has a representation as a cartesian product of $\mathcal{U}^l \in \{\{0\},\{1\},[0,1]\}$ and contains $\mu[L]$ if and only if

$$\forall l \in [p] : \left(\mathcal{U}^l = \{0\} \Rightarrow \mu[L = l] = 0\right) \wedge \left(\mathcal{U}^l = \{1\} \Rightarrow \mu[L = l] = 1\right) .$$

The minimal face containing $\mu[L]$ is therefore Q^μ. ∎

Lemma 8.26 *$\mu[L]$ is reproducible by a Hybrid Logic Network, if and only if it is in the effective interior of the set*

$$\mathcal{M}_{\mathcal{F},\mathbb{I}} \cap Q^\mu .$$

Here Q^μ is the minimal face of the cube containing $\mu[L]$.

Proof. This follows from Thm. 8.24 and Lem. 8.25. ∎

We can extend the discussion to $\Lambda^{\mathcal{F},\mathcal{G}}$, where $\mathcal{G}$ is an arbitrary hypergraph. Whenever a normalized face measure is in $\Lambda^{\mathcal{F},\mathcal{G}}$, then the all mean parameters on the interior of the face can be reproduced by $\Lambda^{\mathcal{F},\mathcal{G}}$.

8.5.5 Expressivity of Hybrid Logic Networks

In Figure 8.5 we classify the distributions by their mean parameters in $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ and draw a corresponding sketch of the mean parameter polytope in Figure 8.6. We have already shown, that the distributions in $\Lambda^{\mathcal{F},\mathrm{MAX}}$ suffice to reproduce all mean parameters in $\mathcal{M}_{\mathcal{F},\mathbb{I}}$. The Hybrid Logic

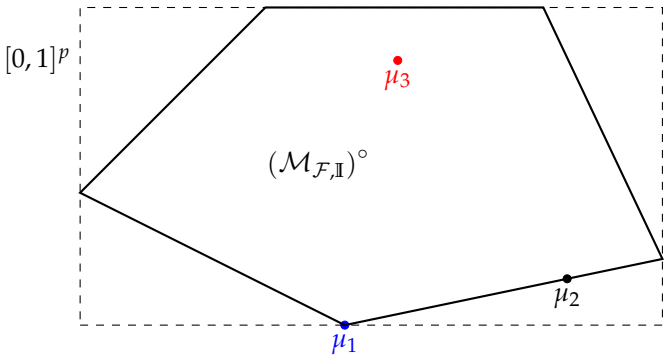


Figure 8.6: Sketch of the mean polytope $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ to a statistic $\mathcal{F}$ which is minimal with respect to ν, as a special case of the more generic sketch Figure 2.5. The mean polytope is a subset of the p-dimensional cube $[0,1]^p$ (dashed in the sketch), where each mean parameter in one of the three cases μ_1, μ_2 or μ_3. Face centers μ_1 (in particular the vertices $\mathcal{M}_{\mathcal{F},\mathbb{I}} \cap \{0,1\}^p$) are reproducible by a Hard Logic Network given $\mathcal{F}$. Mean points on non-vertex faces outside the interior $\mu_2 \in \mathcal{M}_{\mathcal{F},\mathbb{I}} / (\mathcal{M}_{\mathcal{F},\mathbb{I}}{}^\circ \cap \{0,1\}^p)$ are reproducible by Hybrid Logic Networks with statistic $\mathcal{F}$ and refined base measure $\tilde{\nu}$. Interior points $\mu_3 \in \mathcal{M}_{\mathcal{F},\mathbb{I}}{}^\circ$ are reproducible by a Markov Logic Network with statistic $\mathcal{F}$.

Networks are the subset $\Lambda^{\mathcal{F},\mathrm{EL}} \subset \Lambda^{\mathcal{F},\mathrm{MAX}}$, which can be computed with elementary activation cores. We now investigate, in which cases also $\Lambda^{\mathcal{F},\mathrm{EL}}$ suffices to reproduce all mean parameters in $\mathcal{M}_{\mathcal{F},\mathbb{I}}$, that is in which cases $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ coincides with

$$\left\{ \left\langle \mathbb{P}, \sigma^{\mathcal{F}} \right\rangle [L] \,:\, \mathbb{P} \in \Lambda^{\mathcal{F},\mathrm{EL}} \right\} .$$

While this set is obviously a subset of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$, we investigate, 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 8.27 (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.

$\Diamond$

We can relate our two standard examples of the atomic and the minterm formula sets to well-studied polytopes, namely the d-dimensional hypercube and the standard simplex (see Lecture 0 in [Zie13]).

Example 8.28 (Atomic formulas) Let us consider the case of atomic formulas. The mean polytope in this case is the d-dimensional hypercube

$$\mathcal{M}_{\mathcal{F}_{[d]},\mathbb{I}} = [0,1]^p$$

which is called a simple polytope, since each vertex is contained in the minimal number of d facets.

The faces of a hypercube are enumerated in the following way. Each face is characterized by the projections onto each variable, which is either $\{0\}$, $\{1\}$ or $[0,1]$. The projections are represented by the tuple (A, y_A) defined in the following way:

- We define the set $A \subset [d]$ of variables, such that the projection onto the variable is $\{0\}$ or $\{1\}$
- We define to each $l \in A$ an index $y_l = 0$ if the projection is $\{0\}$ and $y_l = 1$ if the projection is $\{1\}$.

Trivially, each face of the hypercube is a cube face and $\mathcal{M}^{\mathrm{EL}}_{\mathcal{F}_{[d]}} = \mathcal{M}_{\mathcal{F}_{[d]}}$. $\Diamond$

With the findings of the above sections and the enumeration of cube faces, we can not characterize the set of mean parameters realizable by Hybrid Logic Networks.

Definition 8.29 The elementary realizable mean parameters to a statistic $\mathcal{F}$ and a base measure ν is the set

$$\mathcal{M}^{\mathrm{EL}}_{\mathcal{F},\nu} = \bigcup_{A \subset [p]} \bigcup_{y_A \in \times_{l \in A}[2]} \left(\mathcal{M}_{\mathcal{F},\nu} \cup Q^{A,y_A} \right)^{\circ}$$

where Q^{A,y_A} is the face of the hypercube (see Example 8.28). We say, that a mean polytope $\mathcal{M}_{\mathcal{S},\nu}$ is cube-like, if all mean parameters are elementary realizable, that is

$$\mathcal{M}^{\mathrm{EL}}_{\mathcal{F},\nu} = \mathcal{M}_{\mathcal{S},\nu}\,.$$

We have

$$\mathcal{M}^{\mathrm{EL}}_{\mathcal{F},\nu} = \left\{ \left\langle \mathbb{P}, \sigma^{\mathcal{F}} \right\rangle [L] \,:\, \mathbb{P} \in \Lambda^{\mathcal{F},\mathrm{EL}} \right\}.$$

A further example of cube-like polytopes, i.e. where $\mathcal{M}^{\mathrm{EL}}_{\mathcal{F},\nu} = \mathcal{M}_{\mathcal{F},\mathbb{I}}$, are families of minterm statistics.

Example 8.30 (Minterm formulas) The mean polytope is in the case of the minterm statistic (also referred to as universal statistic) and a base measure ν the $\langle \nu \rangle [\varnothing] - 1$-dimensional standard simplex

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

In this case, $\Lambda^{\mathcal{F}_\wedge,\mathrm{EL}}$ contains any distribution and therefore trivially realizes any mean parameter in $\mathcal{M}_{\mathcal{F}_\wedge,\mathbb{I}}$.

The faces of the standard simplex are itself standard simplices to base measures $\tilde{\nu}$ with $\tilde{\nu} \prec \nu$. We store them by the tuple (A, y_A), where A is the support of $\tilde{\nu}$ and $y_l = 0$ for $l \in A$. Each face is a cube face, since it is the intersection of $\Lambda^{\mathcal{F}_\wedge,\mathrm{EL}}$ with the cube face (A, y_A). In particular, we have $\mathcal{M}^{\mathrm{EL}}_{\mathcal{F}_\wedge} = \mathcal{M}_{\mathcal{F}_\wedge}$. $\Diamond$

8.5.6 The Limit of Hard Logic

While Lem. 8.6 describes the limiting distributions of a statistic consistent of a single formula under annealing, we now turn towards statistics of multiple formulas.

Theorem 8.31 *Let $\mathcal{F}$ be a boolean statistics, $\theta[L] \in \mathbb{R}^p$ and $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$ the face such that $\theta[L] \in C^{\mathcal{I}}_{\mathcal{F},\nu}$. In the limit $\beta \to \infty$ the distribution $\mathbb{P}^{\mathcal{F},\beta\cdot\theta}\left[X_{[d]}\right]$ converges coordinatewise to the normalized face*

measure $f^{\mathcal{F},(\mathcal{I})}$, that is for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have

$$\mathbb{P}^{\mathcal{F},\beta\cdot\theta}\left[X_{[d]} = x_{[d]}\right] \to \frac{1}{\left\langle f^{\mathcal{F},(\mathcal{I})}\left[X_{[d]}\right]\right\rangle[\varnothing]} \cdot f^{\mathcal{F},(\mathcal{I})}\left[X_{[d]} = x_{[d]}\right] .$$

Proof. The limit is supported on the set of states, for which $\left\langle \theta[L], \sigma^{\mathcal{F}}\left[X_{[d]} = x_{[d]}, L\right]\right\rangle[\varnothing]$ is maximal. This set has an basis encoding by the face measure (see Def. 2.32) and the limiting distribution is thus the normalized face measure. ∎

We can now use characterizations of the face measure in case of satisfiable formulas to get a more explicit representation of the limiting distribution.

Theorem 8.32 *If for a $\theta[L] \in \mathbb{R}^p$ the formula*

$$f = \bigwedge_{l\,:\,\theta[L=l]\neq 0} \neg^{(1-\mathbb{I}_{>0}(\theta[L=l]))} f_l$$

is satisfiable, then $\mathbb{P}^{\mathcal{F},\beta\cdot\theta}\left[X_{[d]}\right]$ converges for $\beta \to \infty$ coordinatewise to the normalized uniform distribution among the models of f.

Proof. If f is satisfiable, then the face measure $f^{\mathcal{F},(\mathcal{I})}$ to the face $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$ with $\theta[L] \in C^{\mathcal{I}}_{\mathcal{F},\nu}$ is the uniform distribution among the models of f. The claim follows from Thm. 8.31. ∎

We notice that f is computable with elementary activation cores, that is for

$$\kappa\left[Y_{[p]}\right] = \left(\bigotimes_{l\,:\,\theta[L=l]=0} \mathbb{I}[Y_l]\right) \otimes \left(\bigotimes_{l\,:\,\theta[L=l]>0} \epsilon_1[Y_l]\right) \otimes \left(\bigotimes_{l\,:\,\theta[L=l]<0} \epsilon_0[Y_l]\right)$$

we have

$$f\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \kappa\left[Y_{[p]}\right]\right\rangle\left[X_{[d]}|\varnothing\right] .$$

8.6 Discussion

We understand Hybrid Logic Networks as neuro-symbolic models:

- **Neural Paradigm** is the representation of models as compositions of smaller models. For Hybrid Logic Networks, the decompositions of logical formulas into their connectives implements this neural paradigm. In more generality the decompositions of sufficient statistics into composed functions has a tensor network representation based on basis calculus. Deeper nodes as carrying correlations of lower nodes.
- **Symbolic Paradigm** refers to the formalization of reasoning by human-interpretable symbols. Since the sufficient statistics of Hybrid Logic Networks are propositional formulas, they can be verbalized and interpreted based on their syntactical decompositions.

9

Hybrid Logic Inference

In this chapter we investigate the inference properties of Hybrid Logic Networks, exploiting the characterizations of the corresponding mean parameter polytopes in Chapter 8. We investigate unconstrained parameter estimated for Markov Logic Networks and Hybrid Logic Networks, which are special cases of the backward mappings introduced in Chapter 2. We then study structure learning and present heuristic strategies leading to efficient algorithms.

9.1 Entropy Optimization

We now motivate Hybrid Logic Networks as distributions with maximum entropy under a moment constraint and then exploit cross-entropy minimization schemes to estimate the parameters of Hybrid Logic Networks.

9.1.1 Entropy Maximization

The Maximum Entropy Problem $\mathrm{P}^{\mathbb{H}}_{\mathcal{F},\nu,\mu^*}$ for a boolean statistic $\mathcal{F}$ is

$$\operatorname{argmax}_{\mathbb{P}\in\Lambda^{\delta,\mathrm{MAX}}} \quad \mathbb{H}\left[\mathbb{P}\right] \quad \text{subject to} \quad \left\langle \mathbb{P}, \sigma^{\mathcal{F}} \right\rangle [L] = \mu^*[L] \tag{$\mathrm{P}^{\mathbb{H}}_{\mathcal{F},\mu^*}$}$$

where by $\Lambda^{\delta,\mathrm{MAX}}$ we denote all probability distributions.

Theorem 9.1 *Let $\mu^*[L] \in \mathcal{M}_{\mathcal{F},\mathbb{I}}$ and the minimal face of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$, which includes $\mu^*[L]$, be $Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$, and let*

$$\theta_* = B^{\mathcal{F},f^{\mathcal{F},(\mathcal{I})}}(\mu^*[L]),$$

where $B^{\mathcal{F},f^{\mathcal{F},(\mathcal{I})}}$ is a backward mapping in the exponential family $\Gamma^{\mathcal{F},f^{\mathcal{F},(\mathcal{I})}}$ and $f^{\mathcal{F},(\mathcal{I})}$ is the face measure to $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ (see Thm. 8.23). The solution of the Maximum Entropy Problem ($\mathrm{P}^{\mathbb{H}}_{\mathcal{F},\mu^*}$) is then

$$\mathbb{P}\left[X_{[d]}\right] = \mathbb{P}^{(\mathcal{F},\theta_*,f^{\mathcal{F},(\mathcal{I})})}[X_{[d]}].$$

Proof. Any feasible distribution has to be representable by $f^{\mathcal{F},(\mathcal{I})}$, since $\mu^*[L] \in Q^{\mathcal{I}}_{\mathcal{F},\mathbb{I}}$. The solution therefore coincides with the solution of the maximum entropy problem with respect to $\mathcal{F}$ and $f^{\mathcal{F},(\mathcal{I})}$ as a base measure. Since $Q^{\mathcal{I}}_{\mathcal{S},\nu}$ is minimal, $\mu[L]$ is on the effective interior of the face and the claim follows with Thm. 3.20. ∎

Thm. 9.1 characterizes the solution of the Maximum Entropy Problem for arbitrary positions of the mean parameter. Note, that if $\mu^*[L] \notin \mathcal{M}_{\mathcal{S},\nu}$ then no distribution is feasible for Problem ($\mathrm{P}^{\mathbb{H}}_{\mathcal{F},\mu^*}$) and there is no solution. We are especially interested in situations, where the solution is a Hybrid Logic Network. As we state next, this is exactly the case if the mean parameter is reproducible by a Hybrid Logic Network (see Sect. 8.5.4).

Theorem 9.2 *The solution of the Maximum Entropy Problem* ($\mathrm{P}^{\mathbb{H}}_{\mathcal{F},\mu^*}$) *is a Hybrid Logic Network, if and only if* $\mu^*[L]$ *is reproducible by a Hybrid Logic Network.*

Proof. We notice that if and only if the solution of the Maximum Entropy Distribution is a Hybrid Logic Network, then the normalization of the corresponding face measure $f^{\mathcal{F},(\mathcal{I})}$ of the minimal face containing $\mu^*[L]$ is in $\Lambda^{\mathcal{F},\mathrm{EL}}$. This is equivalent to $\mu^*[L]$ being reproducible by a Hybrid Logic Network. ∎

9.1.2 Cross Entropy Minimization

Different to Maximum Entropy Problems, we formulate the Maximum Likelihood Problem as cross entropy minimization with respect to Hybrid Logic Networks, that is

$$\operatorname{argmin}_{\mathbb{P}\in\Lambda^{\mathcal{F},\mathrm{EL}}}\mathbb{H}\left[\mathbb{P}^*,\mathbb{P}\right] \qquad (\mathrm{P}^{\mathrm{M}}_{\Lambda^{\mathcal{F},\mathrm{EL}},\mathbb{P}^*})$$

When choosing $\mathbb{P}^*$ by an empirical distribution, this minimization problem is the Maximum Likelihood Problem (see Chapter 3). To solve the Maximum Likelihood Problem on Hybrid Logic Networks we choose the parametrization by tuples $(A, y_A, \theta) \in \mathcal{P}_p$ and aim to solve

$$\operatorname{argmin}_{(A,y_A,\theta)\in\mathcal{P}_p}\mathbb{H}\left[\mathbb{P}^*,\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right].$$

Lemma 9.3 *For any* θ *we have*

$$\min_{(A,y_A)}\mathbb{H}\left[\mathbb{P}^*,\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right]=\mathbb{H}\left[\mathbb{P}^*,\mathbb{P}^{\mathcal{F},(A^{\mu^*},y_A^{\mu^*},\theta)}\right]$$

where $(A^{\mu^*}, y_A^{\mu^*})$ *parametrize the smallest cube face containing* $\mu^*[L]=\left\langle\mathbb{P}^*\left[X_{[d]}\right],\sigma^{\mathcal{F}}\left[X_{[d]},L\right]\right\rangle[L]$.

Proof. We decompose the cross entropy in three terms

$$\begin{aligned}\mathbb{H}\left[\mathbb{P}^*,\mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right]&=\left\langle\mathbb{P}^*\left[X_{[d]}\right],-\ln\left[f^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right]\right]\right\rangle[\varnothing]\\&+\left\langle\mathbb{P}^*\left[X_{[d]}\right],\sigma^{\mathcal{F}}\left[X_{[d]},L\right],-\ln\left[\alpha^{\theta}\left[Y_{[p]}\right]\right]\right\rangle[\varnothing]\\&+\ln\left[\left\langle\beta^{\mathcal{F}}\left[Y_{[p]},X_{[d]}\right],\kappa^{(A,y_A)}\left[Y_{[p]}\right],\alpha^{\theta}\left[Y_{[p]}\right]\right\rangle[\varnothing]\,X_{[d]}\right].\end{aligned}$$

The first term can be characterized by

$$\left\langle\mathbb{P}^*\left[X_{[d]}\right],-\ln\left[f^{\mathcal{F},(A,y_A)}\left[X_{[d]}\right]\right]\right\rangle[\varnothing]=\begin{cases}0 & \text{if}\quad \mathbb{P}^*\models f^{\mathcal{F},(A,y_A)}\\ \infty & \text{if}\quad \mathbb{P}^*\not\models f^{\mathcal{F},(A,y_A)}\end{cases}.$$

The cross entropy is therefore finite, if and only if $A\subset A^{\mu^*}$ and $y_A=y_A^{\mu^*}|_A$. Among those tuples, only the third term of the cross-entropy varies, where we have

$$\ln\left[\left\langle\beta^{\mathcal{F}}\left[Y_{[p]},X_{[d]}\right],\kappa^{(A,y_A)}\left[Y_{[p]}\right],\alpha^{\theta}\left[Y_{[p]}\right]\right\rangle[\varnothing]\right]\leq\ln\left[\left\langle\beta^{\mathcal{F}}\left[Y_{[p]},X_{[d]}\right],\kappa^{(A^{\mu^*},y_A^{\mu^*})}\left[Y_{[p]}\right],\alpha^{\theta}\left[Y_{[p]}\right]\right\rangle[\varnothing]\right]$$

The minimum of the cross entropy is therefore taken at $A=A^{\mu^*}$ and $y_A=y_A^{\mu^*}$. ∎

We conclude from Lem. 9.3 that the hard parameter to the minimum of the cross entropy parametrizes the smallest cube face containing $\mu^*[L]$, provided that the minimum exists. With

Lem. 3.17 we can now characterize the solution of the cross entropy minimization problem Problem ($\mathrm{P}^{\mathrm{M}}_{\Lambda^{\mathcal{F},\mathrm{EL}},\mathbb{P}^*}$) for Hybrid Logic Networks.

Theorem 9.4 *Let $\mathbb{P}^*\left[X_{[d]}\right]$ be a distribution, $\mathcal{F}$ a boolean statistic.We build the mean parameter $\mu^*[L] = \left\langle \mathbb{P}^*\left[X_{[d]}\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right]\right\rangle[L]$ and have the following:*

(1) If $\mu^*[L] \in (\mathcal{M}_{\mathcal{F},\mathbb{I}})^\circ$ then

$$\min_{(A,y_A,\theta)\in\mathcal{P}_p} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right] = \mathbb{H}\left[\mathbb{P}^{\mathcal{F},(A^{\mu^*},y_A^{\mu^*},\hat{\theta})}\right]$$

where $\hat{\theta} = B^{\mathcal{F},f^{\mathcal{F},((A^{\mu^*[L]},y_{A^{\mu^*[L]}}))}}(\mu^*)$.

(2) If $\mu^*[L] \notin (\mathcal{M}_{\mathcal{F},\mathbb{I}})^\circ$ and $\mu^*[L] \in \overline{\mathcal{M}_{\mathcal{F},\mathbb{I}}}$ then there is a sequence $(\mu_n[L])_{n\in\mathbb{N}} \subset \mathcal{M}_{\mathcal{F},\mathbb{I}}$ converging coordinatewise to $\mu^*[L]$ and

$$\min_{(A,y_A,\theta)\in\mathcal{P}_p} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right] = \lim_{\mu_n[L]\to\mu^*[L]} \mathbb{H}\left[\mathbb{P}^{\mathcal{F},(A^{\mu_n},y_A^{\mu_n},\theta_n)}\right]$$

where $\theta_n = B^{\mathcal{F},f^{\mathcal{F},((A^{\mu_n},y_{A^{\mu_n}}))}}(\mu_n)$.

(3) If $\mu^*[L] \notin \overline{\mathcal{M}_{\mathcal{F},\mathbb{I}}}$ then

$$\min_{(A,y_A,\theta)\in\mathcal{P}_p} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right] = \infty\,.$$

Proof. This follows from Lem. 3.17. ∎

When $\mu^*[L]$ is not reproduceable by a Hybrid Logic Network, we are in the case where the smallest face, such that $\mu^*[L]$ is contained is not an intersection of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ with a cube face. In this case, there is no solution of the Maximum Likelihood Problem ($\mathrm{P}^{\mathrm{M}}_{\Lambda^{\mathcal{F},\mathrm{EL}},\mathbb{P}^*}$) in the set of Hybrid Logic Networks, since the minimum is not taken. The reason for this lies in the expressivity problem of Hybrid Logic Networks, which do not reproduce the interior of such faces, but tend in a limit of large canonical parameters to any mean parameter on such faces.

9.2 Forward and Backward Mappings

Forward and backward mappings have been introduced for exponential families in Chapter 2. We now generalize them to Hybrid Logic Networks, which are parametrized by tuples $(A, y_A, \theta) \in \mathcal{P}_p$.

Definition 9.5 The forward mapping for a Hybrid Logic Network is the function

$$F^{\mathcal{F}} : \mathcal{P}_p \to \mathcal{M}_{\mathcal{F},\mathbb{I}} \subset \mathbb{R}^p$$

where for $(A, y_A, \theta) \in \mathcal{P}_p$

$$F^{\mathcal{F}}((A, y_A, \theta)) = \left\langle \left\langle \tau^{(A,y_A,\theta)}\left[Y_{[p]}\right], \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right]\right\rangle\left[X_{[d]}|\varnothing\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right]\right\rangle[L]\,.$$

A backward mapping for a Hybrid Logic Network is any function

$$B^{\mathcal{F}} : \mathrm{im}\left(F^{\mathcal{F}}\right) \to \mathcal{P}_p$$

such that for any $(A, y_A, \theta) \in \mathcal{P}_p$ we have $B^{\mathcal{F}}(F^{\mathcal{F}}(\tau^{\mathrm{EL}}))$.

From the expressivity study in Chapter 8 we know that for any $\mu[L]$ there is a $(A, y_A, \theta) \in \mathcal{P}_p$ with $F^{\mathcal{F}}((A, y_A, \theta)) = \mu[L]$, if and only if $\mu[L] \in \mathcal{M}^{\mathrm{EL}}_{\mathcal{F},\nu}$. In particular, this implies that the image of $F^{\mathcal{F}}$ is the subset $\mathcal{M}^{\mathrm{EL}}_{\mathcal{F},\nu} \subset \mathcal{M}_{\mathcal{F},\mathbb{I}}$, which is the union of cube face interiors.

A backward mapping can be constructed as follows:

- Choose $A = \{l : \mu[L = l] \in \{0, 1\}\}$, and for $l \in A$ $y_L = \mu[L = l]$
- Use the backward mapping of the exponential family $\Gamma^{\mathcal{F}, f^{\mathcal{F},(A,y_A)}}$ to compute

$$\theta[L] = B^{\mathcal{F}, f^{\mathcal{F},(A,y_A)}}(\mu[L])$$

9.2.1 Backward Mappings in Closed Form

We recall from Chapter 3, 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 mapping is in general implicitly characterized by a maximum entropy problem constrained to matching expected sufficient statistics. We investigate in this section the specific examples of the minterm, maxterm and the atomic statistic, where closed forms are available for the backward mapping. 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. Furthermore, for these statistics we have $\mathcal{M}^{\mathrm{EL}}_{\mathcal{F},\nu} = \mathcal{M}_{\mathcal{F},\mathbb{I}}$.

9.2.1.1 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. 8.1.3 how to represent any positive distribution by a distribution in the family of minterms (respectively maxterms), see Thm. 8.4. Here we extend to the representation of distributions with arbitrary supports and provide forward and backward mappings.

We use the tuple enumeration of the maxterms and minterms by $\bigtimes_{k\in[d]}[2]$ introduced in Sect. 4.3.3. With respect to this enumeration the canonical parameters and mean parameters are tensors in $\bigotimes_{k\in[d]} \mathbb{R}^2$.

Theorem 9.6 *For the Hybrid Logic Networks to the minterm and maxterm statistics*

$$\mathcal{F}_\wedge := \{Z^\wedge_{x_{[d]}} : x_{[d]} \in \bigtimes_{k\in[d]}[2]\} \quad \textit{and} \quad \mathcal{F}_\vee := \{Z^\vee_{x_{[d]}} : x_{[d]} \in \bigtimes_{k\in[d]}[2]\}$$

we have the forward mappings

$$F^\wedge((A, y_A, \theta)) = \left\langle \beta^{\mathcal{F}_\wedge}\left[Y_\wedge, X_{[d]}\right], \tau^{(A,y_A,\theta)}[Y_\wedge], \delta\left[X_{[d]}, L_{[d]}\right]\right\rangle \left[L_{[d]}|\varnothing\right]$$

and

$$F^\vee((A, y_A, \theta)) = \mathbb{I}\left[L_{[d]}\right] - \left\langle \beta^{\mathcal{F}_\vee}\left[Y_\vee, X_{[d]}\right], \tau^{(A,y_A,\theta)}[Y_\vee], \delta\left[X_{[d]}, L_{[d]}\right]\right\rangle \left[L_{[d]}|\varnothing\right].$$

Further, the map

$$B^{\wedge}(\mu\left[L_{[d]}\right]) = \left(\left\{l : l \in [p], \mu\left[L = l\right] = 0\right\}, 0_A, \ln\left[\mu\left[L_{[d]}\right]\right]\right)$$

(we set here $\ln[0] = 0$*) is a backward mapping for the minterm statistic.*
The map

$$B^{\vee}(\mu\left[L_{[d]}\right]) = \left(\left\{l : l \in [p], \mu\left[L = l\right] = 1\right\}, 1_A, -\ln\left[\mu\left[L_{[d]}\right]\right]\right)$$

(we set here $\ln[0] = 0$*) is a backward mapping for the minterm statistic.*

Proof. The minterm statistic $\mathcal{F}_{\wedge}$ has a selection encoding

$$\sigma^{\mathcal{F}_{\wedge}}\left[X_{[d]}, L_{[d]}\right] = \delta\left[X_{[d]}, L_{[d]}\right]$$

and the mean parameter to any distribution $\mathbb{P}\left[X_{[d]}\right]$ has therefore the coordinates to $l_{[d]} \in \bigtimes_{k\in[d]}[2]$ by

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

For the maxterm statistic $\mathcal{F}_{\vee}$ we analogously have

$$\sigma^{\mathcal{F}_{\vee}}\left[X_{[d]}, L_{[d]}\right] = \mathbb{I}\left[X_{[d]}, L_{[d]}\right] - \delta\left[X_{[d]}, L_{[d]}\right]$$

and thus the mean parameter to any distribution $\mathbb{P}\left[X_{[d]}\right]$ has therefore the coordinates to $l_{[d]} \in \bigtimes_{k\in[d]}[2]$ by

$$\mu\left[L_{[d]} = l_{[d]}\right] = 1 - \mathbb{P}\left[X_{[d]} = l_{[d]}\right].$$

The claim on the forward mappings follows for

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{F}_{\vee}}\left[Y_{\vee}, X_{[d]}\right], \tau^{\mathrm{EL}}\left[Y_{\vee}\right]\right\rangle\left[X_{[d]}|\varnothing\right].$$

■

Any probability distribution can thus be represented by a Hybrid Logic Network in the minterm statistic, as well as in the maxterm statistic. Thus, we have identified sets of 2^d formulas, which is rich enough to fit any distribution.

9.2.1.2 Atomic Statistic

Let us now derive a closed form backward mapping for the statistic

$$\mathcal{F}_{[d]} \coloneqq \{f^k : k \in [d]\}$$

of atomic formulas, which coincides with a variable selection map. The selection encoding of this statistic is the tensor

$$\sigma^{\mathcal{F}_{[d]}} X_{[d]}, L = \sum_{k\in[d]} \epsilon_1\left[X_k\right] \otimes \mathbb{I}\left[X_{[d]/\{k\}}\right] \otimes \epsilon_k\left[L\right].$$

For each probability distribution $\mathbb{P}\left[X_{[d]}\right]$ we the corresponding mean parameter has coordinates at $k \in [d]$ by

$$\mu\left[L=k\right] = \mathbb{P}\left[X_k = 1\right].$$

Theorem 9.7 *For the Hybrid Logic Networks to the statistic of atomic formulas $\mathcal{F}_{[d]}$ we have the forward mapping*

$$F^{[d]}((A, y_A, \theta))[L=k] = \begin{cases} y_k & \text{if} \quad k \in A \\ \frac{\exp[\theta[L=k]]}{1+\exp[\theta[L=k]]} & \text{if} \quad k \notin A \end{cases}$$

Proof. Let $(A, y_A, \theta) \in \mathcal{P}_d$ and denote the

$$\mu\left[L\right] = F^{[d]}((A, y_A, \theta)).$$

By definition we have for any $k \in [d]$

$$\begin{aligned} \mu\left[L=k\right] &= \left\langle \sigma^{\mathcal{F}_{[d]}} X_{[d]}, L, \left\langle \beta^{\mathcal{F}_{[d]}}\left[Y_{[d]}, X_{[d]}\right], \xi^{(A,y_A,\theta)}\left[Y_{[d]}\right]\right\rangle\left[X_{[d]}|\varnothing\right]\right\rangle[L=k] \\ &= \left\langle \epsilon_1\left[X_k\right], \left\langle \beta^{f^k}\left[y_k, X_k\right], \xi^l Y_k\right\rangle\left[X_k|\varnothing\right]\right\rangle[\varnothing] \end{aligned}$$

Now, if $k \in A$ we have $\xi^l Y_k = \epsilon_{y_k}\left[Y_k\right]$ and $\mu\left[L=k\right] = y_k$. If $k \notin A$ then $\xi^l Y_k = \alpha^{k,\theta}\left[Y_k\right]$ and

$$\mu\left[L=k\right] = \frac{\exp\left[\theta\left[L=k\right]\right]}{1+\exp\left[\theta\left[L=k\right]\right]}.$$

■

A backward mapping to the atomic statistic is given by

$$B^{[d]}(\mu\left[L\right]) = (A, y_A, \theta) = \left(\{k : k \in [d]\mu\left[L=k\right] \in \{0,1\}\}, [\mu\left[L=k\right] :]\right)$$

where

- $A = \{k : k \in [d]\mu\left[L=k\right] \in \{0,1\}\}$
- For $k \in A$ $y_k = \mu\left[L=k\right]$
- For $k \in A$ we have $\theta\left[L=k\right] = 0$ and for $k \notin A$

$$\theta\left[L=k\right] = \ln\left[\frac{\mu\left[L=k\right]}{1-\mu\left[L=k\right]}\right]$$

The forward and backward mapping on the soft atomic formulas are the coordinatewise sigmoid and logit, respectively.

9.3 Alternating Moment Matching

While above we have investigated specific examples of backward mappings in closed form, we now derive a generic algorithm for the Maximum Likelihood Estimation in case of Hybrid Logic Networks. One can understand this algorithm as an iterative refinement of the approximation to the backward mapping. To derive the algorithm, we first solve local cross-entropy minimization problems, which are then alternated to find global solutions.

9.3.1 Local Updates

Let us now varying a distribution $\tilde{\mathbb{P}}[X_{[d]}]$ by adding an additional feature f

$$\mathbb{P}^{\xi}\left[X_{[d]}\right] \coloneqq \left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{f}\left[Y_f, X_{[d]}\right], \xi\left[Y_f\right]\right\rangle\left[X_{[d]}|\varnothing\right] .$$

Note, that the normalization exists for positive $\xi\left[Y_f\right]$ and if $\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing] \notin \{0,1\}$ then also for any non-vanishing $\xi\left[Y_f\right]$. If $\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing] \in \{0,1\}$, then the $\mathbb{P}^{\xi}$ is constant among $\xi\left[Y_f\right]$, when it exists.

We want to solve the local cross-entropy minimization problem

$$\min_{\xi} \mathbb{H}\left[\mathbb{P}^{D}, \mathbb{P}^{\xi}\right] .$$

If $\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing] \in \{0,1\}$ then the minimum is taken at any $\xi\left[Y_f\right]$ such that $\mathbb{P}^{\xi}$ exists.

Lemma 9.8 *Let $\mathbb{P}^{D}$, $\tilde{\mathbb{P}}$ be distributions and f a formula such that $\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing] \notin \{0,1\}$. If $\left\langle \mathbb{P}^{D}, f\right\rangle[\varnothing] \in \{0,1\}$ the solution of the local cross-entropy minimization is*

$$\xi\left[Y_f\right] = \epsilon_{\left\langle \mathbb{P}^{D}, f\right\rangle[\varnothing]}\left[Y_f\right]$$

If $\left\langle \mathbb{P}^{D}, f\right\rangle[\varnothing] \notin \{0,1\}$ the solution is

$$\xi\left[Y_f\right] = \alpha^{\theta}\left[Y_f\right]$$

where

$$\theta = \ln\left[\frac{\left\langle \mathbb{P}^{D}, f\right\rangle[\varnothing]}{\left(1-\left\langle \mathbb{P}^{D}, f\right\rangle[\varnothing]\right)} \cdot \frac{1-\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing]}{\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing]}\right] .$$

Proof. The cross entropy is decomposed into

$$\begin{aligned}\mathbb{H}\left[\mathbb{P}^{D}, \mathbb{P}^{\xi}\right] = \mathbb{H}\left[\mathbb{P}^{D}, \tilde{\mathbb{P}}\right] \\ + \left\langle \mathbb{P}^{D}\left[X_{[d]}\right], \beta^{f}\left[Y_f, X_{[d]}\right], -\ln\left[\xi\left[Y_f\right]\right]\right\rangle[\varnothing] \\ + \ln\left[\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{f}\left[Y_f, X_{[d]}\right], \xi\left[Y_f\right]\right\rangle[\varnothing]\right] .\end{aligned}$$

Since the first term is constant among ξ, we focus on the minimization of the second and third term. For each $\xi\left[Y_f\right]$ and its boolean support $\kappa\left[Y_f\right] \coloneqq \mathbb{I}_{>0}\left(\xi\left[Y_f\right]\right)$ we find $\lambda > 0$ and $\theta \in \mathbb{R}$ such that

$$\xi\left[Y_f\right] = \lambda \cdot \left\langle \kappa\left[Y_f\right], \alpha^{\theta}\right\rangle\left[Y_f\right] .$$

Given this parametrization we have

$$\begin{aligned}&\left\langle \mathbb{P}^{D}\left[X_{[d]}\right], \beta^{f}\left[Y_f, X_{[d]}\right], -\ln\left[\xi\left[Y_f\right]\right]\right\rangle[\varnothing] + \ln\left[\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{f}\left[Y_f, X_{[d]}\right], \xi\left[Y_f\right]\right\rangle[\varnothing]\right] \\ &= \left\langle \mathbb{P}^{D}\left[X_{[d]}\right], \beta^{f}\left[Y_f, X_{[d]}\right], -\ln\left[\kappa\left[Y_f\right]\right]\right\rangle[\varnothing] - \theta \cdot \left\langle \mathbb{P}^{D}\left[X_{[d]}\right], f\left[X_{[d]}\right]\right\rangle[\varnothing] \\ &\quad + \ln\left[\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{f}\left[Y_f, X_{[d]}\right], \kappa\left[Y_f\right], \alpha^{\theta}\left[Y_f\right]\right\rangle[\varnothing]\right] .\end{aligned}$$

The minimum over κ is taken at

$$\kappa\left[Y_f\right] = \begin{cases} \epsilon_1\left[Y_f\right] & \text{if} \quad \left\langle \mathbb{P}^D, f\right\rangle[\varnothing] = 1 \\ \epsilon_0\left[Y_f\right] & \text{if} \quad \left\langle \mathbb{P}^D, f\right\rangle[\varnothing] = 0 \\ \mathbb{I}\left[Y_f\right] & \text{else} \end{cases}.$$

If the optimal κ is not the trivial vector $\mathbb{I}\left[Y_f\right]$, the parameter $\theta \in \mathbb{R}$ does not influence the distribution and we can arrive at the claim when choosing $\theta = 0$. If the optimal κ is trivial, we optimize further over α^θ

$$\min_{\theta\in\mathbb{R}} \left\langle \mathbb{P}^D\left[X_{[d]}\right], \beta^f\left[Y_f, X_{[d]}\right], -\ln\left[\xi\left[Y_f\right]\right]\right\rangle[\varnothing] + \ln\left[\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^f\left[Y_f, X_{[d]}\right], \alpha^\theta\left[Y_f\right]\right\rangle[\varnothing]\right] ..$$

The derivation of the objective is

$$\begin{aligned} &\frac{\partial\left\langle \mathbb{P}^D\left[X_{[d]}\right], \beta^f\left[Y_f, X_{[d]}\right], -\ln\left[\xi\left[Y_f\right]\right]\right\rangle[\varnothing] + \ln\left[\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^f\left[Y_f, X_{[d]}\right], \alpha^\theta\left[Y_f\right]\right\rangle[\varnothing]\right]}{\partial\theta} \\ &= \left\langle \mathbb{P}^D\left[X_{[d]}\right], f\left[X_{[d]}\right]\right\rangle[\varnothing] - \left\langle \mathbb{P}^{\alpha^\theta}\left[X_{[d]}\right], f\left[X_{[d]}\right]\right\rangle[\varnothing] \\ &= \left\langle \mathbb{P}^D, f\right\rangle[\varnothing] - \frac{\exp[\theta]\cdot\left\langle \tilde{\mathbb{P}}[X_{[d]}], f\left[X_{[d]}\right]\right\rangle[\varnothing]}{\exp[\theta]\cdot\left\langle \tilde{\mathbb{P}}[X_{[d]}], f\left[X_{[d]}\right]\right\rangle[\varnothing] + (1 - \left\langle \tilde{\mathbb{P}}[X_{[d]}], f\left[X_{[d]}\right]\right\rangle[\varnothing])}. \end{aligned}$$

The derivative vanished thus at the unique minimum of the cross entropy at

$$\theta = \ln\left[\frac{\left\langle \mathbb{P}^D, f\right\rangle[\varnothing]}{(1 - \left\langle \mathbb{P}^D, f\right\rangle[\varnothing])} \cdot \frac{(1 - \left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing])}{\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing]}\right]. \qquad \blacksquare$$

9.3.1.1 Markov Logic Networks

In case of boolean statistics, we can provide a particular simple implementation of the Alternating Moment Matching Algorithm 4. In the following section we will then generalize to Hybrid Logic Networks by allowing hard cores.

Iteratively we update leg vectors of the activation tensor. This is the above local variation with

$$\tilde{\mathbb{P}}[X_{[d]}] = \left\langle \{\beta^{f_l} : l \in [p]\} \cup \{\alpha^{\tilde{l},\theta} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\}\right\rangle[Y_l|\varnothing]$$

To solve the moment matching condition at a formula f_l we do not have to compute the normalization constant, since we only require the quotient

$$\frac{1 - \left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing]}{\left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing]}.$$

The updated canonical coordinate is thus computed as

$$\theta[L = l] = \ln\left[\frac{\mu_D[L = l]}{(1 - \mu_D[L = l])} \cdot \frac{\tau[Y_l = 0]}{\tau[Y_l = 1]}\right]$$

where by $\tau[Y_l]$ we denote the contraction

$$\tau[Y_l] = \left\langle \{\beta^{f_l} : l \in [p]\} \cup \{\alpha^{\tilde{l},\theta} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\}\right\rangle[Y_l].$$

Algorithm 8 Alternating Moment Matching for Markov Logic Networks

Input: Mean parameter $\mu[L]$ with $\forall l \in [p] : \mu[L = l] \notin \{0, 1\}$, boolean statistic $\mathcal{F}$, base measure ν

Output: Canonical parameter $\theta[L]$, such that $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$ is the (approximative) moment projection of $\mathbb{P}^D$ onto $\Gamma^{\mathcal{S},\nu}$

Set $\theta[L] = 0[L]$
while Convergence criterion is not met **do**
  for all $l \in \tilde{\mathcal{V}}$ **do**
    Compute

$$\tau[Y_l] = \left\langle \{\beta^{f_l} : l \in [p]\} \cup \{\alpha^{\tilde{l},\theta} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\} \right\rangle [Y_l]$$

    Set

$$\theta[L = l] = \ln\left[\frac{\mu[L = l]}{(1 - \mu[L = l])} \cdot \frac{\tau[Y_l = 0]}{\tau[Y_l = 1]}\right]$$

  end for
end while
return $\theta[L]$

Note that while $\forall l \in [p] : \mu[L = l] \notin \{0, 1\}$ ensures the well-definedness of the update equations in Algorithm 8 (otherwise the update could not be computed), it is only a necessary but not always sufficient criterion for the existence of a solution. The moment matching conditions are simultaneously satisfiable, if and only if $\mu[L] \in (\mathcal{M}_{\mathcal{F},\nu})^\circ$.

The algorithm would be finished after a single pass through the loop, if the variables X_f are independent. This would be for example 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.

Since the likelihood is concave (see [KF09]), there are not local maxima the coordinate descent could run into and coordinate descent will give a monotonic improvement of the likelihood.

Solving Equation 9.3.1.1 requires inference of a current model by answering a query. This can be a bottleneck and circumvented by approximative inference, see e.g. CAMEL [Gan+08].

9.3.1.2 Hybrid Logic Networks

We now extend the alternating parameter estimation algorithm to Hybrid Logic Networks, by allowing for mean parameters on the interior of cube faces. To this end, we first find the smallest cube face containing the mean parameter $\mu[L]$ (see Lem. 8.25) and then run the algorithm on the exponential family on that face. In an alternative perspective, we first optimize the leg vectors to features with $\mu[L = l] \in \{0, 1\}$, which are then constant and left out in the further update sweeps. The local moment matching conditions of are satisfied simultanously for some $\theta[L]$, if and only if $\mu[L]$ is elementarily realizable (see Def. 8.29), that is $\mu[L]$ is on the interior of a cube face of $\mathcal{M}_{\mathcal{F},\mathbb{I}}$.

9.3.2 Iterative Proportional Fitting

In a special case of partition statistics we can find simultaneous updates to the canonical parameters. Partition statistics are boolean statistics $\mathcal{F}$ such that

$$\left\langle \sigma^{\mathcal{F}}\left[X_{[d]}, L\right] \right\rangle \left[X_{[d]}\right] .$$

Algorithm 9 Alternating Moment Matching for Hybrid Logic Networks

Input: Mean parameter $\mu[L]$

Output: Canonical parameter $\theta[L]$, such that $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$ is the (approximative) moment projection of $\mathbb{P}^D$ onto $\Lambda^{\mathcal{F},\mathrm{EL}}$

- $$A = \left\{ l : l \in [p],\, \mu[L = l] \in \{0,1\} \right\}$$

  and a tuple y_A with $y_l = \mu[L = l]$ for $l \in A$.
- Run Algorithm 8 (Alternating Moment Matching for Markov Logic Networks) with base measure $f(A, y_A)$ and statistic $\mathcal{F} = \{f_l : l \in A\}$ to get $\theta[L]$.

return $(A, y_A, \theta[L])$

We can thus understand them as a disjoint partition of the state set $\bigtimes_{k\in[d]}[m_k]$ into sets

$$\mathcal{U}^l := \left\{ x_{[d]} : x_{[d]} \in \bigtimes_{k\in[d]} [m_k],\, f_l\left[X_{[d]} = x_{[d]}\right] = 1 \right\}.$$

These statistics will be investigated in more detail in Sect. 14.5.3.

Lemma 9.9 *Given distributions* $\mathbb{P}^D\left[X_{[d]}\right], \tilde{\mathbb{P}}[X_{[d]}]$ *we build*

$$\mu_D[L] = \left\langle \mathbb{P}^D\left[X_{[d]}\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right] \right\rangle[L] \quad \textit{and} \quad \tilde{\mu}[L] = \left\langle \tilde{\mathbb{P}}[X_{[d]}], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right] \right\rangle[L]$$

and assume that $\mathbb{I}_{>0}(\mu_D) \models \mathbb{I}_{>0}(\tilde{\mu})$. *Let us vary the distribution* $\tilde{\mathbb{P}}[X_{[d]}]$ *by elementary tensors* $\xi\left[Y_{[p]}\right]$ *as*

$$\mathbb{P}^{\xi}\left[X_{[d]}\right] := \left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right] \right\rangle \left[X_{[d]} | \varnothing\right].$$

Then

$$\mathrm{argmin}_{\xi} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\xi}\right]$$

is solved at

$$\xi\left[Y_{[p]}\right] = \left\langle \kappa^{(A,0_A)}\left[Y_{[p]}\right], \alpha^{\theta}\left[Y_{[p]}\right] \right\rangle \left[Y_{[p]}\right]$$

where $A = \{l : \mu_D[L = l] = 0\}$ *and for* $l \in [p]/A$

$$\theta[L = l] = \ln\left[\frac{\mu_D[L = l]}{\tilde{\mu}[L = l]}\right].$$

Proof. We use Thm. 14.35 to compute

$$\begin{aligned}\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right] \right\rangle[L] &= \left\langle \tilde{\mathbb{P}}[X_{[d]}], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right], \exp[\theta[L]], \epsilon_{[p]/A}[L] \right\rangle[L] \\ &= \left\langle \tilde{\mu}[L], \exp[\theta[L]], \epsilon_{[p]/A}[L] \right\rangle[L] = \mu_D[L]\end{aligned}$$

Here $\epsilon_{[p]/A}[L]$ is the indicator vector, whether $l \notin A$ (see for more details Def. 14.1).

Using that $\mathcal{F}$ is a partition statistic we get

$$\begin{aligned}\langle \mu_D[L] \rangle [\varnothing] &= \left\langle \mathbb{P}^D, \sigma^{\mathcal{F}}\left[X_{[d]}, L\right] \right\rangle [\varnothing] \\ &= \left\langle \mathbb{P}^D \right\rangle [\varnothing] \\ &= 1\,.\end{aligned}$$

It follows that

$$\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right] \right\rangle [\varnothing] = 1$$

and due to trivial partition function term that

$$\mathbb{P}^{\xi}\left[X_{[d]}\right] = \left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \xi\left[Y_{[p]}\right] \right\rangle \left[X_{[d]}\right]\,.$$

We conclude

$$\left\langle \mathbb{P}^{\xi}\left[X_{[d]}\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right] \right\rangle [L] = \mu_D[L]\,.$$

Thus, the moment matching condition is satisfied for each $l \in [p]$ and the cross-entropy is minimized. ■

If the $\mathbb{I}_{>0}(\mu_D) \models \mathbb{I}_{>0}(\tilde{\mu})$ is violated, then there is no solution to the moment matching condition, since the support of the mean parameter cannot be increased by an activation tensor.

The assumptions of a partition statistic 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 referred to as Iterative Proportional Fitting [WJ08].

9.4 Structure Learning

Structure learning refers to the learning of the statistic $\mathcal{F}$ itself. Let there be a set $\mathcal{H}$ of statistics we build the set of parametrized distributions

$$\bigcup_{\mathcal{F}\in\mathcal{H}} \Lambda^{\mathcal{F},\mathrm{EL}}$$

and pose the structure learning problem as the minimization of the cross entropy

$$\min_{\mathcal{F}\in\mathcal{H}} \min_{(A,y_A,\theta)\in\mathcal{P}_p} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right]\,.$$

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. To avoid intractabilities, one can choose a greedy approach and learn in addition formulas f when already having learned a set $\mathcal{F}$ of formulas.

9.4.1 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

$$\min_{f\in\mathcal{H}} \min_{(A,y_A,\theta)\in\mathcal{P}_{|\mathcal{F}|+1}} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F}\cup\{f\},(A,y_A,\theta)}\right] . \tag{$\mathrm{P}^{\mathrm{greedy}}_{D,\mathcal{F},\mathcal{H}}$}$$

A brute force solution of Problem ($\mathrm{P}^{\mathrm{greedy}}_{D,\mathcal{F},\mathcal{H}}$) would require parameter estimation for each candidate in $\mathcal{H}$. We provide two more efficient approximative heuristics in the following (see Chapter 20 in [KF09]), which are faster to compute estimates of the cross entropy improvement from adding a formula to an existing statistic.

9.4.2 Gain Heuristic

In the gain heuristic, only the parameters of the new formula are optimized and the others left unchanged. Let $(\tilde{A}, \tilde{y}_{\tilde{A}}, \tilde{\theta})$ be the canonical parameter of the reference distribution on the statistic $\mathcal{F}$. When adding a feature $f \in \mathcal{H}$ we extend $\mathcal{F}$ by f and restrict the parameters to coincide with $(\tilde{A}, \tilde{y}_{\tilde{A}}, \tilde{\theta})$ on the first $|\mathcal{F}|$ coordinates. We denote this constraint by

$$(A, y_A, \theta)|_{[|\mathcal{F}|]} = (\tilde{A}, \tilde{y}_{\tilde{A}}, \tilde{\theta}) .$$

The greedy gain heuristic is then the problem

$$\min_{f\in\mathcal{H}} \min_{\mathcal{P}_{|\mathcal{F}|+1}\,:\,(A,y_A,\theta)|_{[|\mathcal{F}|]}=(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta})} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(A,y_A,\theta)}\right] . \tag{$\mathrm{P}^{\mathrm{gain}}_{D,\mathcal{F},(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta}),\mathcal{H}}$}$$

To provide further insight into the solution of the gain heuristic, let us quantify the improvement of the cross entropy when adding a feature f.

Lemma 9.10 *Let $\mathbb{P}^D$, $\tilde{\mathbb{P}}$ be distributions and f a formula such that $\left\langle \tilde{\mathbb{P}}, f \right\rangle [\varnothing] \notin \{0,1\}$. Then we have*

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F},(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta})}\right] - \min_{\xi} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\xi}\right] = \mathrm{D}_{\mathrm{KL}}\left[B(\left\langle \mathbb{P}^D, f \right\rangle [\varnothing]) || B(\left\langle \tilde{\mathbb{P}}, f \right\rangle [\varnothing])\right]$$

where by $B(p)$ we denote the Bernoulli distribution with parameter $p \in [0,1]$.

Proof. We use the characterization of the local update by Lem. 9.8 for $\tilde{\mathbb{P}} = \mathbb{P}^{\mathcal{F},(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta})}$. The update on the added feature is then parametrized by the two-dimensional vector $\xi\left[Y_f\right]$ and we have

$$\mathbb{P}^{\mathcal{F},(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta})} = \mathbb{P}^{\mathbb{I}} .$$

Let us abbreviate $\mu_D := \left\langle \mathbb{P}^D, f \right\rangle [\varnothing]$ and $\tilde{\mu} := \left\langle \mathbb{P}^{\mathcal{F},(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta})}, f \right\rangle [\varnothing]$. We distinguish the cases $\mu_D \in (0,1)$ and $\mu_D \in \{0,1\}$.

In the case $\mu_D \in (0,1)$, we have $A = \tilde{A}$, $y_A = \tilde{y}_A$ and $\theta\,[L = l] = \tilde{\theta}\,[L = l]$ for $l \in [|\mathcal{F}|]$. The additional coordinate of the canonical parameter is then

$$\theta\,[L = |\mathcal{F}|] = \ln\left[\frac{\mu_D}{(1-\mu_D)} \cdot \frac{1-\tilde{\mu}}{\tilde{\mu}}\right] .$$

Based on this characterization, the cross entropy difference is

$$\begin{aligned}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathbb{I}}\right] - \min_{\xi} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\xi}\right] \\ = \mu_D \cdot \theta\,[L = |\mathcal{F}|] - \ln\left[\left\langle \tilde{\mathbb{P}}, \beta^f\left[Y_f, X_{[d]}\right], \alpha^{|\mathcal{F}|,\theta}\left[Y_f\right]\right\rangle [\varnothing]\right] .\end{aligned}$$

We simplify

$$\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^f \left[Y_f, X_{[d]} \right], \alpha^{|\mathcal{F}|,\theta} \left[Y_f \right] \right\rangle [\varnothing] = (1-\tilde{\mu}) + \tilde{\mu} \cdot \exp\left[\theta \left[L = |\mathcal{F}|\right]\right] = \frac{(1-\tilde{\mu})}{(1-\mu_D)}.$$

We further have

$$\mu_D \cdot \theta \left[L = |\mathcal{F}|\right] = \mu_D \cdot \left[\ln \left[\frac{\mu_D}{(1-\mu_D)} \cdot \frac{(1-\tilde{\mu})}{\tilde{\mu}} \right] \right] = \mu_D \ln[\mu_D] - \mu_D \ln[1-\mu_D] + \mu_D \ln[1-\tilde{\mu}] - \mu_D \ln[\tilde{\mu}]$$

and arrive at

$$\begin{aligned}
&\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathbb{I}}\right] - \min_{\xi} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\xi}\right] \\
&= \mu_D \ln[\mu_D] - \mu_D \ln[1-\mu_D] + \mu_D \ln[1-\tilde{\mu}] - \mu_D \ln[\tilde{\mu}] - \ln[1-\tilde{\mu}] - \ln[1-\mu_D] \\
&= (-\mu_D \ln[\tilde{\mu}] - (1-\mu_D)\ln[1-\tilde{\mu}]) - (-\mu_D \ln[\mu_D] - (1-\mu_D)\ln[1-\mu_D]) \\
&= \mathrm{D}_{\mathrm{KL}}\left[B(\mu_D) || B(\tilde{\mu})\right].
\end{aligned}$$

In the case $\mu_D \in \{0,1\}$, the optimal ξ is boolean and only the partition function term is changed by the update. We then have

$$\begin{aligned}
\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathbb{I}}\right] - \min_{\xi} \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\xi}\right] &= -\ln\left[\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^f \left[Y_f, X_{[d]} \right], \xi \left[Y_f \right] \right\rangle [\varnothing]\right] \\
&= \begin{cases} \ln[\tilde{\mu}] & \text{if} \quad \mu_D = 0 \\ \ln[1-\tilde{\mu}] & \text{if} \quad \mu_D = 1 \end{cases} \\
&= \mathrm{D}_{\mathrm{KL}}\left[B(\mu_D) || B(\tilde{\mu})\right].
\end{aligned}$$

■

Problem $(\mathrm{P}^{\mathrm{gain}}_{D,\mathcal{F},(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta}),\mathcal{H}})$ is thus solved for

$$\hat{f} \in \mathrm{argmax}_{f \in \mathcal{F}} \mathrm{D}_{\mathrm{KL}}\left[B(\left\langle \mathbb{P}^D, f \right\rangle [\varnothing]) || B(\langle \tilde{\mathbb{P}}, f \rangle [\varnothing])\right]$$

and (A, y_A, θ) characterized in Lem. 9.8. The minimum is taken at

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F},(\tilde{A},\tilde{y}_{\tilde{A}},\tilde{\theta})}\right] - \mathrm{D}_{\mathrm{KL}}\left[B(\left\langle \mathbb{P}^D, \hat{f} \right\rangle [\varnothing]) || B(\left\langle \tilde{\mathbb{P}}, \hat{f} \right\rangle [\varnothing])\right].$$

The gain heuristic thus searches for the mode of 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.

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. Unfortunately, backward mappings cannot be expressed in closed form in general.

9.4.3 Gradient Heuristic

Gradient heuristic is another approach to select a feature. We show here how the effective selection tensor network representation of exponentially many formulas described in Chapter 7 can be utilized.

In the gradient heuristic, we estimate the cross entropy improvement by the gradient of the

cross entropy with respect to varying with another feature. Given a distribution $\tilde{\mathbb{P}}[X_{[d]}]$, we vary

$$\mathbb{P}^{\alpha^\theta}\left[X_{[d]}\right] \coloneqq \left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^f\left[Y_f, X_{[d]}\right], \alpha^\theta\left[Y_{Y_f}\right]\right\rangle\left[X_{[d]}|\varnothing\right] .$$

In the gradient heuristic we choose the steepest gradient direction

$$\min_{f\in\mathcal{H}} \frac{\partial \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\alpha^\theta}\left[X_{[d]}\right]\right]}{\partial\theta} . \tag{$\mathrm{P}^{\mathrm{grad}}_{D,\tilde{\mathbb{P}},\mathcal{H}}$}$$

Lemma 9.11 *For any formula f we have*

$$\frac{\partial \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\alpha^\theta}\right]}{\partial\theta} = -\left\langle \mathbb{P}^D, f\right\rangle[\varnothing] + \left\langle \mathbb{P}^{\alpha^\theta}, f\right\rangle[\varnothing] .$$

Proof. We have

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\alpha^\theta}\right] = \left\langle \mathbb{P}^D, \beta^f\left[Y_f, X_{[d]}\right], -\ln\left[\alpha^\theta\left[Y_{Y_f}\right]\right]\right\rangle[\varnothing] + \ln\left[\left\langle \tilde{\mathbb{P}}[X_{[d]}], \beta^f\left[Y_f, X_{[d]}\right], \alpha^\theta\left[Y_{Y_f}\right]\right\rangle[\varnothing]\right]$$

The derivation of the first term at $\theta = 0$ is $-\left\langle \mathbb{P}^D, f\right\rangle[\varnothing]$ and of the second $\left\langle \mathbb{P}^{\alpha^\theta}, f\right\rangle[\varnothing]$. ∎

Problem ($\mathrm{P}^{\mathrm{grad}}_{D,\tilde{\mathbb{P}},\mathcal{H}}$) is thus

$$\min_{f\in\mathcal{H}} \left\langle \tilde{\mathbb{P}}, f\right\rangle[\varnothing] - \left\langle \mathbb{P}^D, f\right\rangle[\varnothing] .$$

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.

9.4.4 Proposal Distributions

We now frame the selection of a formula as a sampling problem of proposal distributions. Each of the scores to candidate formulas are understood as coordinates of an energy tensor. Among the heuristics, the gradient heuristic has the most accessible form, since the energy tensor is available as a tensor network of the selection encoding of the formulas.

Definition 9.12 (Gradient Heuristic Proposal Distribution) Let there be a base distribution $\tilde{\mathbb{P}}$, a targeted distribution $\mathbb{P}^D$ and a formula selecting map $\mathcal{H}$. The proposal distribution at inverse temperature $\beta > 0$ is the distribution of L defined by

$$\left\langle \exp\left[\left\langle \beta\cdot(\mathbb{P}^D\left[X_{[d]}\right] - \tilde{\mathbb{P}}\left[X_{[d]}\right]), \mathcal{H}\left[X_{[d]}, L\right]\right\rangle[L]\right]\right\rangle[L|\varnothing] .$$

The proposal distribution is the member of the exponential family with statistics $\mathcal{H}$ and canonical 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 gradient heuristic optimization Problem ($\mathrm{P}^{\mathrm{grad}}_{D,\tilde{\mathbb{P}},\mathcal{H}}$) is the search for the mode of the proposal distribution. To solve the gradient heuristic, we thus need to answer a mode query, for which we can apply the methods introduced in Chapter 3, such as Gibbs Sampling or Mean Field Approximations in combination with annealing.

The mean parameter polytope of any proposal distribution to statistic $\mathcal{F}^T$ is the convex hull of the formulas in $\mathcal{F}$, that is

$$\mathcal{M}_{\mathcal{F}^T} = \mathrm{conv}\left(\sigma^{\mathcal{F}^T} L = l, X_{[d]} \,:\, l \in [p]\right) = \mathrm{conv}\left(f\left[X_{[d]}\right] \,:\, f \in \mathcal{H}\right)$$

As it was the case for Markov Logic Networks, the mean parameter polytopes are instances of a 0/1-polytopes [Zie00; Gil07]. The vertices are the formulas selectable by the formula selecting map $\mathcal{H}$.

9.4.5 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 10, which is a greedy algorithm adding iteratively the currently best feature.

Algorithm 10 Greedy Structure Learning

Input: Empirical distribution $\mathbb{P}^D$, hypothesis $\mathcal{H}$ of formulas
Output: Distribution $\mathbb{P}^{(\mathcal{S},\theta,\nu)}$ approximating $\mathbb{P}^D$

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 ($\mathrm{P}^{\mathrm{greedy}}_{D,\mathcal{F},\mathcal{H}}$) 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 $\theta\,[l = p] = 0$
- **Weight Estimation:** Estimate the best weights for the added formula and recalibrate the weights of the previous formulas, by calling Algorithm 9.

  $$\tilde{\mathbb{P}} \leftarrow \mathbb{P}^{\mathcal{F},(A,y_A,\theta)}$$

end while
return $\mathcal{F}$, (A, y_A, θ)

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 multiple 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 9.13 (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. $\Diamond$

10

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 3 and Chapter 9 are robust against such effects.

10.1 Preparations

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.

10.1.1 Fluctuation of the Empirical Distribution

When drawing random states $D(j) \in \bigtimes_{k\in[d]}[m_k]$ by a distribution $\mathbb{P}^*$, we use the one-hot encoding to forward each random state to the random tensor

$$\epsilon_{D(j)}\left[X_{[d]}\right] .$$

The expectation of this random tensor is

$$\mathbb{E}\left[\epsilon_{D(j)}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \mathbb{P}^*\left[X_{[d]} = x_{[d]}\right] \epsilon_{x_{[d]}}\left[X_{[d]}\right] = \mathbb{P}^*\left[X_{[d]}\right] .$$

The empirical distribution is then the average of independent random one-hot encodings, namely the random tensor

$$\mathbb{P}^D\left[X_{[d]}\right] = \frac{1}{m}\sum_{j\in[m]} \epsilon_{D(j)}\left[X_{[d]}\right] .$$

To avoid confusion let us strengthen, that in this chapter we interpret $\mathbb{P}^D$ as a random tensor taking values in $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$, whereas each supported value of $\mathbb{P}^D$ is an empirical distribution taking values in $\bigtimes_{k\in[d]}[m_k]$. The forwarding of $\bigtimes_{k\in[d]}[m_k]$ under the one-hot encoding is a multinomial random variable, see Thm. 10.14.

When the marginal of each datapoint is $\mathbb{P}^*$, the expectation of the empirical distribution is

$$\mathbb{E}\left[\mathbb{P}^D\right] = \frac{1}{m}\sum_{j\in[m]} \mathbb{E}\left[\epsilon_{D(j)}\right] = \mathbb{P}^* .$$

From the law of large numbers it follows, that in the limit of $m \to \infty$ at any coordinate $x \in \bigtimes_{k\in[d]}[m_k]$ almost everywhere

$$\mathbb{P}^D\left[X_{[d]} = x_{[d]}\right] \to \mathbb{E}\left[\mathbb{P}^D\left[X_{[d]} = x_{[d]}\right]\right] = \mathbb{P}^*\left[X_{[d]} = x_{[d]}\right] .$$

At finite m the empirical distribution differs from the by the difference

$$\mathbb{P}^D - \mathbb{P}^*$$

which we call a fluctuation tensor.

10.1.2 Mean Parameter of the Empirical Distribution

We now investigate the empirical mean parameter

$$\mu_D[L] = \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \mathbb{P}^D\left[X_{[d]}\right]\right\rangle[L] \ .$$

Each coordinate of μ_D is decomposed as

$$\mu_D[L = l] = \frac{1}{m}\sum_{j\in[m]} \mathcal{S}_l[D(j)]$$

and thus stores the empirical average of the feature s_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 10.1 *When drawing data independently from $\mathbb{P}^*$, we have $\mathbb{E}[\mu_D[L]] = \mu^*[L]$, where we call*

$$\mu^*[L] = \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \mathbb{P}^D\left[X_{[d]}\right]\right\rangle[L]$$

the expected mean parameter.

Proof. Since the expectation commutes with linear functions. ∎

For each $l \in [p]$ the law of large numbers guarantees that $\mu_D[L = l]$ converges almost surely against $\mu^*[L = l]$ when $m \to \infty$. To utilize these we need to approach the following issues:

- **Non-asymptotic bounds:** We need non-asymptotic convergence bounds, since one has access to finite data when learning
- **Uniform convergence:** The convergence has to happen uniformly for all $l \in [p]$.
- **Interpretability:** 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.

10.1.3 Face Recovery

Lemma 10.2 *Let $\left((x_0^j, \ldots, x_{d-1}^j) : j \in [m]\right)$ be drawn from a distribution $\mathbb{P}^*$ with mean parameter $\mu^*[L]$ and let us denote by $Q^{\mathcal{T}}_{\mathcal{S},\nu}$ the unique face of $\mathcal{M}_{\mathcal{S},\nu}$, such that $\mu^*[L] \in \left(Q^{\mathcal{T}}_{\mathcal{S},\nu}\right)^\circ$. We always have $\mu_D[L] \in Q^{\mathcal{T}}_{\mathcal{S},\nu}$. If and only if $\mu_D[L] \in \left(Q^{\mathcal{T}}_{\mathcal{S},\nu}\right)^\circ$, then the support of $\mathbb{P}^{(\mathcal{S},\theta_D,\nu)}$ coincides with the face measure of $Q^{\mathcal{T}}_{\mathcal{S},\nu}$.*

Proof. Since $\left((x_0^j, \ldots, x_{d-1}^j) : j \in [m]\right)$ is drawn from $\mathbb{P}^*$, the support of $\mathbb{P}^D$ is contained in the support of $\mathbb{P}^*$. Thus, the support of $\mathbb{P}^D$ is contained in the support of the face measure to $Q^{\mathcal{T}}_{\mathcal{S},\nu}$ and $\mu_D[L] \in Q^{\mathcal{T}}_{\mathcal{S},\nu}$. The second claim follows from the fact, that the maximum entropy distribution

reproducing θ_D is positive with respect to the face measure of the face containing $\mu_D[L]$ in its relative interior. ∎

To find probabilistic guarantees on the recovery of the support of the generating distribution, we thus need to show probabilistic bounds on the event $\mu_D[L] \in \left(Q^{\mathcal{I}}_{\mathcal{S},\nu}\right)^{\circ}$. We call this event *face recovery*.

10.1.4 Mode Recovery

Another interesting recovery scenario is whether the mode of the approximating distribution coincides with the mode of the generating distribution. This event is called *mode recovery*.

Let $\hat{\theta}$ be the estimator of the canonical parameter θ_*, then the mode set of both coincide, if and only if they are elements in the same max cone (see Sect. 3.6.2), i.e.

$$C^{\hat{\theta}}_{\mathcal{S},\nu} = C^{\theta_*}_{\mathcal{S},\nu}\,.$$

Equivalently, this is the case if $\mu^*[L]$ and $\mu_D[L]$ are elements of the image of the same normal cone under the forward mapping.

10.1.5 Noise Tensor and its Width

Motivated by Thm. 10.1, we build our derivation of probabilistic guarantees on non-asymptotic and uniform convergence bounds for $\mu_D[L]$. Let us first define the fluctuations of the empirical mean parameter, when drawing the data independently from a random distribution, as the noise tensor.

Definition 10.3 Given a statistic $\mathcal{S}$, $m \in \mathbb{N}$ and a distribution $\mathbb{P}^*$, we call

$$\eta^{\mathcal{S},\mathbb{P}^*,m} = \left\langle (\mathbb{P}^D - \mathbb{P}^*), \sigma^{\mathcal{S}} \right\rangle [L]$$

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 universal 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 $\eta^{\mathcal{S},\mathbb{P}^*,m}$ is the centered mean parameter to the empirical distribution, that is

$$\mu_D - \mathbb{E}[\mu_D] = \left\langle \sigma^{\mathcal{S}}, \mathbb{P}^D - \mathbb{P}^* \right\rangle [L]\,.$$

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 10.4 Given a set $\Gamma \subset \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ and $\eta^{\mathcal{S},\mathbb{P}^*,m}$ a random tensor taking values in $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we define the width as the random variable

$$\Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) = \sup_{\theta\in\Gamma} \left| \left\langle \theta, \eta^{\mathcal{S},\mathbb{P}^*,m} \right\rangle [\varnothing] \right|\,.$$

Bounds on the widths are also called uniform concentration bounds [Goe21] and generic probabilistic bounds will be provided in Sect. 10.4.

Atomic norms induce metrics, which are widths (see [Cha+12] and Chapter 5 in [Goe21]).

Definition 10.5 Given a set $\Gamma \subset \mathbb{R}^p$ such that an open neighborhood of the origin is contained in conv (Γ). Then

$$\|\theta\|_\Gamma = \Omega_\Gamma(\theta)$$

is the dual atomic norm and

$$d_\Gamma(\theta, \tilde{\theta}) = \Omega_\Gamma(\theta - \tilde{\theta})$$

is the dual atomic distance to Γ.

Examples of atomic norms are:

- Euclidean distance ℓ_2, when $\Gamma = \mathbb{S}$.
- Supremum distance ℓ_∞, when $\Gamma = \{\lambda \cdot \epsilon_l[L] : \lambda \in \{-1, +1\}, l \in [p]\}$.

10.2 Generic Guarantees based on Noise Widths

We now derive error bounds for parameter estimation and structure learning, as introduced in Chapter 9. When combined with probabilistic bounds on the noise width, they are probabilistic success guarantees.

10.2.1 Parameter Estimation in Exponential Families

Parameter Estimation is the M-projection of the empirical distribution onto an exponential family. In Chapter 3 we have characterized those by the backward mapping acting on the mean parameter. Thus, while we are interested in the expected canonical parameter

$$\theta_*[L] = B^{(\mathcal{S},\nu)}(\mu^*[L])$$

we get an estimation by the empirical canonical parameter

$$\theta_D[L] = B^{(\mathcal{S},\nu)}(\mu_D[L]).$$

Unfortunately, since the backward mapping is not linear, we in general do not have that $\mathbb{E}\left[B^{(\mathcal{S},\nu)}(\mu_D)\right]$ coincides with $B^{(\mathcal{S},\nu)}(\mu^*)$. To build intuition on the concentration we recall the expression of the backward mapping as

$$B^{(\mathcal{S},\nu)}(\mu) = \operatorname{argmax}_\theta - \mathbb{H}\left[\mathbb{P}^\mu, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]$$

where $\mathbb{P}^\mu$ is any distribution reproducing the mean parameter. We want to compare the solutions $B^{(\mathcal{S},\nu)}(\mu_D)$ and $B^{(\mathcal{S},\nu)}(\mu^*)$, in which case $\mathbb{P}^\mu$ can be chosen as $\mathbb{P}^D$ and $\mathbb{P}^*$. It is common to call the objectives $\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]$ and $\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]$ empirical and expected risk [SB14]. Since the empirical risk has a linear dependence on μ_D, we have at each θ

$$\begin{aligned}
\mathbb{E}\left[\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]\right] &= \mathbb{E}\left[\langle \mu_D, \theta\rangle[\varnothing] - A^{(\mathcal{S},\nu)}(\theta)\right] \\
&= \langle \mathbb{E}[\mu_D], \theta\rangle[\varnothing] - A^{(\mathcal{S},\nu)}(\theta) \\
&= \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]
\end{aligned}$$

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 mapping 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 10.6 *For any Γ and D we have*

$$\Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) = \sup_{\theta\in\Gamma}\left|\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]\right| .$$

Proof. For any $\theta \in \Gamma$ and by $\mathbb{P}^\mu$ realizable mean parameter μ we have

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

It follows that

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right] = -\left\langle (\mu_D - \mu^*), \theta\right\rangle[\varnothing]$$

and the claim follows from comparison with Def. 10.3 and Def. 10.4. ∎

As a direct consequence, we have at any $\theta \in \Gamma$

$$\left|\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]\right| \leq \Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) .$$

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

$$\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] \leq \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] + \Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) .$$

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 10.7 *Let us assume that for $\theta_* \in \Gamma$ we have $\mathbb{P}^* = \mathbb{P}^{(\mathcal{S},\theta_*,\nu)}$. Then for any solution θ_D of the empirical problem*

$$\mathrm{argmin}_{\theta\in\Gamma}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta,\nu)}\right]$$

we have

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}||\mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] \leq 2\Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) .$$

Proof. For the solution θ_D of the empirical risk minimization on Γ we have since $\theta_* \in \Gamma$ that

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] \leq \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_*,\nu)}\right] .$$

It follows that

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}||\mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] \leq \mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}||\mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] + \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_*,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right]$$

$$= \left(\mathbb{H}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right]\right) - \left(\mathbb{H}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}, \mathbb{P}^{(\mathcal{S},\theta_*,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_*,\nu)}\right]\right) ,$$

where we expanded the KL-divergence as a difference of cross entropies. We apply Lem. 10.6 to estimate the terms in brackets and get

$$\left(\mathbb{H}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right]\right) - \left(\mathbb{H}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}, \mathbb{P}^{(\mathcal{S},\theta_*,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_*,\nu)}\right]\right) \leq 2\Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) .$$

Combined with the above inequality we arrive at

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{(\mathcal{S},\theta_*,\nu)} || \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] \leq 2\Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) . \qquad \blacksquare$$

One technical issue arises from the fact, that when we allow for $\Gamma = \mathbb{R}^p$, then $\Omega_\Gamma\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right)$ 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 $\mathbb{S} \subset \mathbb{R}^p$ we have

$$\|\mu_D - \mu^*\|_2 = \Omega_{\mathbb{S}}\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right) ,$$

We apply this insight to state the following guarantee for unconstrained parameter estimation.

Theorem 10.8 *For any $\theta_D \in \mathbb{R}^p$ we have*

$$\left|\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right]\right| \leq \Omega_{\mathbb{S}}\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right) \cdot \|\theta_D\|_2 .$$

Proof. As in the proof of Lem. 10.6 we use that

$$\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{S},\theta_D,\nu)}\right] = \langle \mu_D - \mu^*, \theta_D \rangle [\varnothing] .$$

By Cauchy-Schwartz we further have

$$|\langle \mu_D - \mu^*, \theta_D \rangle [\varnothing]| \leq \|\mu_D - \mu^*\|_2 \cdot \|\theta_D\|_2 .$$

Using that $\|\mu_D - \mu^*\|_2 = \Omega_{\mathbb{S}}\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right)$ we arrive at the claim. $\blacksquare$

10.2.2 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 10.9 The max gap of a tensor $\tau\left[X_{[d]}\right]$ is the quantity

$$\Delta(\tau) = \left(\max_{x_{[d]}} \tau\left[X_{[d]} = x_{[d]}\right]\right) - \left(\max_{x_{[d]} \notin \operatorname{argmax}_{x_{[d]}} \tau\left[X_{[d]} = x_{[d]}\right]} \tau\left[X_{[d]} = x_{[d]}\right]\right) .$$

When comparing the gap with the noise width, we get the following guarantee.

Theorem 10.10 *Whenever*

$$\Delta(\mu^*) > 2 \cdot \Omega_{\{\epsilon_{x_{[d]}} : x_{[d]} \in \times_{k\in[d]}[m_k]\}}\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) ,$$

then any mode $x_{[d]}$ of the empirical proposal distribution is a mode of the expected proposal distribution.

Proof. Let us assume that for a mode $l^D \in \operatorname{argmax}_{l\in[p]} \mu_D[L = l]$ of the empirical mean parameter we have

$$l^D \notin \operatorname{argmax}_{l\in[p]} \mu^*[L = l] .$$

For a mode $l^* \in \operatorname{argmax}_{l\in[p]} \mu^*[L = l]$ of the expected mean parameter we then have

$$\mu^*[L = l^D] \leq \mu^*[L = l^*] - \Delta(\mu^*)$$

and

$$\mu_D\left[L = l^D\right] \geq \mu_D[L = l^*] .$$

Comparing both inequalities we get

$$\left(\mu_D\left[L = l^D\right] - \mu^*[L = l^D]\right) + \left(-\mu_D[L = l^*] + \mu^*[L = l^*]\right) \geq \Delta(\mu^*) .$$

Estimating the terms in the bracket by the width of the noise tensor with respect to basis vectors, we get

$$2 \cdot \Omega_{\{\epsilon_{x_{[d]}} : x_{[d]} \in \times_{k\in[d]}[m_k]\}}\left(\eta^{\mathcal{S},\mathbb{P}^*,m}\right) \geq \Delta(\mu^*) ,$$

which is a contradiction to the assumption. Thus, any mode of the empirical mean parameter is also a model of the expected mean parameter. ∎

10.2.3 Mode Recovery

Let us now consider the more general mode recovery problem. To find a probabilistic bound on the mode recovery event, we generalize the gap at θ_* as the minimal distance to other cones and bound uniform concentration events implying that the distance between $\hat{\theta}$ and θ_* is smaller than the gap.

Definition 10.11 Let $d(\cdot,\cdot)$ be a metric on $\mathbb{R}^p$, then the generalized gap of $\theta \in \mathbb{R}^p$ is defined as

$$\Delta_d(\theta) = \inf_{\tilde{\theta} \notin C^{\theta}_{\mathcal{S},\nu}} d\left(\tilde{\theta}, \theta\right) .$$

Let us now show, that the mode recovery event holds, whenever the width of the atom set Γ is smaller than the generalized gap.

Theorem 10.12 *Let $\Gamma \subset \mathbb{R}^p$ induce an atomic norm. If*

$$\Omega_\Gamma\left(\hat{\theta} - \theta_*\right) < \Delta_{d_\Gamma}\left(\theta_*\right)$$

then the modes of $\mathbb{P}^{(\mathcal{S},\hat{\theta},\nu)}$ and $\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}$ coincide.

Proof. If $\hat{\theta} \notin C^{\theta_*}_{\mathcal{S},\nu}$, then

$$\Delta_{d_\Gamma}\left(\theta_*\right) \geq d_\Gamma\left(\hat{\theta}, \theta_*\right) = \Omega_\Gamma\left(\hat{\theta} - \theta_*\right) ,$$

which contradicts the assumption. Therefore if the assumption holds, then $\hat{\theta} \in C^{\theta_*}_{\mathcal{S},\nu}$ and the modes of $\mathbb{P}^{(\mathcal{S},\hat{\theta},\nu)}$ and $\mathbb{P}^{(\mathcal{S},\theta_*,\nu)}$ coincide. ∎

The guarantee on structure learning is the special case, where $\Gamma = \{\lambda \cdot \epsilon_l\left[L\right] : \lambda \in \{-1, +1\}, l \in [p]\}$ and

$$\Delta_{d_\Gamma}\left(\theta\right) = \frac{1}{2} \max_{l \notin \operatorname{argmax}_l \theta[L=l]} \left| \theta\left[L = l\right] - \max_l \theta\left[L = l\right] \right| .$$

10.3 Probabilistic Guarantees for Hybrid Logic Networks

For Hybrid Logic Networks we have statistics consistent of boolean statistics f_l, which can be interpreted as formulas in propositional logics. In this case the marginal distributions of the coordinates of $\eta^{\mathcal{S},\mathbb{P}^*,m}$ are scaled and centered binomials, which we show now.

Theorem 10.13 *For any $\mathcal{F}$ the marginal distribution of the coordinate $\eta^{\mathcal{F},\mathbb{P}^*,m}\left[L = l\right]$ 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

$$\epsilon_{D(j)}\left[X_{[d]}\right]$$

and since $\operatorname{im}\left(s_l\right) \subset \{0, 1\}$ the contraction

$$\left\langle s_l, \epsilon_{D(j)}\left[X_{[d]}\right]\right\rangle\left[\varnothing\right]$$

is a random variable taking values in $\{0, 1\}$. The variable therefore follows a Bernoulli distribution with mean parameter

$$\mu\left[L = l\right] = \mathbb{E}\left[\left\langle s_l, \epsilon_{D(j)}\left[X_{[d]}\right]\right\rangle\left[\varnothing\right]\right] = \left\langle s_l, \mathbb{P}^*\right\rangle\left[\varnothing\right]$$ ∎

The mean parameter of the M-projection of the empirical distribution on the family of Markov

Logic Networks with statistic $\mathcal{F}$ is the random tensor

$$\mu_D[L] = \left\langle \sigma^{\mathcal{F}}, \mathbb{P}^D \right\rangle [L] .$$

The expectation of this random tensor is

$$\mathbb{E}[\mu_D] = \left\langle \sigma^{\mathcal{F}}, \mathbb{E}\left[\mathbb{P}^D\right] \right\rangle [L] = \left\langle \sigma^{\mathcal{F}}, \mathbb{P}^* \right\rangle [L] = \mu^* ,$$

where we used that the expectation and contraction operation can be commuted due to the multilinearity of contractions.

10.3.1 Energy Tensor of 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

$$\begin{aligned}\mathbb{E}\left[\phi^{\mathcal{F}^T, \mathbb{P}^D - \tilde{\mathbb{P}}}\right] &= \mathbb{E}\left[\left\langle \sigma^{\mathcal{F}^T}, \mathbb{P}^D - \tilde{\mathbb{P}} \right\rangle [L]\right] = \left\langle \sigma^{\mathcal{F}^T}, \mathbb{E}\left[\mathbb{P}^D - \tilde{\mathbb{P}}\right] \right\rangle [L] = \left\langle \sigma^{\mathcal{F}^T}, \mathbb{P}^* - \tilde{\mathbb{P}} \right\rangle [L] \\ &= \mathbb{E}\left[\phi^{\mathcal{F}^T, \mathbb{P}^* - \tilde{\mathbb{P}}}\right] .\end{aligned}$$

The fluctuation of this random tensor is

$$\mathbb{E}\left[\phi^{\mathcal{F}^T, \mathbb{P}^D - \tilde{\mathbb{P}}}\right] - \mathbb{E}\left[\phi^{\mathcal{F}^T, \mathbb{P}^* - \tilde{\mathbb{P}}}\right] = \mathbb{E}\left[\phi^{\mathcal{F}^T, \mathbb{P}^D - \mathbb{P}^*}\right]$$

and coincides with $\eta^{\mathcal{F}, \mathbb{P}^*, m}$.

10.3.2 Universal Exponential Family

In case of the universal exponential family, we have $\mathcal{S} = \delta\left[X_{[d]}, L\right]$ and the noise tensor is

$$\eta^{\delta, \mathbb{P}^*, m} = \mathbb{P}^D - \mathbb{P}^* .$$

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

$$\sum_{j \in [m]} \epsilon_{D(j)}\left[X_{[d]}\right] \sim \underline{B}(m, \mathbb{P}^*) .$$

Theorem 10.14 *The noise tensor $\eta^{\delta, \mathbb{P}^*, m}$ is a by $\frac{1}{m}$ rescaled centered multinomial random tensor with parameters $\mathbb{P}^*$ and m, that is*

$$\eta^{\delta, \mathbb{P}^*, m} \sim \frac{1}{m}\left(\underline{B}(m, \mathbb{P}^*) - \mathbb{P}^*\right) .$$

Proof. By the above construction we have

$$\mathbb{P}^D - \mathbb{P}^* = \frac{1}{m} \sum_{j \in [m]} \left(\epsilon_{D(j)}\left[X_{[d]}\right] - \mathbb{E}\left[\epsilon_{D(j)}\left[X_{[d]}\right]\right]\right)$$

We further have

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

■

The noise tensor characterization by multinomial distributions, which holds for universal statistics, is a more detailed characterization compared to the characterization of its marginals by binomial distribution in Thm. 10.13, which holds for generic statistics $\mathcal{F}$.

10.3.3 Guarantees for the 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 10.15 *Whenever the energy tensor of the expected proposal distribution has a gap of Δ, then for every $u > 0$ any mode of the empirical proposal distribution coincides is also a mode of the expected proposal distribution with probability at least $1 - \exp\left[-\frac{1}{u^2}\right]$, provided that*

$$m > C\frac{(1 + \ln[p])}{\Delta^2}$$

where C is a universal constant.

Proof. To prove the theorem we combine the deterministic guarantee Thm. 10.10 with the width bound of Thm. 10.22, 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 $u > 0$ we have

$$\Omega_{\{\epsilon_l[L]\,:\,l\in[p]\}}\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right) < 2\Delta$$

with probability at least $1 - \exp\left[-\frac{1}{u^2}\right]$. The claim thus follows with Thm. 10.10. ■

Example 10.16 (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

$$\Delta\left(\phi^{(\mathcal{F},\mathbb{P}^{(\{f\},\theta)}-\langle\mathbb{I}\rangle[X_{[d]}|\varnothing])}\right) = \frac{1}{2^d - 1 + \exp[-\theta]}$$

If $\theta > 0$ we have an exponentially small gap. Thus, for the above Lemma to apply, the width needs to be exponentially in d small.

Let there be the MLN of a minterm f with d variables, then

$$\Delta(\phi^{(\mathcal{F},\mathbb{P}^{(\{f\},\theta)}-\langle\mathbb{I}\rangle[X_{[d]}|\varnothing])}) = \frac{1}{1 + (2^d - 1)\cdot\exp[-\theta]}$$

For large θ and d, the gap tends to 1. ◊

10.3.4 Guarantees for Unconstrained Parameter Estimation

We here use the sphere bounds and combine with Thm. 10.8 to derive a probabilistic guarantee for unconstrained parameter estimation.

Theorem 10.17 *For any $u \in (0,1)$ we have the following with probability at least $1 - u$. Let θ_D be*

the estimated canonical parameter, then we have for any $t > 0$

$$\left|\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{(\mathcal{F},\theta_D,\nu)}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{(\mathcal{F},\theta_D,\nu)}\right]\right| \leq \tau \cdot \|\theta_D\|_2$$

provided that

$$m \geq \frac{\langle \mu^* \rangle [\varnothing] - \langle (\mu^*)^2 \rangle [\varnothing]}{ut^2} .$$

Proof. The claim follows from the deterministic guarantee Thm. 10.8 with the probabilistic width bound Thm. 10.23 to be shown in the next section. ∎

10.3.5 Face Recovery in Hybrid Logic Networks

Theorem 10.18 *Let $\mathcal{F}$ be a statistic such that $\mathcal{M}_{\mathcal{F},\mathbb{I}}$ is a cube-like polytope (see Def. 8.29). Let $\left((x_0^j, \ldots, x_{d-1}^j) : j \in [m]\right)$ be drawn from a distribution $\mathbb{P}^{\mathcal{F},(A^*,y_{A^*},\theta^*)}$ and let*

$$\Delta^* := \min\{\mu^*[L = l] : l \notin A^*\} \cup \{(1 - \mu^*[L = l]) : l \notin A^*\} .$$

For any $u \in (0,1)$ we have the following with probability at least $1 - u$ that

$$A^{\mu_D} = A^* ,$$

provided that

$$m \geq C \frac{(1 + \ln[p])^2 \cdot (1 - \ln[1-u]^2}{(\Delta^*)^2} ,$$

where C is a universal constant.

Proof. We notice that there is a ℓ_∞ ball of radius Δ^* around μ^* contained in the face, in which interior μ^* lies. It therefore suffices to show, that within the claimed probability the width of the noise tensor with respect to the basis vectors is smaller than Δ^*. To this end, we apply the sub-Gaussian norm bound of Thm. 10.22, which states that

$$\left\|\Omega_{\{\epsilon_l[L] : l \in [p]\}}\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right)\right\|_{\psi_2} \leq C_1 \frac{\sqrt{1 + \ln[p]}}{\sqrt{m}} .$$

Thus, we have for any $t > 0$ (see Lem. 10.20)

$$\mathbb{P}\left[\Omega_{\{\epsilon_l[L] : l \in [p]\}}\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right) \geq C_1 \frac{\sqrt{1 + \ln[p]}}{\sqrt{m}} \cdot t\right] \leq \exp[1 - t] .$$

We choose $t = 1 - \ln[1-u]$ and notice, that the claimed bound holds with the universal constant $C = (C_1)^2$, since in this case

$$C_1 \frac{\sqrt{1 + \ln[p]}}{\sqrt{m}} \cdot (1 - \ln[1-u]) \leq \Delta^* .$$ ∎

10.4 Width Bounds for Hybrid Logic Networks

We here provide width bounds on the noise tensors $\eta^{\mathcal{F},\mathbb{P}^*,m}$ to Hybrid Logic Networks, which coordinates have marginal distributions by Binomials, as shown in Thm. 10.13. 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.

10.4.1 Basis Vectors

We first introduce the sub-Gaussian Norm and show how we can exploit it to state concentration inequalities.

Definition 10.19 (Sub-Gaussian Norm, see Def. 2.5.6 in [Ver18]) The sub-Gaussian norm of a random variable X is defined as

$$\|X\|_{\psi_2} = \inf\left\{C > 0 \,:\, \mathbb{E}\left[\exp\left[\frac{X^2}{C^2}\right]\right] \leq 2\right\}.$$

The moment bound used to define the sub-Gaussian norm can then be combined with the Markov inequality to state concentration bounds.

Lemma 10.20 *For a random variable X with finite sub-Gaussian norm $\|X\|_{\psi_2}$ we have for any $t > 0$*

$$\mathbb{P}\left[|X| > \|X\|_{\psi_2} \cdot t\right] \leq \exp\left[1 - t^2\right].$$

Proof. By the Markov inequality we have for any $t > 0$ and $C > \|X\|_{\psi_2}$

$$\begin{aligned}\mathbb{P}\left[|X| > C \cdot t\right] &= \mathbb{P}\left[\exp\left[\frac{X^2}{C^2}\right] > \exp\left[t^2\right]\right] \\ &\leq \frac{\mathbb{E}\left[\exp\left[\frac{X^2}{C^2}\right]\right]}{\exp\left[t^2\right]} \leq 2 \cdot \exp\left[-t^2\right] = \exp\left[1 - t^2\right].\end{aligned}$$

The claim follows by taking the limit $C \searrow \|X\|_{\psi_2}$. ∎

We now show a sub-Gaussian norm bound on the coordinates of the noise tensor in case of Hybrid Logic Networks.

Lemma 10.21 *The marginal distribution of any coordinate of $\eta^{\mathcal{F},\mathbb{P}^*,m}[L]$ is sub-Gaussian with*

$$\left\|\eta^{\mathcal{F},\mathbb{P}^*,m}[L = l]\right\|_{\psi_2} \leq C_0 \frac{1}{\sqrt{m}}$$

where $C_0 > 0$ is a universal constant .

Proof. Any centered Bernoulli variable is bounded and therefore sub-Gaussian with

$$\left\|\left\langle f_l\left[X_{[d]}\right], \epsilon_{D(j)}\left[X_{[d]}\right]\right\rangle[\varnothing] - \left\langle f_l\left[X_{[d]}\right], \mathbb{P}^*\right\rangle[\varnothing]\right\|_{\psi_2} \leq \frac{1}{\sqrt{\ln[2]}}.$$

Binomial variables are sums of independent Bernoulli variables. We apply the sub-Gaussian norm bound for sums from Proposition 2.6.1 in [Ver18], which states that for a universal constant $C > 0$ we have

$$\left\|\left\langle f_l\left[X_{[d]}\right], \left(\sum_{j\in[m]} \epsilon_{D(j)}\left[X_{[d]}\right] - \mathbb{P}^*\right)\right\rangle[\varnothing]\right\|_{\psi_2} \leq \frac{C \cdot \sqrt{m}}{\sqrt{\ln[2]}}.$$

We therefore have

$$\left\|\eta^{\mathcal{F},\mathbb{P}^*,m}\left[L=l\right]\right\|_{\psi_2} = \frac{1}{m}\left\|\left\langle f_l\left[X_{[d]}\right], \left(\sum_{j\in[m]}\epsilon_{D(j)}\left[X_{[d]}\right]-\mathbb{P}^*\right)\right\rangle[\varnothing]\right\|_{\psi_2} \leq \frac{C}{\sqrt{\ln[2]\cdot m}}.$$

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 10.22 *For the set of basis vectors*

$$\Gamma = \{\epsilon_l[L] : l \in [p]\}$$

we have

$$\left\|\Omega_\Gamma\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right)\right\|_{\psi_2} \leq C_1\sqrt{\frac{1+\ln[p]}{m}},$$

where $C_1 > 0$ is a universal constant.

Proof. We first notice, that

$$\Omega_\Gamma(\eta) = \max_{l\in[p]}\left|\eta^{\mathcal{F},\mathbb{P}^*,m}\left[L=l\right]\right|$$

By a generic bound on the supremum of sub-Gaussian variables (see Exercise 2.5.10 in [Ver18]) we have for a universal constant $C > 0$

$$\left\|\max_{l\in[p]}\left|\eta^{\mathcal{F},\mathbb{P}^*,m}\left[L=l\right]\right|\right\|_{\psi_2} \leq C\left(\max_{l\in[p]}\left\|\eta^{\mathcal{F},\mathbb{P}^*,m}\left[L=l\right]\right\|_{\psi_2}\right)\sqrt{1+\ln[p]}.$$

We now apply Lem. 10.21 and get with $C_1 = C\cdot C_0$ that

$$\left\|\Omega_\Gamma\left(\eta^{\mathcal{F},\mathbb{P}^*,m}\right)\right\|_{\psi_2} \leq C_1\sqrt{\frac{1+\ln[p]}{m}}.$$ ∎

The bound in Thm. 10.22 is furthermore sharp, see the construction of an identically scaling lower bound in Exercise 2.5.11 in [Ver18]. Note that the binomials used here tend to normal distributed variables used in the construction therein.

10.4.2 Sphere

For any tensor $\eta[L]$ and the sphere $\mathbb{S} \subset \mathbb{R}^p$ we have

$$\Omega_{\mathbb{S}}(\eta[L]) = \|\eta[L]\|_2.$$

To show probabilistic width bounds with respect to the sphere, we therefore apply in the following Chebyshevs inequality on the norm of random tensors.

Theorem 10.23 *Let $\mu[L]$ be a deterministic vector with coordinates in $[0,1]$ and $\eta[L]$ a random*

vector, which coordiantes are for $l \in [p]$ marginally distributed as

$$\eta\left[L = l\right] \sim B\left(m, \mu\left[L = l\right]\right) .$$

Then we have for any $u > 0$, $t > 0$ and $m \in \mathbb{N}$ with probability at least $1 - u$

$$\left\| \frac{\eta - \mathbb{E}\left[\eta\right]}{m} \right\|_2 \leq t$$

provided that

$$m \geq \frac{\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right]}{u \cdot t^2} .$$

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 $t > 0$

$$\mathbb{P}\left[\left\| \frac{\eta - \mathbb{E}\left[\eta\right]}{m} \right\|_2 > t\right] = \mathbb{P}\left[\left(\left\| \frac{\eta - \mathbb{E}\left[\eta\right]}{m} \right\|_2\right)^2 > t^2\right] \leq \frac{\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right]}{m \cdot t^2}$$

For a $u > 0$ we choose any m with

$$m \geq \frac{\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right]}{t^2 u}$$

and get

$$\mathbb{P}\left[\left\| \frac{\eta - \mathbb{E}\left[\eta\right]}{m} \right\|_2 > t\right] \leq u .$$

Thus, we have

$$\mathbb{P}\left[\left\| \frac{\eta - \mathbb{E}\left[\eta\right]}{m} \right\|_2 \leq t\right] = 1 - \mathbb{P}\left[\left\| \frac{\eta - \mathbb{E}\left[\eta\right]}{m} \right\|_2 > t\right] \geq 1 - u . \qquad \blacksquare$$

For the universal exponential family where $\mathcal{F} = \delta$ the noise tensor is a rescaled and centered multinomial. In that case, the bound of Thm. 10.23 can be simplified by

$$\left\langle \mu\left[L\right], \left(\mathbb{I}\left[L\right] - \mu\left[L\right]\right)\right\rangle\left[\varnothing\right] = 1 - \left\langle \mu\left[L\right]^2 \right\rangle\left[\varnothing\right] .$$

10.5 Discussion

We in this chapter only provided probabilistic width bounds for Hybrid Logic Networks, which are characterized by boolean statistics. Similar recovery bounds for parameter estimation and

structure learning for more general exponential families would require width bounds in these generic cases. A general approach towards width bounds are chaining techniques on stochastic processes, see [Tal14]. While we showed bounds based on the sub-Gaussian norm, more general sub-exponential bounds could be used, see [Wai19].

We further assumed that our random tensors to be projected are empirical distributions. More general random tensor networks and corresponding width bounds have been developed in [Goe21].

11

First-Order Logic

We in this chapter extend the tensor network based treatment of propositional logic towards first-order logic. In contrast to propositional logic, worlds in first-order logic contain objects, which have relations between each others. We show in this chapter, how these relations are captured by the substitution structure of each world and derive tensor representations of each world. The substitution structure is then combined with the semantic structure, which enumerates possible worlds in a theory. We then generalize Hybrid Logic Networks to the situation of first-order logics and show in particular, that the likelihood of Hybrid Logic Networks on a single first-order logic world is in some cases equivalent to the likelihood of a dataset in propositoinal logics.

11.1 Syntax and Semantics of first-order logic

The framework of first-order logic generalizes propositional logic analogously to the generalization of factored system representations towards structured system representations [RN21]. This more expressive framework is designed to reason about relations and functions between objects. To capture this framework by tensors we here first define the syntax and then investigate tensor representations of the semantics.

11.1.1 Syntax

A first-order logic theory consists in a finite set of constant symbols $\mathcal{U}$, a set of function symbols $\{v_l : l \in [r]\}$, a set of predicate symbols $\{g_k : k \in [d]\}$ and an arity map assigning the arity n_l to the function v_l and the arity n_k to the predicate g_k.

11.1.2 Tensor Representation of Semantics

We here follow the model-theoretic semantics of first-order logic, where sets of possible worlds are considered. We only treat in this work database semantics, where we assume that the domain $\mathcal{U}$ of each world has a one-to-one mapping to the constants of the theory and is therefore constant among the worlds. Database semantics is central to combine the semantic structure with the substitution structure. Under this assumption, the dimension of the substitution structure does not depend on the specific world, and we can combine both structures in a single tensor space to be defined in the next sections.

To each world there is a world domain $\mathcal{U}$ of objects, which we assume to be finite. 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 $I(o_0), \ldots, I(o_{n-1})$.

Functions in first-order logic are object-valued maps

$$v|_{x_W} : \mathcal{U}^r \to \mathcal{U} .$$

While relations are represented by their coordinate encodings, functions are represented by directed basis encodings

$$\beta^{v|_{x_W}} \left[O_v, O_{[n]}\right]$$

where to each object tuple $I(o_0), \ldots, I(o_{n-1})$ the vector $\beta^{v|_{x_W}} \left[O_v, O_{[n]} = o_{[n]}\right]$ is the one-hot encoding of the object, that is

$$\beta^{v|_{x_W}} \left[O_v, O_{[n]} = o_{[n]}\right] = \epsilon_{v|_{x_W}[I(o_0),\ldots,I(o_{n-1})]} [O_v] .$$

11.1.3 Two Tensor Structures

In comparison with propositional logic, first-order logic bears two natural tensor structures.

- **Semantic structure:** As in propositional logic, we enumerate possible worlds by a collection of variables X. This is a representation of the model-theoretic semantics, where subsets of worlds are represented by boolean tensors.
- **Substitution structure:** Different to propositional logic, a formula can have variables and the evaluation of a formula on a world is represented by its grounding tensor. Given a world which contains objects in the domain $\mathcal{U}$, we can substitute variables by objects in the domain. In this way, each predicate with r variables is represented in that world as a boolean tensor of order r.

While the semantic structure appears already in factored representations of systems, the substitution structure arises in more generality when treating structured representations of systems [RN21].

11.2 Substitution Structure

We in this section investigate the structure of terms and formulas in a single first-order logic world, which we for now index by x_W. As we have derived in Chapter 4 for propositional logic, we develop efficient tensor network representations of the representing tensors based on the corresponding syntax.

11.2.1 Grounding Tensors

Given a first-order logic world x_W, 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 logic in Chapter 4.

Definition 11.1 (Grounding of a first-order formula given a world) Given a specific world x_W, with a domain $\mathcal{U}$ enumerated by $[r]$, the grounding of a formula q with variables O_q is the tensor

$$q|_{x_W}\left[O_q\right] : \bigtimes_{l \in [O_q]} [r] \to \{0,1\}\,.$$

Each coordinate represents thereby the boolean, whether the substitution of the variables in the formula is satisfied given a world x_W, that is

$$q|_{x_W}\left[O_q = o_q\right] = 1$$

if and only if the substitution of q with the variables O_q replaced by the objects $I(o_l)$ is satisfied on the world x_W.

11.2.2 Terms

Terms are object valued maps on $\mathcal{U}^n$. The basis encoding of each term corresponds is a boolean and directed tensor. Constants are the functions with $n = 0$, and are represented by basis vectors in $\mathbb{R}^{|\mathcal{U}|}$. Given database semantics, this basis vector does not vary among different worlds. Most general terms are combinations of functions and constants and their basis encodings are acyclic contractions of basis encodings to functions.

11.2.3 Formula Synthesis by Connectives

In order to have a sound semantic, the grounding of first-order logic 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 logic in the β-formalism. While the atomic formulas where delta tensors copying states, they are more involved here.

Theorem 11.2 *For any connective $\circ$ and formulas q_1 and q_2 we have*

$$\begin{aligned}&(q_1 \circ q_2)|_{x_W}\left[O_{q_1} \cup O_{q_2}\right]\\ &\quad = \left\langle \beta^{q_1|_{x_W}}\left[Y_{q_1}, O_{q_1}\right], \beta^{q_2|_{x_W}}\left[Y_{q_2}, O_{q_2}\right], \beta^{\circ}\left[Y_{q_1\circ q_2}, Y_{q_1}, Y_{q_2}\right], \epsilon_1\left[Y_{q_1\circ q_2}\right]\right\rangle \left[O_{[n]}\right]\,.\end{aligned}$$

Proof. This directly follows from Thm. 14.12. ∎

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.

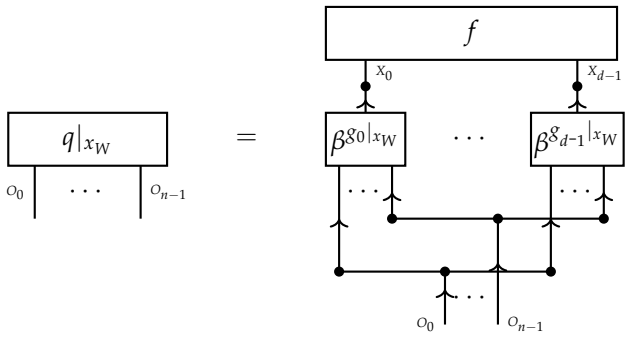
When interpreting the head variables of relational encoded atomic formulas as the atoms of a propositional theory, we find a propositional formula f associated with any decomposable first-order logic formula.

Definition 11.3 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 g_k in first-order logic, when for any world x_W we have

$$q|_{x_W}\left[O_q\right] = \left\langle \{\beta^{g_k|_{x_W}}\left[X_k, O_{g_k}\right] : k \in [d]\} \cup \{f\left[X_{[d]}\right]\}\right\rangle \left[O_q\right]\,.$$

We here denote the head variables of the basis encoding to $\beta^{g_k|_{x_W}}$ by X_k to highlight their interpretation as propositional atoms.

We depict the relation of a grounding tensor to a propositional formula as:



11.2.4 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 11.4 Given a grounding tensor

$$q|_{x_W}\,[O_0,\ldots,O_{n-1}]$$

the existential and universal quantification with respect to the first variable are the tensors

$$(\exists_{o_0} q)|_{x_W}\,[O_1,\ldots,O_{n-1}] \quad \text{and} \quad (\forall_{o_0} q)|_{x_W}\,[O_1,\ldots,O_{n-1}]$$

with coordinates as follows. For an assignment $o_1,\ldots,o$ to the non-quantified variables we have

$$(\exists_{o_0} q)|_{x_W}\,[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1$$

if and only if there is an assignment $o_0 \in [r_0]$ such that

$$q|_{x_W}\,[O_0 = o_0, O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1\,.$$

Conversely, we have for the universal quantification that

$$(\forall_{o_0} q)|_{x_W}\,[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1$$

if and only if for any assignment $o_0 \in [r_0]$ we have

$$q|_{x_W}\,[O_0 = o_0, O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = 1\,.$$

Let us now show, that existential and universal quantification are coordinatewise transforms (see Def. 13.5) of contracted grounding tensors. To this end, let us introduce the greater-z indicator $\mathbb{I}_{>z}$, where $z \in \mathbb{R}$, as the function

$$\mathbb{I}_{>z} : \mathbb{R} \to \{0,1\} \quad , \quad \mathbb{I}_{>z}\,(x) = \begin{cases} 1 & \text{if} \quad x > z \\ 0 & \text{else} \end{cases}\,.$$

Theorem 11.5 *For any formula q with variables $O_{[n]}$ we have*

$$(\exists o_0 q)|_{x_W}[O_1,\ldots,O_{n-1}] = \mathbb{I}_{>0}\left(\langle q|_{x_W}\rangle[O_1,\ldots,O_{n-1}]\right)[O_1,\ldots,O_{n-1}]$$

and

$$(\forall o_0 q)|_{x_W}[O_1,\ldots,O_{n-1}] = \mathbb{I}_{>r-1}\left(\langle q|_{x_W}\rangle[O_1,\ldots,O_{n-1}]\right)[O_1,\ldots,O_{n-1}]$$

Proof. We proof the claimed equalities to arbitrary slices of the remaining variables, which amount to arbitrary substitutions of the formulas. For any indices $o_1 \in [r_1],\ldots,o_{n-1} \in [r_{n-1}]$ We notice that

$$\begin{aligned}\langle q|_{x_W}\rangle[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] &= \sum_{o_0\in[r_0]} q|_{x_W}[O_0 = o_0,\ldots,O_{n-1} = o_{n-1}] \\ &= |o_0 \in [r_0] : q|_{x_W}[O_0 = o_0,\ldots,O_{n-1} = o_{n-1}] = 1| \,.\end{aligned}$$

We can thus understand the contracted grounding tensor as storing in its coordinates the count of the coordinate extensions to the zeroth variable, such that the grounding tensor is satisfied. This is analogous to our interpretation of contracted propositional formulas as world counts. From this it is obvious, that the existential quantification is satisfied, if the count is different from zero, which is captured by the coordinatewise transform with $\mathbb{I}_{>0}$. We therefore arrive at

$$(\exists_{o_0} q)|_{x_W}[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = \mathbb{I}_{>0}\left(\langle q|_{x_W}\rangle[O_1,\ldots,O_{n-1}]\right)[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] \,.$$

The first claim follows, since the assignment to the non-quantified variables was arbitrary. The universal quantification is satisfied, when all extensions are satisfied, and the count is r. Since r is the maximal count, this is captured by the coordinatewise transform with $\mathbb{I}_{>r-1}$ and we get

$$(\forall o_0 q)|_{x_W}[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] = \mathbb{I}_{>r-1}\left(\langle q|_{x_W}\rangle[O_1,\ldots,O_{n-1}]\right)[O_1 = o_1,\ldots,O_{n-1} = o_{n-1}] \,.$$

With the same argument, the second claim is established. ∎

We can extend this discussion towards more generic counting quantifiers, of which the existential and the universal quantifier are extreme cases. One can define quantifiers by demanding that at least $z \in \mathbb{N}$ compatible groundings are satisfied, and show that they amount to coordinatewise transforms with $\mathbb{I}_{>z}$. What is more, quantifiers demanding that at most $z \in \mathbb{N}$ are satisfied would be representable by transforms with an analogously defined function $\mathbb{I}_{\leq z}$. Such customized quantifiers appear for example in the OWL 2 standard of description logics (see [Rud11] and Sect. 11.4).

As will be discussed in Chapter 14, any coordinatewise transform can be performed by a contraction of a basis encoding of the tensor with a head vector prepared by the transform function (see Thm. 14.13). 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 quantifier free part. We can represent its grounding tensor in three steps:

- Instantiations of the atom groundings with the assigned variables, as contractions of the basis 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.

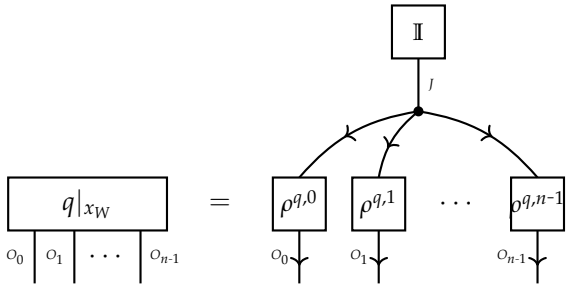


Figure 11.1: Basis CP Decomposition of the grounding of q, following the scheme of Thm. 15.4. Instead of direct storage of the grounding tensor $q|_{x_W}$, the non-zero coordinates are enumerated by a variable J and the corresponding coordinates stored in leg-matrices $\rho^{q,l}$.

11.2.5 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 Thm. 15.4. 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 Thm. 11.1.

Most logical syntaxes exploit ℓ_0-sparsity, explicitly storing only known assertions. The interpretation of unspecified assertions depends on the underlying assumptions. Under the Closed World Assumption, for example, all unspecified assertions are assumed to be false.

11.2.6 Summary

Let us summarize the tensor encodings of the representations of the different concepts, given a single first-order logic world:

Concept	**Representation**	**Decomposition**
Constant Symbol	Basis vector	
Function Symbol	Basis encoding: Boolean and directed tensor	
Term	""	Acyclic tensor network of represented functions
Predicate Symbol	Coordinate encoding: Boolean tensor	
Quantifier-free Formula	""	Contraction of represented terms with relations
Formula	""	Transform of corresponding quantifier-free formula

11.2.7 Example: Relational Databases

A database is understood as a specific first-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 $\rho^{q,l}$ (see Figure 11.1).

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 11.6 (Outer join) Let there be a world x_W and formulas g_l depending on variables $O_{\mathcal{V}^l}$, which have grounding tensors by

$$g_l|_{x_W}[O_v] : \bigtimes_{v\in\mathcal{V}^l}[r_v] \to \{0,1\}\,.$$

Then their (outer) JOIN is defined as the grounding of their conjunctions, as

$$\text{JOIN}\left(g_0,\ldots,g_{p-1}\right)|_{x_W}\left[\bigcup_{l\in[p]} O_{\mathcal{V}^l}\right] = \langle g_l|_{x_W}[O_{\mathcal{V}^l}] : l\in[p]\rangle\left[\bigcup_{l\in[p]} O_{\mathcal{V}^l}\right].$$

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. 13.5) with the product as transform function. To make this connection solid, one would need to extend each joined formula trivially to the variables appearing in other formulas.

The efficiency of evaluating the contraction to a JOIN operation might be improved by understanding it as an Constraint Satisfaction Problem (see Chapter 5). 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 $g_l|_{x_W}$ would be used to initialize Knowledge Cores κ^e and sequentially sparsified during the algorithm.

11.3 Semantic Structure

We now investigate the semantic structure of a first-order logic theory, when restricting to database semantics. This involves the representation of collections of worlds, where each has a substitution structure as discussed above.

11.3.1 World Enumerating Variables

Given database semantics, we have a finite set of worlds, which we enumerate by tuples of variables X.

Definition 11.7 (World enumerating variables) Given a first-order logic theory with constant symbols $\mathcal{U}$, function symbols $\{v_l : l\in[r]\}$ and predicate symbols $\{g_k : k\in[d]\}$, we

enumerate the world under database semantics by a tuple

$$X_W = \left(\bigtimes_{\tilde{k} \in [d]} \bigtimes_{o_{\tilde{k}} \in [r]^{n_{\tilde{k}}+1}} X_{\tilde{k},o_{\tilde{k}}} \right) \times \left(\bigtimes_{k \in [d]} \bigtimes_{o_k \in \in [r]^{n_k}} X_{k,o_k} \right)$$

of boolean variables. In the world indexed by x_W, the grounding tensor of the function v_l is

$$\beta^{v_l|_{x_W}} \left[O_l, O_{[r_l]} \right] = \sum_{o_{\tilde{k}} \in [r]^{n_{\tilde{k}}+1} : o_{\tilde{k}}=1} \epsilon_{o_{\tilde{k}}} \left[O_{\tilde{k}} \right]$$

and of the predicate g_k

$$g_k|_{x_W} \left[O_{[r_k]} \right] = \sum_{o_{\tilde{k}} \in [r]^{n_k} : o_k=1} \epsilon_{o_k} \left[O_k \right] .$$

We further have the restriction that each basis encoded function is a directed tensor. To restrict to those worlds, where this is true, we further introduce the validation base measure $\nu \left[X_W \right]$ with coordinates

$$\nu \left[X_W = x_W \right] = \begin{cases} 1 & \text{if} \quad \forall l \in [r] : \sum_{o_l \in [r]^{n_{\tilde{k}}+1}} = 1 \\ 0 & \text{else} \end{cases} .$$

11.3.2 Representation of Terms and Formulas

Combining its substitution structure and semantic structure we represent a first-order logic term v with arity r_v as a tensor

$$\beta^v \left[O_v, O_{[r_v]}, X_W \right] = \sum_{x_W : \nu[X_W = x_W]=1} \beta^{v|_{x_W}} \left[O_v, O_{[r_v]} \right] \otimes \epsilon_{x_W} \left[X_W \right] .$$

and a formula as the tensor

$$q \left[O_q, X_W \right] = \sum_{x_W : \nu[X_W = x_W]=1} q|_{x_W} \left[O_q \right] \otimes \epsilon_{x_W} \left[X_W \right] .$$

From Def. 11.7 the representation of predicate and function symbols are directly derived from the X_W, that is

$$g_k \left[O_{[r_k]}, X_W = x_W \right] = \sum_{o_{\tilde{k}} \in [r]^{n_{\tilde{k}}+1} : o_{\tilde{k}}=1} \epsilon_{o_{\tilde{k}}} \left[O_{\tilde{k}} \right]$$

and

$$\beta^{v_l} \left[O_l, O_{[r_l]}, X_W = x_W \right] = \sum_{o_{\tilde{k}} \in [r]^{n_{\tilde{k}}+1} : o_{\tilde{k}}=1} \epsilon_{o_{\tilde{k}}} \left[O_{\tilde{k}} \right]$$

11.3.3 Case of Propositional Logic

Propositional logic as discussed in (see Chapter 4) is the special case of first-order logic with empty sets of constant and function symbols, and $n_k = 0$ for the predicate symbols. With the

convention $[]^0 = [2]$ the worlds are enumerated by the tuple

$$X_W = \bigtimes_{k \in [d]} X_k$$

which we have in previous chapters abbreviated by $x_{[d]}$.

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 logic, this is

$$\epsilon_{x_W}\left[X_{[d]}\right] = \bigotimes_{k \in [d]} \epsilon_{x_k}\left[X_k\right] .$$

11.3.4 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

$$\epsilon : \left(\bigtimes_{\tilde{k} \in [d]} \bigtimes_{o_{\tilde{k}} \in [r]^{n_{\tilde{k}}+1}} [2]\right) \times \left(\bigtimes_{k \in [d]} \bigtimes_{o_k \in [r]^{n_k}} [2]\right) \to \left(\bigotimes_{\tilde{k} \in [d]} \bigotimes_{o_{\tilde{k}} \in [r]^{n_{\tilde{k}}+1}} \mathbb{R}^2\right) \otimes \left(\bigotimes_{k \in [d]} \bigotimes_{o_k \in [r]^{n_k}} \mathbb{R}^2\right)$$

defined by

$$\epsilon_{x_W}\left[X_W\right] = \left(\bigotimes_{\tilde{k} \in [d]} \bigotimes_{o_{\tilde{k}} \in [r]^{n_{\tilde{k}}+1}} \epsilon_{x_{\tilde{k},o_{\tilde{k}}}}\left[X_{\tilde{k},o_{\tilde{k}}}\right]\right) \otimes \left(\bigotimes_{k \in [d]} \bigotimes_{o_k \in [r]^{n_k}} \epsilon_{x_{k,o_k}}\left[X_{k,o_k}\right]\right)$$

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 $d \cdot r^n$.

11.3.5 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 normalized tensors $\mathbb{P}\left[X_W\right]$ 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

$$\left(\sum_{k \in [d]} r^{n_k}\right) + \left(\sum_{\tilde{k} \in [d]} r^{n_{\tilde{k}}}\right)$$

Booleans. 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.

11.4 Representation of Knowledge Graphs

Let us now represent a specific fragment of first-order logic, namely Description Logics which Knowledge Bases are often referred to as Knowledge Graphs. We here use the OWL 2 standard, which encodes the syntax of the description logic $\mathcal{SROIQ}(\mathcal{D})$ [Rud11].

11.4.1 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 $\epsilon_0[O_1]$. A Knowledge Graph on the set $\mathcal{U}$ of constants (owl:NamedIndividuals) is then the tensor

$$\mathrm{KG}|_{x_W}[L, O_0, O_1] : [d] \times [r] \times [r] \to \{0, 1\}.$$

11.4.2 Representation as Ternary Predicate

It has been particularly convenient to represent a Knowledge Graph instead as a grounding of a single ternary predicate RDF. To this end, the predicates g_k and another object rdftype are added to a domain $\mathcal{U}$, by extending the r and the index interpretation function accordingly.

Following our notation we understand a Knowledge Graph as a grounding of the rdf triple relation RDF (being a formula of order 3) on a specific world $\mathrm{KG}|_{x_W}$ with individuals $\mathcal{U}$

We then construct a grounding tensor $\mathrm{RDF}|_{x_W}$ out of the world $\mathrm{KG}|_{x_W}[L, O_0, O_1]$ by

$$\mathrm{RDF}|_{x_W} : [r] \times [r] \times [r] \to \{0, 1\}$$

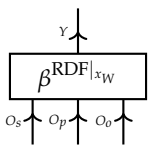
where

$$\mathrm{RDF}|_{x_W}\left[O_s = o_s, O_p = o_p, O_o = o_o\right] = \begin{cases} \mathrm{KG}|_{x_W}[L = o_s, O_0 = o_o, O_1 = 0] & \text{if} \quad o_p = I^{-1}(\text{rdftype}) \\ \mathrm{KG}|_{x_W}\left[L = o_p, O_0 = o_s, O_1 = o_o\right] & \text{if} \quad o_p = I^{-1}(g_k) \quad \text{for some} \quad k \\ 0 & \text{else} \end{cases}.$$

Slicing the tensor $\mathrm{RDF}|_{x_W}$ along the predicate axis retrieves specific information about roles and can be efficiently be performed on these formats. The role rdftype has a specific meaning, since it contains from a DL perspective classifications (memberships of named concepts). Further slicing the tensor along object axis therefore results in membership lists for specific classes (concepts). One can thus regard rdftype as a placeholder for unitary formulas in a space of binary formulas.

Exploiting the ℓ_0-sparsity now leads to a so-called triple store, where $\mathrm{RDF}|_{x_W}$ is stored by a listing of those triples o_s, o_p, o_o such that $\mathrm{RDF}|_{x_W}\left[O_s = o_s, O_p = o_p, O_o = o_o\right] = 1$ A recent implementation of a triple store exploiting these intuitions is TENTRIS, see [Big+20]. In this work, such decompositions are generalized into more generic CP formats, see Chapter 15. Approximations of grounding tensors by decompositions leads to embeddings of the individuals such as Tucker, ComplEx and RESCAL (see [Nic+16]).

For our purposes of evaluating logical formulas such as SPARQL queries we use the basis encoding of the groundings, which are depicted by



11.4.3 SPARQL Queries

The SPARQL query language is a syntax to express first-order logic formulas q and intended to be evaluated given a Knowledge Graph. We here consider tensor network representations of the WHERE· block. Given a specific knowledge graph 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.

11.4.3.1 Triple Patterns

Central to SPARQL queries are triple patterns, which we understand as slicings of the tensor RDF$|_{x_W}$. To each so-called triple pattern we build a corresponding creation tensor. The triple pattern is then evaluated by contraction of the atom creating tensor with 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 RDF$|_{x_W}$. The corresponding pattern tensor is then

$$\psi_{\langle Z,\text{rdftype},g_k\rangle}\left[O_s,O_p,O_o,Z\right] = \delta\left[O_s,Z\right]\otimes\epsilon_{I^{-1}(\text{rdftype})}\left[O_p\right]\otimes\epsilon_{I^{-1}(g_k)}\left[O_o\right].$$

Binary triple patterns come with two projection variables, typically related with the subject and the object variables O_s and O_o. The pattern tensor to the k-th predicate is then

$$\psi_{\langle Z_0,g_k,Z_1\rangle}\left[O_s,O_p,O_o,Z_0,Z_1\right] = \delta\left[O_s,Z_0\right]\otimes\epsilon_{I^{-1}(g_k)}\left[O_p\right]\otimes\delta\left[O_o,Z_1\right].$$

Contraction with these pattern tensor evaluated the specific triple pattern, and outputs in a boolean tensor the indicator, which objects are members of a specific class (for unary patterns) or which pair of objects are related by a specific relation. Again, the output of such contractions is a subset encodings of the set of solutions (see Def. 14.1).

Examples of triple patterns, drawn in Figure 11.2 are

- Unary triple pattern with one variable, representing a formula with a single projection variable. For the example $\langle Z,\text{rdftype},C\rangle$ see Figure 11.2a.

  $$\psi_{\langle Z,\text{rdftype},g_k\rangle}\left[O_s,O_p,O_o,Z\right] = \delta\left[O_s,Z\right]\otimes\epsilon_{I^{-1}(\text{rdftype})}\left[O_p\right]\otimes\epsilon_{I^{-1}(C)}\left[O_o\right]$$

  If and only if the output slice is ϵ_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 11.2b. If and only if the output slice is ϵ_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.

11.4.3.2 Basic Graph Patterns

Generic SPARQL queries are compositions of triple patterns by logical connectives. These triple patterns possibly share projection variables. Statements in SPARQL can be translated into propositional logic combining the triple patterns:

SPARQL	Propositional logic	Tensor Representation
$\{f_1,f_2\}$	$f_1\wedge f_2$	$\beta^{\wedge}\left[Y_{f_1\wedge f_2},Y_{f_1},Y_{f_2}\right]$
UNION$\{f_1,f_2\}$	$f_1\vee f_2$	$\beta^{\vee}\left[Y_{f_1\vee f_2},Y_{f_1},Y_{f_2}\right]$
FILTER NOT EXISTS$\{f\}$	$\neg f$	$\beta^{\neg}\left[Y_{\neg f},Y_f\right]$

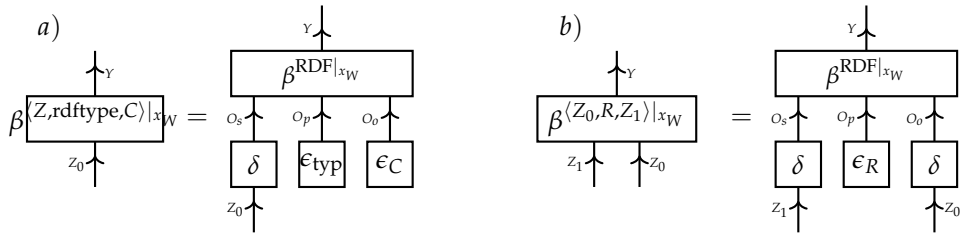


Figure 11.2: Triple patterns of SPARQL as tensor networks. a) Example of unary triple pattern $\langle Z, \text{rdftype}, C\rangle$ specifying whether an individual I_{o_1} is a member of class C. b) Example of a binary triple pattern $\langle Z_0, R, Z_1\rangle$ specifying whether individuals I_{o_1} and I_{o_2} have a relation R. By $\epsilon_{\text{typ}}, \epsilon_C, \epsilon_R$ we denote the one-hot encodings of the enumeration of the resources $rdf : type, C$ and R.

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. 11.2.3. To this end, we prepare for each appearing triple pattern the corresponding pattern tensor, and a copy of $\text{RDF}|_{x_W}$. Here we also copy the term variables O_s, O_p and O_o, to ensure that each copy of $\text{RDF}|_{x_W}$ 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 basis encoding of logical connectives according to the hierarchy specified in the SPARQL query. Finally we add a ϵ_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 hybrid 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 ϵ_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 $\neg$.

11.5 Probabilistic Relational Models

So far we have studied Markov Logic Networks in propositional logic as probability distributions over worlds. In FOL they define probability distributions over relations in worlds with a fixed set of objects. More generally, such models are probabilistic relational models (see for an overview [GT19].

We in this section treat random worlds in first-order logic 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 logic.

11.5.1 Hybrid First-Order Logic Networks

Following [RD06] Markov Logic Networks in first-order logic 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.

We count the number of true groundings to a formula by

$$\#q_l|_{x_W} = \left\langle q_l|_{x_W} \left[O_{q_l}\right] \right\rangle [\varnothing] \ .$$

Using these counts as statistics, we now define Hybrid First-Order Logic Networks as a Computation Activation Network.

Definition 11.8 (Hybrid First-Order Logic Networks (HFLN)) Let there be a set $\mathcal{U}$ of objects and a set $\mathcal{Q}$ of first-order logic formulas. Further let (A, y_A, θ) as in Def. 8.10 be a tuple of a subset $A \subset [p]$, a tuple $y_A \in \bigtimes_{l \in A}[2]$ and $\theta\,[L] \in \mathbb{R}^p$. We then define the Hybrid First-Order Logic Network as the distribution

$$\begin{aligned}\mathbb{P}^{\#\mathcal{Q},(A,y_A,\theta)}\,[X_W] = \big\langle &\{\beta^{\#q_l}\,[Y_l, X_W] \,:\, l \in [p]\} \\ &\cup \{\alpha^{l,\theta}\,[Y_l] \,:\, l \in [p]\} \cup \{\kappa^{l,(A,y_A)}\,[Y_l] \,:\, l \in [p]\}\big\rangle\,[X_W|\varnothing]\end{aligned}$$

where

$$\kappa^{l,(A,y_A)}\,[Y_l] = \begin{cases} \epsilon_0\,[Y_l] & \text{if} \quad l \in A,\, y_l = 0 \\ \epsilon_{n_l-1}\,[Y_l] & \text{if} \quad l \in A,\, y_l = 1 \\ \mathbb{I}\,[Y_l] & \text{else}\,. \end{cases}$$

The mean parameter polytope is the convex hull of the vectors

$$\sigma^{\mathcal{Q}}\,[X_W = x_W, L]$$

to the worlds x_W with $\nu\,[X_W = x_W] = 1$. These vectors store are the counts of satisfied groundings to each formula, that is

$$\sigma^{\mathcal{Q}}\,[X_W = x_W, L] = \left|\{o_{q_l} \,:\, q_l|_{x_W}\left[O_{q_l} = o_{q_l}\right] = 1\}\right| \ .$$

11.5.2 Propositionalization

Let us notice, that different to the case of propositional Hybrid Logic Networks treated in Chapter 8, 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 Hybrid First-Order Logic Network when posing certain weight sharing conditions on θ. The downside of this construction is the increase in the number of features from p to $\sum_{l\in[p]} |\mathcal{U}|^{|O_{q_l}|}$. This polynomial in the cardinality of the domain set increase poses significant computational challenges, see [RD06].

We now show that the Hybrid First-Order Logic Network can be propositionalized to a Hybrid Logic Network in propositional logic.

Lemma 11.9 *We have*

$$\left\langle \beta^{\#q_l}\left[Y_l, X_W\right], \alpha^{l,\theta}\left[Y_l\right]\right\rangle\left[X_W\right] = \left\langle \{\beta^{q_l|_{x_W}\left[x_W, o_{q_l}\right]}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\} \cup \{\alpha^{l,\theta}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\}\right\rangle\left[X_W\right]$$

and

$$\left\langle \beta^{\#q_l}\left[Y_l, X_W\right], \kappa^{l,(A,y_A)}\left[Y_l\right]\right\rangle\left[X_W\right] = \left\langle \{\beta^{q_l|_{x_W}\left[x_W, o_{q_l}\right]}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\} \cup \{\kappa^{l,(A,y_A)}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\}\right\rangle\left[X_W\right] .$$

Proof. The first claim holds, since for any world x_W we have

$$\begin{aligned}
&\left\langle \beta^{\#q_l}\left[Y_l, X_W\right], \alpha^{l,\theta}\left[Y_l\right]\right\rangle\left[X_W = x_W\right] \\
&= \prod_{o_{q_l} : q_l|_{x_W}\left[x_W, o_{q_l}\right]=1} \exp\left[\theta\left[L = l\right]\right] \\
&= \left\langle \{\beta^{q_l|_{x_W}\left[x_W, o_{q_l}\right]}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\} \cup \{\alpha^{l,\theta}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\}\right\rangle\left[X_W = x_W\right] .
\end{aligned}$$

For $l \notin A$ the second claim is trivial, since any contraction of a basis encoding with a trivial head is trivial. If $l \in A$ and $y_l = 0$ then

$$\begin{aligned}
&\left\langle \beta^{\#q_l}\left[Y_l, X_W\right], \kappa^{l,(A,y_A)}\left[Y_l\right]\right\rangle\left[X_W = x_W\right] \\
&= \begin{cases} 1 & \text{if} \quad \forall o_{q_l} : q_l|_{x_W}\left[x_W, o_{q_l}\right] = 0 \\ 0 & \text{else} \end{cases} \\
&= \left\langle \{\beta^{q_l|_{x_W}\left[x_W, o_{q_l}\right]}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\} \cup \{\kappa^{l,(A,y_A)}\left[Y_{o_{q_l}}\right] : o_{q_l} \in [r_{q_l}]\}\right\rangle\left[X_W = x_W\right] .
\end{aligned}$$

For $y_l = 1$ this holds analogously. ∎

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.

11.5.3 Conditioning on an Importance Formula

To reduce the number of object tuples influencing the probability distribution, we now condition Hybrid First-Order Logic Networks on situations where a formula, called the importance formula, has a fixed grounding tensor.

To this end, we mark pairs of term indices relevant to the distributions by an auxiliary index $j \in [m]$. Given a set $\{o^j_{[n]} : j \in [m]\}$ of indices to the important tuples we build a set encoding (see Def. 14.1)

$$\underline{p} = \sum_{j \in [m]} \left(\bigotimes_{l \in [n]} \epsilon_{o^j_l}\left[O_l\right] \right) .$$

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 Hybrid First-Order Logic Network 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 Hybrid First-Order Logic Network 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.

11.5.4 Decomposition of the Log Likelihood

To reduce the likelihood of a world to we make the assumption that all formulas in a Hybrid First-Order Logic Network are of the form

$$q_l\left[O_{q_l}\right] = \left(p\left[O_{[n]}\right] \Rightarrow h_l\left[O_{q_l}\right]\right) \tag{11.1}$$

that is a rule with the importance formula being the premise. In particular, we assume, that they depend on all term variables $O_{[n]}$. If this is not the case, we extend the formula trivially on the missing term variables. When this assumption holds, we can think of the importance formula as a conditions on individuals to satisfy a statistical relation given by h.

Towards connecting with propositional logic, we further make the assumption, that we can decompose the formula h_l in what we will call extraction formulas.

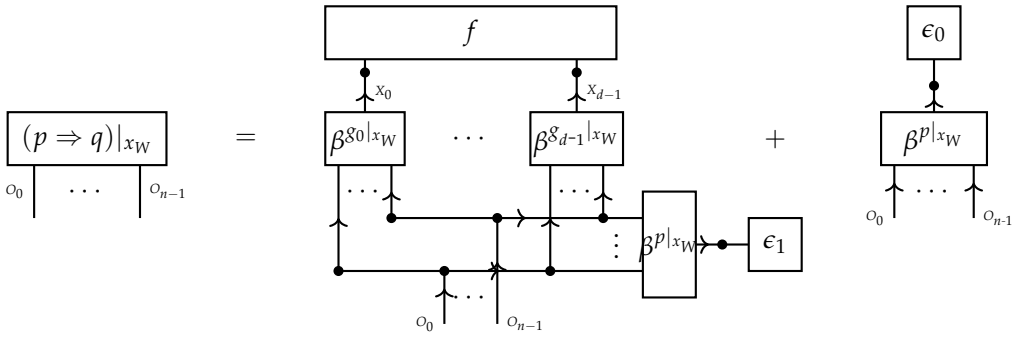
Assumption 11.10 We assume that there exist formulas $\{g_k\left[X_W, O_{[n]}\right] : k \in [d]\}$, which we refer to as atom extraction formulas, and an importance formula $p\left[O_{[n]}\right]$ such that the following holds. To each first-order logic formula q_l there is another first-order logic formula $h_l\left[O_{q_l}\right]$ and a propositional formula $f_l\left[X_{[d]}\right]$ such that

$$q_l\left[X_W, O_{[n]}\right] = \left(p\left[X_W, O_{[n]}\right] \Rightarrow h_l\left[X_W, O_{[n]}\right]\right)$$

and

$$h_l\left[O_{[n]}\right] = \left\langle \{f_l\left[X_{[d]}\right]\} \cup \{\beta^{g_k}\left[X_k, X_W, O_{[n]}\right] : k \in [d]\}\right\rangle\left[O_{[n]}\right].$$

We depict the assumption, that any formula is of the form (11.1) in the diagram



where the second summand depends only on the query p and therefore does not appear in the likelihood.

11.5.5 Reduction to Propositional Logic

We now make additional assumptions to decompose the partition function of an Hybrid First-Order Logic Network as a product of Hybrid Logic Network partition functions.

Assumption 11.11 Given an importance formula p and a tensor $\underline{p}$ we consider the base measure

$$\nu^{\underline{p}}[X_W] = \begin{cases} 1 & \text{if} \quad p|_{x_W}[O_p] = \underline{p}[O_p] \\ 0 & \text{else} \end{cases}$$

of worlds with grounding of p by $\underline{p}$. Let further enumerate by $j \in [m]$ the support of $\underline{p}$, that is $o^j_{[n]}$ are the indices of $O_{[n]}$ with $\underline{p}\left[O_{[n]} = o^j_{[n]}\right] = 1$. We assume that with respect to $\underline{p}$ the variables

$$\left(g_k|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]\right)$$

are for $k \in [d]$ and $j \in [m]$ independent and uniformly distributed.

When the above assumption holds, we now show that the probability of a first-order logic world with respect to a Hybrid First-Order Logic Network coincides with the likelihood of a dataset in a propositional Hybrid Logic Network.

Theorem 11.12 *Let there be a set of formulas $\mathcal{Q}$ such that Assumption 11.10 and Assumption 11.11 hold with an importance formula p and a tensor $\underline{p}$. Then for any (A, y_A, θ) and any world x_W with $p|_{x_W} = \underline{p}$ we have*

$$\frac{1}{m}\ln\left[\mathbb{P}^{\#\mathcal{Q},(A,y_A,\theta)}\left[X_W = x_W | p|_{x_W} = \underline{p}\right]\right] = \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F},(A,y_A,\theta)}\right] - \frac{\ln\left[\left\langle \nu^{\underline{p}}[X_W]\right\rangle[\varnothing]\right]}{m}$$

where $\mathcal{F}$ is the set of propositional equivalents to $\mathcal{Q}$ (see Assumption 11.10) and D the data map with evaluation at $j \in [m]$ by the enumerated non-vanishing coordinates of $\underline{p}[O_p]$

$$D(j) = \left(g_0|_{x_W}\left[O_{[n]} = o^j_{[n]}\right], \ldots, g_{d-1}|_{x_W}\left[O_{[n]} = o^j_{[n]}\right]\right).$$

To show the theorem, we show first in the following lemma the factorization of the partition function of the Hybrid First-Order Logic Network.

Lemma 11.13 *Under the assumptions of Thm. 11.12, we have*

$$\begin{aligned} &\left\langle \beta^{\mathcal{Q}|_{\mathcal{U}}}\left[Y_{[p]}, X_W\right], \tau^{(A,y_A,\theta)}\left[Y_{[p]}\right], \nu^{\underline{p}}[X_W]\right\rangle[\varnothing] \\ &\quad = \left\langle \nu^{\underline{p}}[X_W]\right\rangle[\varnothing] \cdot \left(\prod_{o_{[n]} : \underline{p}\left[O_{[n]} = o_{[n]}\right] = 0} \tau^{l,(A,y_A,\theta)}[Y = 1]\right) \cdot \\ &\qquad \left(\frac{1}{2^d} \cdot \left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \tau^{(A,y_A,\theta)}\left[Y_{[p]}\right]\right\rangle[\varnothing]\right)^m. \end{aligned}$$

Proof. Using the assumption on the structure of the formulas we have

$$\#q_l|_{x_W} = \#\neg p|_{x_W} + \#(p \wedge h_l)|_{x_W}$$

and for any x_W

$$\left\langle \beta^{\#q_l}\left[Y_l\right], \tau^{l,(A,y_A,\theta)}\left[Y\right], \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[X_W = x_W\right]$$
$$= \left(\prod_{o_{[n]}\,:\,\underline{p}\left[O_{[n]}=o_{[n]}\right]=0} \tau^{l,(A,y_A,\theta)}\left[Y=1\right]\right)\cdot\left\langle \beta^{\#(p\wedge h_l)}\left[Y_l\right], \tau^{l,(A,y_A,\theta)}\left[Y_l\right], \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[X_W = x_W\right]$$

Here by $\tau^{l,(A,y_A,\theta)}\left[Y\right]$ we denote the two-dimensional realization of the activation core. We use that the grounding tensor of p is constant among the by $\nu^{\underline{p}}\left[X_W\right]$ supported worlds and thus

$$\left\langle \beta^{\#(p\wedge h_l)}\left[Y_l\right], \tau^{l,(A,y_A,\theta)}\left[Y_l\right], \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[X_W\right]$$
$$= \left\langle \bigcup_{j\in[m]}\left\{\beta^{h_l|_{o^j_{[n]}}}\left[Y_{l,j}\right], \tau^{l,(A,y_A,\theta)}\left[Y_{l,j}\right]\right\}\cup\left\{\nu^{\underline{p}}\left[X_W\right]\right\}\right\rangle\left[X_W\right].$$

For each $j\in[m]$ we have by Assumption 11.10

$$\left\langle \beta^{h_l|_{o^j_{[n]}}}\left[Y_{l,j}\right], \tau^{l,(A,y_A,\theta)}\left[Y_{l,j}\right]\right\rangle\left[X_W\right]$$
$$= \left\langle \left\{\beta^{g_k|_{o^j_{[n]}}}\left[Y_{k,j}\right] \,:\, k\in[d]\right\}\cup\left\{\beta^{f_l}\left[Y_f, Y_{k,j}\right], \tau^{l,(A,y_A,\theta)}\left[Y_f\right]\right\}\right\rangle\left[X_W\right].$$

From Assumption 11.11 we know

$$\left\langle \left\{\beta^{g_k|_{o^j_{[n]}}}\left[Y_{k,j}\right] \,:\, k\in[d],\, j\in[m]\right\}\cup\left\{\nu^{\underline{p}}\left[X_W\right]\right\}\right\rangle\left[Y_{[d]\times[m]}\right] = \frac{\left\langle \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[\varnothing\right]}{2^{d\cdot m}}\cdot\mathbb{I}\left[Y_{[d]\times[m]}\right].$$

Combining the above we get

$$\left\langle \beta^{\mathcal{Q}|_{\mathcal{U}}}\left[Y_{[p]}, X_W\right], \tau^{(A,y_A,\theta)}\left[Y_{[p]}\right], \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[\varnothing\right]$$
$$= \frac{\left\langle \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[\varnothing\right]}{2^{d\cdot m}}\cdot$$
$$\left(\prod_{o_{[n]}\,:\,\underline{p}\left[O_{[n]}=o_{[n]}\right]=0} \tau^{l,(A,y_A,\theta)}\left[Y=1\right]\right)\prod_{j\in[m]}\left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, Y_{[d]\times\{j\}}\right], \tau^{(A,y_A,\theta)}\left[Y_{[p]}\right]\right\rangle\left[\varnothing\right]$$
$$= \left\langle \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[\varnothing\right]\cdot\left(\prod_{o_{[n]}\,:\,\underline{p}\left[O_{[n]}=o_{[n]}\right]=0} \tau^{l,(A,y_A,\theta)}\left[Y=1\right]\right)$$
$$\left(\frac{1}{2^d}\cdot\left\langle \beta^{\mathcal{F}}\left[Y_{[p]}, X_{[d]}\right], \tau^{(A,y_A,\theta)}\left[Y_{[p]}\right]\right\rangle\left[\varnothing\right]\right)^m.$$

■

We notice that for the event $p|_{x_W} = \underline{p}$ to be of non-vanishing probability, we need to have $y_A = \mathbb{I}_A$ or $\underline{p}\left[O_p\right] = \mathbb{I}\left[O_p\right]$. With this lemma, we are now show Thm. 11.12.

Proof of Thm. 11.12. We have

$$\ln\left[\mathbb{P}^{\#\mathcal{Q},(A,y_A,\theta)}\left[X_W = x_W \middle| p|_{x_W} = \underline{p}\right]\right] = \ln\left[\left\langle \beta^{\mathcal{Q}|_{\mathcal{U}}}\left[Y_{[p]}, X_W\right], \tau^{(A,y_A,\theta)}\left[Y_{[p]}\right], \nu^{\underline{p}}\left[X_W\right]\right\rangle\left[X_W = x_W\right]\right]$$

$$- \ln \left[\left\langle \beta^{\mathcal{Q}|\mathcal{U}} \left[Y_{[p]}, X_W \right], \tau^{(A,y_A,\theta)} \left[Y_{[p]} \right], \nu^{\underline{p}} \left[X_W \right] \right\rangle [\varnothing] \right]$$

While the second term is decomposed by Lem. 11.13 we now derive a decomposition of the first term. By Assumption 11.10 and $p|_{x_W} = \underline{p}$ we have

$$\begin{aligned}
&\left\langle \beta^{\mathcal{Q}|\mathcal{U}} \left[Y_{[p]}, X_W \right], \tau^{(A,y_A,\theta)} \left[Y_{[p]} \right], \nu^{\underline{p}} \left[X_W \right] \right\rangle [X_W = x_W] \\
&= \left(\prod_{o_{[n]} : \underline{p}[O_{[n]} = o_{[n]}] = 0} \tau^{l,(A,y_A,\theta)} [Y = 1] \right) \cdot \\
&\quad \prod_{j \in [m]} \left\langle \{ \beta^{g_k|_{o^j_{[n]}}} [Y_k, X_W = x_W] : k \in [d] \} \cup \{ \beta^{\mathcal{F}} \left[Y_{[p]}, X_{[d]} \right], \tau^{(A,y_A,\theta)} Y_{[p]} \} \right\rangle [\varnothing] \, .
\end{aligned}$$

With Lem. 11.13 we then have

$$\begin{aligned}
&\frac{1}{m} \ln \left[\mathbb{P}^{\#\mathcal{Q},(A,y_A,\theta)} \left[X_W = x_W | p|_{x_W} = \underline{p} \right] \right] \\
&= \frac{1}{m} \sum_{j \in [m]} \ln \left[\left\langle \{ \beta^{g_k|_{o^j_{[n]}}} [Y_k, X_W = x_W] : k \in [d] \} \cup \{ \beta^{\mathcal{F}} \left[Y_{[p]}, X_{[d]} \right], \tau^{(A,y_A,\theta)} Y_{[p]} \} \right\rangle [\varnothing] \right] \\
&\quad - \ln \left[\left\langle \{ \beta^{g_k|_{o^j_{[n]}}} [Y_k, X_W] : k \in [d] \} \cup \{ \beta^{\mathcal{F}} \left[Y_{[p]}, X_{[d]} \right], \tau^{(A,y_A,\theta)} Y_{[p]} \} \right\rangle [\varnothing] \right] - \frac{\ln \left[\left\langle \nu^{\underline{p}} \left[X_W \right] \right\rangle [\varnothing] \right]}{m} \\
&= \mathbb{H} \left[\mathbb{P}^D, \mathbb{P}^{\mathcal{F},(A,y_A,\theta)} \right] - \frac{\ln \left[\left\langle \nu^{\underline{p}} \left[X_W \right] \right\rangle [\varnothing] \right]}{m} \, .
\end{aligned}$$

■

Let us now investigate, in which cases the Assumption 11.11 of independent data can be matched.

Example 11.14 If the p and $g_0, \ldots, g_{d-1}$ are predicates applied on variables, and the index tuples $o^j_{[n]}$ are pairwise different, then Assumption 11.11 is met. This is the case, since the values $g_k \left[O_{[n]} = o^j_{[n]} \right] |_{x_W}$ are determined by different variables in X_W. ♢

There are situations, where Assumption 11.11 is violated.

- 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

Remark 11.15 (Approximation by Independent Samples) As argued above, we do not have independent samples in general. As a consequence, we cannot apply Lem. 11.13 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.

♢

11.5.6 Sample Extraction from First-Order Logic Worlds

We have observed that in certain situations the log-likelihood of a first-order logic world with respect to a Hybrid First-Order Logic Network coincides with the likelihood of a data set in a propositional Hybrid Logic Network. Let us now investigate the extraction process of these data set. 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 11.16 Given a first-order logic world x_W, an importance formula p and extraction formulas g_k for $k \in [d]$, we define the extraction relation

$$\mathcal{R} \subset \left(\bigtimes_{l \in [n]} [r] \right) \otimes \left(\bigtimes_{k \in [d]} [2] \right)$$

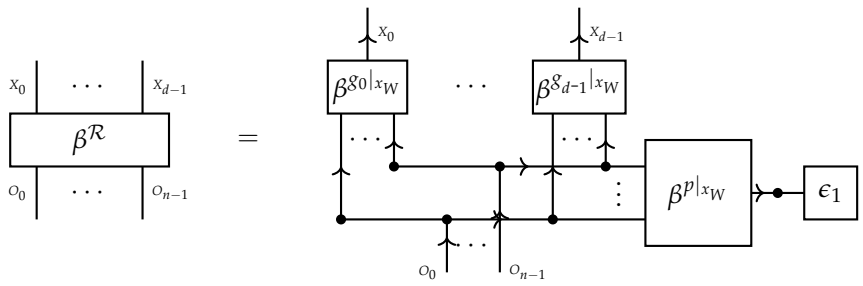
by

$$\mathcal{R} = \left\{ (o_{[n]}, x_{[d]}) \,:\, p|_{x_W} \left[O_{[n]} = o_{[n]} \right] = 1, \forall k \in [d] : x_k = g_k \left[O_{[n]} = o_{[n]} \right] \right\}.$$

The encoding of an extraction relation is the tensor

$$\beta^{\mathcal{R}} \left[O_{[n]}, X_{[d]} \right] \subset \left(\bigotimes_{l \in [n]} \mathbb{R}^r \right) \otimes \left(\bigotimes_{k \in [d]} \mathbb{R}^2 \right)$$

and drawn in a contraction diagram by



Here the contraction of β^p with the truth vector ϵ_1 represents the matching condition posed by p when extracting pairs of individuals.

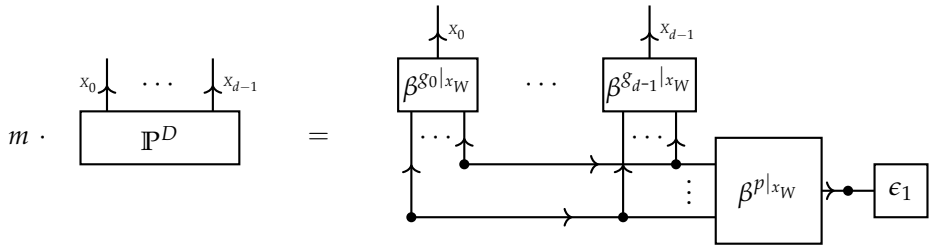
The empirical distribution of the extracted data is then the normalized contraction leaving only the legs to the extracted atomic formulas open, that is

$$\mathbb{P}^D = \frac{\left\langle \beta^{\mathcal{R}} \right\rangle \left[X_{[d]} \right]}{\left\langle \beta^{\mathcal{R}} \right\rangle [\varnothing]}.$$

Here the number of extracted data is the denominator

$$m = \left\langle \beta^{\mathcal{R}} \right\rangle [\varnothing] = \left\langle \beta^{p} \left[Y_p, O_{[n]} \right], \epsilon_1 \left[Y_p \right] \right\rangle [\varnothing].$$

We depict this by



To connect with the empirical distribution introduced in Sect. 3.3.1 we now show how the empirical distribution extracted from the interpretations of the formulas $p, g_0, \ldots, g_{d-1}$ on a first-order logic world x_W can be represented by tensor networks.

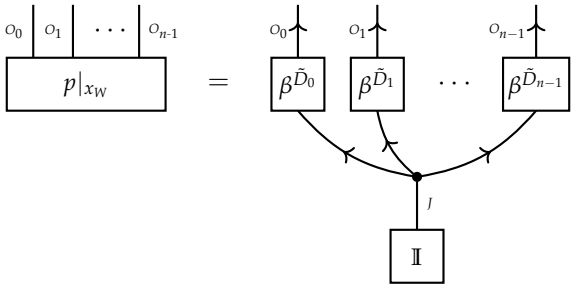
First of all, we decompose the importance formula into a basis CP format (see Chapter 15), that is a decomposition

$$p|_{x_W}\left[O_{[n]}\right] = \left\langle \{\rho^l\left[O_l, J\right] : l \in [n]\}\right\rangle \left[O_{[n]}\right]$$

such that all $\rho^l\left[O_l, J\right]$ 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 basis encoding of an term selection map $\tilde{D}$ with coordinate maps defined such that

$$\beta^{\tilde{D}_l}\left[O_l, J\right] = \rho^l\left[O_l, J\right] .$$

We depict this decomposition by:



Based on these construction, we now provide a tensor network decomposition of the extracted empirical distribution.

Theorem 11.17 *Given a first-order logic world x_W, an importance formula p and extraction formulas g_k for $k \in [d]$, we have*

$$\beta^{\mathcal{R}}\left[O_{[n]}, X_{[d]}\right] = \left\langle \{\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right] : k \in [d]\} \cup \{\beta^{\tilde{D}_l}\left[O_l, J\right] : l \in [n]\}\right\rangle \left[O_{[n]}, X_{[d]}\right]$$

and thus

$$\mathbb{P}^D\left[X_{[d]}\right] = \frac{1}{m}\left\langle \{\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right] : k \in [d]\} \cup \{\beta^{\tilde{D}_l}\left[O_l, J\right] : l \in [n]\}\right\rangle \left[X_{[d]}\right] .$$

Proof. To show the first claim, let us choose arbitrary state tuples $o_{[n]}$ and $x_{[d]}$. We then have

$$\begin{aligned}&\left\langle \{\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right] : k \in [d]\} \cup \{\beta^{\tilde{D}_l}\left[O_l, J\right] : l \in [n]\}\right\rangle \left[O_{[n]} = o_{[n]}, X_{[d]} = x_{[d]}\right] \\ &\quad = \left\langle \{\beta^{g_k|_{x_W}}\left[X_k = x_k, O_{[n]} = o_{[n]}\right] : k \in [d]\} \cup \{\beta^{\tilde{D}_l}\left[O_l = o_l, J\right] : l \in [n]\}\right\rangle [\varnothing] .\end{aligned}$$

This contraction evaluates to 1, if and only if for all $k \in [d]$ we have $\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right] = 1$ and

$$\left\langle \{\beta^{\tilde{D}_l}\left[O_l = o_l, J\right] : l \in [n]\}\right\rangle [\varnothing] = 1\,.$$

The first condition is equal to $x_k = g_k\left[O_{[n]} = o_{[n]}\right]$ 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. 11.16), we notice that these conditions are equal to $(o_{[n]}, x_{[d]}) \in \mathcal{R}$ and therefore to

$$\beta^{\mathcal{R}}\left[O_{[n]} = o_{[n]}, X_{[d]} = x_{[d]}\right]\,.$$

The first claim follows, since $\beta^{\mathcal{R}}$ is boolean, as is the contraction of the cores $\beta^{g_k|_{x_W}}$ with the cores $\beta^{\tilde{D}_l}$, which leaves the outgoing variables $X_{[d]}$ open. The second claim follows from the first using that $\mathbb{P}^D\left[X_{[d]}\right] = \frac{1}{m}\left\langle \beta^{\mathcal{R}}\right\rangle\left[X_{[d]}\right]$. ∎

To connect with the representation of empirical distributions based on data cores (see Sect. 3.3.1), we now form data cores by contractions with the grounding of extraction formulas with the cores $\beta^{\tilde{D}_l}$ (see Figure 11.3),

$$\beta^{D_k}[X_k, J] = \left\langle \{\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right]\} \cup \{\rho^l\left[O_l, J\right] : l \in [n]\}\right\rangle [X_k, J]\,.$$

The empirical distribution is then a tensor network of these tensors, as we show next.

Theorem 11.18 *We have*

$$\left\langle \beta^{\mathcal{R}}\right\rangle\left[X_{[d]}\right] = \left\langle \{\beta^{D_k}[J, X_k] : k \in [d]\}\right\rangle\left[X_{[d]}\right]$$

and thus

$$\mathbb{P}^D\left[X_{[d]}\right] = \frac{1}{m}\left\langle \{\beta^{D_k}[J, X_k] : k \in [d]\}\right\rangle\left[X_{[d]}\right]\,.$$

Proof. By Thm. 11.17 we have

$$\beta^{\mathcal{R}}\left[O_{[n]}, X_{[d]}\right] = \left\langle \{\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right] : k \in [d]\} \cup \{\beta^{\tilde{D}_l}\left[O_l, J\right] : l \in [n]\}\right\rangle\left[O_{[n]}, X_{[d]}\right]\,.$$

Since $\beta^{\tilde{D}_l}\left[O_l, J\right]$ are directed and boolean, they can be copied and separately contracted with each $g_k|_{x_W}$, without changing the contraction. We arrive at

$$\begin{aligned}
&\beta^{\mathcal{R}}\left[O_{[n]}, X_{[d]}\right] \\
&\quad = \left\langle \{\left\langle \{\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right]\} \cup \{\beta^{\tilde{D}_l}\left[O_l, J\right] : l \in [n]\}\right\rangle [X_k, J] : k \in [d]\}\right\rangle\left[O_{[n]}, X_{[d]}\right] \\
&\quad = \left\langle \{\beta^{D_k}[J, X_k] : k \in [d]\}\right\rangle\left[X_{[d]}\right]\,,
\end{aligned}$$

which established the claim. ∎

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. 5.4.2),

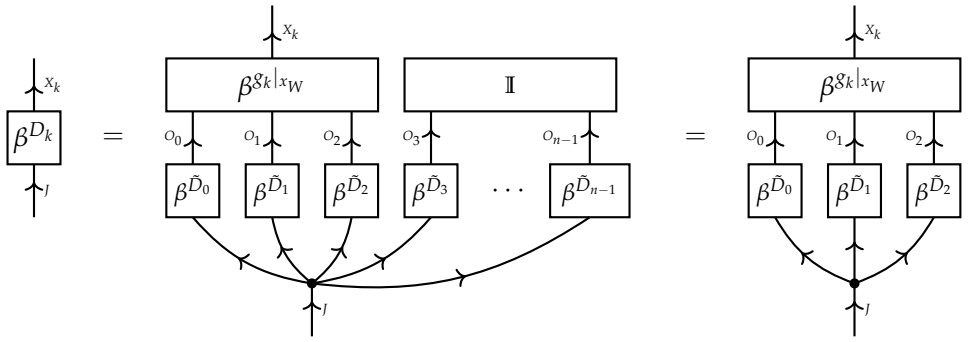


Figure 11.3: Generation of a data core for the variable X_k given an extraction formula g_k and an importance formula, which grounding is decomposed into a basis CP format with leg vectors $\beta^{\tilde{D}_l}[O_l, J]$. Term variables, which are appearing in the importance formula, but not in the extraction formula g_k can be treated trivally by contraction with the trivial tensor (here $O_4, \ldots, O_{n-1}$).

treated as a categorical variable, with respect to the constant in the extraction query. The advantages are that we can avoid the β-formalism and directly model the categorical distributions.

This also enables a batchwise computation of multiple SPARQL queries, which differ only in one constant.

11.6 Generation of First-Order Logic Worlds

So far we have discussed, how Probabilistic Relational Models for first-order logic 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 Hybrid First-Order Logic Network. We now want to investigate methods to reproduce an empirical distribution based on a constructed first-order logic world.

Definition 11.19 (Reproduction of Empirical Distributions) Given an empirical distribution $\mathbb{P}^D \in \bigotimes_{k\in[d]} \mathbb{R}^2$, we say that a triple $(x_W, p, g_{[d]})$ of a first-order logic world x_W an importance formula p and extraction formulas $g_{[d]} = \{g_k : k \in [d]\}$ reproduces $\mathbb{P}^D$, when

$$\mathbb{P}^D = \left\langle \{p|_{x_W}\left[O_{[n]}\right]\} \cup \{\beta^{g_k|_{x_W}}\left[X_k, O_{[n]}\right] : k \in [d]\}\right\rangle \left[X_{[d]}|\varnothing\right] .$$

Note that for distribution $\mathbb{P}$ to be reproducible, 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.

11.6.1 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 11.20 *Let there be an empirical distribution $\mathbb{P}^D$ to a sample selector map D (see Def. 3.4), we construct a world $x_W[L, O]$ with d unary predicates by*

$$x_W[L, O] = \sum_{k\in[d]} \sum_{j\in[m]\,:\,D_k(j)=1} \epsilon_k[L] \otimes \epsilon_j[O] \,.$$

We further choose a trivial importance query, that is

$$p|_{x_W}[O] = \mathbb{I}[O] \,,$$

and extraction queries coinciding with the unary predicates, that is for $k \in [d]$

$$g_k = g_k \,.$$

Then, the triple $(x_W, p, g_{[d]})$ reproduces $\mathbb{P}^D$.

Proof. By Thm. 11.18 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

$$p|_{x_W}[O = j] = 1$$

and choose

$$\tilde{D} = \delta \,.$$

For arbitrary $k \in [d]$ and $j \in [m]$ we now have

$$\begin{aligned}
\beta^{D_k}[X_k, J = j] &= \left\langle \beta^{g_k|_{x_W}}[X_k, O] \,, \rho^0[O, J = j] \right\rangle [X_k, J] \\
&= \left\langle \beta^{g_k|_{x_W}}[X_k, O] \,, \epsilon_{\tilde{D}(j)}[O] \right\rangle [X_k, J = j] \\
&= \epsilon_{D_k(j)}[X_k] \,.
\end{aligned}$$

This coincides with the slice of the data core of the CP representation of empirical distributions used in Thm. 3.5. Since the slice and the core was arbitrary, the tensor network representations in Thm. 3.5 and Thm. 11.18 are equal and thus the triple $(x_W, p, g_{[d]})$ 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 11.21 *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*

$$\mathrm{RDF}|_{x_W} = \sum_{j\in[m]} \epsilon_{I_{s_j}} O_s \otimes \epsilon_{I_{\mathrm{rdftype}}} O_p \otimes \epsilon_{I_C} O_o + \sum_{j\in[m]} \sum_{k\in[d]\,:\,D_k(j)=1} \epsilon_{I_{s_j}} O_s \otimes \epsilon_{I_{\mathrm{rdftype}}} O_p \otimes \epsilon_{I_{C_k}} O_o \,.$$

We further define an importance formula by the SPARQL *query*

p= *SELECT { ?x } WHERE { ?x* rdf : type C .}

and for each $k \in [d]$ *an extraction formula by the query*

g_k = *SELECT { ?x } WHERE { ?x* rdf : type C_k .} .

Then the triple $(\mathrm{KG}|_{x_W}, p, g_{[d]})$ *reproduces* $\mathbb{P}^D$.

Proof. We show the theorem analogously to Thm. 11.20, with the slide difference in the importance formula. We have for the grounding of p on $\mathrm{KG}|_{x_W}$ that

$$p|_{x_W}[O] = \sum_{j\in[m]} \epsilon_{I_{s_j}} O$$

and enumerate the non-vanishing coordinates by J.

For each extraction formula we have

$$g_k|_{x_W}[O] = \sum_{j\in[m]\,:\,D_k(j)=1} \epsilon_{I_{s_j}} O\,.$$

It follows that the data cores used in Thm. 11.18 are

$$\beta^{D_k}[X_k, j] = \epsilon_0[X_k] \otimes \left(\sum_{j\in[m]\,:\,D_k(j)=0} \epsilon_j[J] \right) + \epsilon_1[X_k] \otimes \left(\sum_{j\in[m]\,:\,D_k(j)=1} \epsilon_j[J] \right)$$

and they thus coincide with those in the decomposition in Thm. 3.5. The claim follows therefore with the same argumentation as in the proof of Thm. 11.20. ∎

Let us provide some more insights on the construction of the reproducing Knowledge Graph in Thm. 11.21. By the insertions to the one-hot encodings $\epsilon_{I_{s_j}} O_s \otimes \epsilon_{I_{\mathrm{rdftype}}} O_p \otimes \epsilon_{I_C} O_o$ we mark each sample representing resource by a class and ensure its appearance as a owl : NamedIndividual in the graph. The insertions $\epsilon_{I_{s_j}} O_s \otimes \epsilon_{I_{\mathrm{rdftype}}} O_p \otimes \epsilon_{I_{C_k}} O_o$ 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 $g_0, \ldots, g_{d-1}$.

11.6.2 Samples by Tuples 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 11.22 *Let there a data map* D*, queries* $p, g_{[d]}$ *and a first-order logic world containing objects* $s_{j,l}$ *for* $j \in [m]$ *and* $l \in [n]$ *If*

$$p|_{x_W} = \sum_{j\in[m]} \bigotimes_{l\in[n]} \epsilon_{I_{s_{j,l}}}[O_l]$$

and for any $k \in [d]$

$$g_k|_{x_W} = \sum_{j:D_k(j)=1} \bigotimes_{O_l \in O_{g_k}} \epsilon_{I_{s_{j,l}}}[O_l]\,.$$

Then the tuple $(\mathrm{KG}|_{x_W}, p, \{g_0, \ldots, g_{d-1}\})$ *reproduces* $\mathbb{P}^D$.

Proof. We notice that the grounding of the importance formula is in a basis CP format, since by assumption

$$p|_{x_W} = \sum_{j\in[m]} \bigotimes_{l\in[n]} \epsilon_{I_{s_{j,l}}} [O_l] \ .$$

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

$$\left\langle \{\beta^{g_k|_{x_W}} [X_k, O_{g_k}]\} \cup \{\beta^{\tilde{D}_l} [O_l, J] \ : \ l \in [n]\} \right\rangle [X_k, J] = \beta^{D_k} [X_k, J]$$

and the claim follows with the same argumentation as in the proof of Thm. 11.20. ∎

11.7 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 [Bar16; Gio17], 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 11.23 (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

12

Introduction to Part III

In view of all that..., the many obstacles we appear to have surmounted, what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientists from earliest days. - Richard Bellman [Bel61]

In Part III we provide a more general discussion of tensor calculus, which has been applied in Part I and Part II. Since the contraction operation is central in these calculus schemes, we refer to the set of techniques as contraction calculus.

12.1 Encoding Schemes for Functions

We investigate in more detail the following encoding schemes for functions:

- **coordinate encodings** use the real coordinates multiplying each one-hot encoding to store information. They have implicitly been used in factored representation of systems in Part I, where probability distributions and logical formulas have been treated as tensors. More precise, these tensors are the coordinate encodings of probability distributions and logical formulas, which are maps of the state set into the interval $[0, 1]$ and into the set $\{0, 1\}$. We investigate these encodings in more detail in Chapter 13.

- **basis encodings** are sums of a collection of one-hot encodings. They differ from coordinate encodings that they do not allow weights by general real numbers in these sums, and restrict to booleans. Basis encodings therefore map the set of subsets to an enumerated set bijectively onto the set of boolean tensors. We introduce them in Chapter 14 in most generality as an encoding scheme for subsets of enumerated set, which we then extend towards relations and functions. The main advantage of basis encodings is the efficient representation of function compositions by tensor networks of basis encodings. This scheme is the main representation paradigm to derive efficient tensor network decompositions in artificial intelligence.

- **Selection encodings** are specific coordinate encodings for tensor-valued functions on the state set of a factored representation. Here selection variables are introduced to enumerate the coordinates of the target tensor space and treated as additional variables of a factored system representation. We have exploited them in Part II for efficient representation of function sets, where the functions rely on a common structure.

12.2 Schemes for Tensor Calculus

We applied in Part I and Part II two schemes of tensor calculus, which we now investigate in Part III in more detail.

- **Coordinate calculus:** We study in Chapter 13 schemes to exploit coordinate encodings in calculus. Retrieval of single coordinates is done by contractions of the tensors with one-hot encodings, with no open variables (see Thm. 13.3).

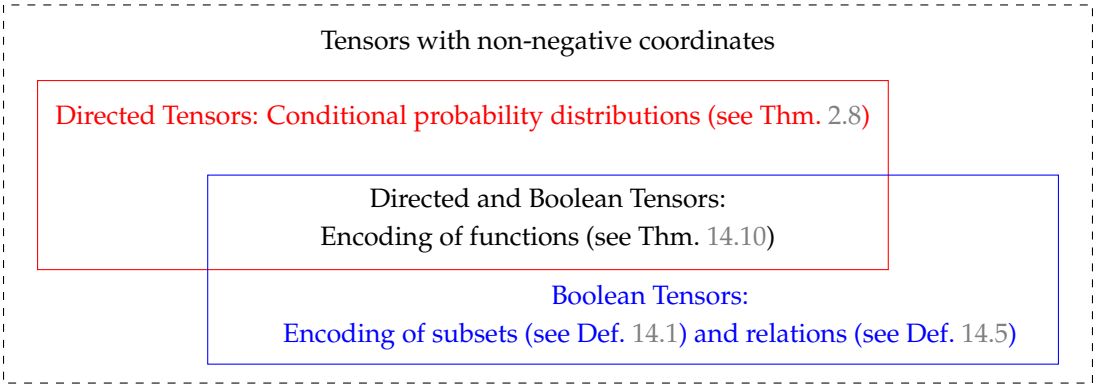


Figure 12.1: Sketch of the tensors with non-negative coordinates. We investigate in this chapter tensors, which are directed and boolean.

- **Basis calculus:** In Chapter 14 we investigate the properties of basis encodings to perform calculus. The evaluation of a function at a state is performed by contractions with one-hot encodings, with the computed variable left open (see Thm. 14.11). As the main advantage of basis encodings over coordinate encodings, we can represent composed function by contractions of basis encodings to each component (see Thm. 14.12).

12.3 Classification of Tensors

We frequently worked in Part I and Part II with tensors, which have non-negative coordinates and occasionally are boolean (see Def. 0.5) or directed (see Def. 0.17). While boolean tensors have appeared as semantical representation of formulas, directed tensors have appeared mostly as conditional distributions. The set of tensors, which are both boolean and directed receive further interest, since they are exactly the basis encodings of functions. We sketch this coarse classification scheme in Figure 12.1.

12.4 Efficient Representation and Reasoning

As a main topic of this work, tensor networks are motivated to provide efficient representation and reasoning schemes in artificial intelligence. More precisely, we investigate in Part III three aspects of efficiency:

- **Efficient storage schemes** can be derived for tensors with specific properties, whereas generic tensor storage schemes suffer from the curse of dimensions. In Chapter 15 we investigate the efficient representation of tensors based on CP decompositions. When restricting the allowed leg tensors in different ways, we show that relational databases can be utilized as sparse storage of tensors.
- **Efficient executions of contractions** will be derived based on message-passing schemes in Chapter 17. They are in most generality derived from expectation propagation schemes in variational inference.
- **Approximation schemes** to tensors are investigated in Chapter 16. Approximating formats are oriented on the efficient representation of the approximating tensor.

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 show, that the one-hot encodings, which we have used to motivate tensor representations, build an orthonormal basis of the respective tensor spaces.

Lemma 13.1 *The image of the one-hot encoding map is an orthonormal basis of the tensor space* $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$*, that is for any* $x_{[d]}, \tilde{x}_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ *we have*

$$\left\langle \epsilon_{x_{[d]}}\left[X_{[d]}\right], \epsilon_{\tilde{x}_{[d]}}\left[X_{[d]}\right]\right\rangle[\varnothing] = \delta_{x_{[d]},\tilde{x}_{[d]}} := \begin{cases} 1 & \text{if} \quad x_{[d]} = \tilde{x}_{[d]} \\ 0 & \text{else} \end{cases}.$$

Any element $\tau \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ *has a decomposition*

$$\tau\left[X_{[d]}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \tau\left[X_{[d]} = x_{[d]}\right] \cdot \epsilon_{x_{[d]}}\left[X_{[d]}\right].$$

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

$$\left\langle \epsilon_{x_{[d]}}\left[X_{[d]}\right], \epsilon_{\tilde{x}_{[d]}}\left[X_{[d]}\right]\right\rangle[\varnothing] = \prod_{k\in[d]} \left\langle \epsilon_{x_k}[X_k], \epsilon_{\tilde{x}_k}[X_k]\right\rangle[\varnothing] = \prod_{k\in[d]} \delta_{x_k,\tilde{x}_k} = \delta_{x_{[d]},\tilde{x}_{[d]}}.$$

To show the second claim, it is enough to notice that for any $\tilde{x}_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have

$$\begin{aligned}\sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \tau\left[X_{[d]} = x_{[d]}\right] \cdot \epsilon_{x_{[d]}}\left[X_{[d]} = \tilde{x}_{[d]}\right] &= \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \tau\left[X_{[d]} = x_{[d]}\right] \cdot \delta_{x_{[d]},\tilde{x}_{[d]}} \\ &= \tau\left[X_{[d]} = \tilde{x}_{[d]}\right].\end{aligned}$$

■

Any tensor can be understood as a coordinate encoding of a real-valued function, as we define next.

Definition 13.2 Given any real-valued function

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

we define the coordinate encoding by

$$\chi^q \left[X_{[d]}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} q(x_{[d]}) \cdot \epsilon_{x_{[d]}} \left[X_{[d]}\right] .$$

In Part I and Part II we did not distinguish between a real-valued function q and its coordinate encoding τ^q, in order to abbreviate notation. Based on coordinate encodings, we now show, that function evaluation can be performed by contractions.

Theorem 13.3 (Function evaluation in Coordinate Calculus) *Given any real-valued function*

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

and any input state $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$*, we have*

$$q(x_{[d]}) = \left\langle \chi^q \left[X_{[d]}\right], \epsilon_{x_{[d]}} \left[X_{[d]}\right] \right\rangle [\varnothing] .$$

Proof. We use the decomposition in Lem. 13.1 and have by linearity of contractions for any index tuple $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$

$$\begin{aligned}
\left\langle \chi^q \left[X_{[d]}\right], \epsilon_{x_{[d]}} \left[X_{[d]}\right] \right\rangle [\varnothing] &= \sum_{\tilde{x}_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \chi^q \left[X_{[d]} = \tilde{x}_{[d]}\right] \cdot \left\langle \epsilon_{\tilde{x}_{[d]}} \left[X_{[d]}\right], \epsilon_{x_{[d]}} \left[X_{[d]}\right] \right\rangle [\varnothing] \\
&= \sum_{\tilde{x}_{[d]}\in\bigtimes_{k\in[d]}[m_k]} q(\tilde{x}_{[d]}) \cdot \delta_{\tilde{x}_{[d]}, x_{[d]}} \\
&= q(x_{[d]})
\end{aligned}$$

where we used that one-hot encodings are orthonormal. ∎

Coordinate calculus is the representation of real-valued functions as tensors, from which its evaluations can be retrieved by the scheme of Thm. 13.3. This is in contrast to the basis calculus scheme to be discussed (see Thm. 14.11), 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 13.4 *Given a tensor network* $\tau^{\mathcal{G}}$ *on a hypergraph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, disjoint subsets* $A, B \subset \mathcal{V}$ *and* $x_B \in [m_B]$*, we have*

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_A, X_B = x_B] = \langle \{ \langle \tau^e \rangle [X_{e/B}, X_{e\cap B} = x_{e\cap B}] : e \in \mathcal{E} \} \rangle [X_A] .$$

Proof. By definition of contractions we have for any x_A

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_A = x_A, X_B = x_B] = \sum_{x_{\mathcal{V}/(A\cup B)}\in\bigtimes_{v\in\mathcal{V}/(A\cup B)}[m_v]} \prod_{e\in\mathcal{E}} \tau^e [X_{e/B} = x_{e/B}, X_{e\cap B} = x_{e\cap B}]$$

$$= \left\langle \{ \langle \tau^e \rangle [\varnothing] X_{e/(A\cup B)}, X_{e\cap A} = x_{e\cap A}, X_{e\cap B} = x_{e\cap B} : e \in \mathcal{E} \} \right\rangle [\varnothing]$$
$$= \langle \{ \langle \tau^e \rangle [\varnothing] X_{e/B}, X_{e\cap B} = x_{e\cap B} : e \in \mathcal{E} \} \rangle [X_A = x_A]$$

and the claim follows. ∎

If we retrieve a single coordinate of a tensor, we have the situation $A = \varnothing$, $B = \mathcal{V}$. In that case, Thm. 13.4 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 13.5 Let $h : \mathbb{R}^p \to \mathbb{R}$ be a function. Then the coordinatewise transform of tensors $\tau^l \left[X_{[d]} \right]$, where $l \in [p]$, under q is the tensor

$$h \left(\tau^0, \ldots, \tau^{p-1} \right) \left[X_{[d]} \right]$$

with coordinates

$$h \left(\tau^0, \ldots, \tau^{p-1} \right) \left[X_{[d]} = x_{[d]} \right] = h \left(\tau^0 \left[X_{[d]} = x_{[d]} \right], \ldots, \tau^{p-1} \left[X_{[d]} = x_{[d]} \right] \right) .$$

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 Thm. 14.13.

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 13.6 *For any function $h : \mathbb{R} \to \mathbb{R}$, any tensor $\tau \left[X_{[d]} \right]$ and index x_A, where $A \subset [d]$, we have*

$$h \left(\tau \left[X_{[d]} \right] \right) \left[X_{[d]/A}, X_A = x_A \right] = h \left(X_{[d]/A}, X_A = x_A \right) \left[X_{[d]/A} \right] .$$

Proof. For any state $x_{[d]/A}$ we have that

$$h \left(\tau \left[X_{[d]} \right] \right) \left[X_{[d]/A} = x_{[d]/A}, X_A = x_A \right] = h \left(\tau \left[X_{[d]} = x_{[d]} \right] \right)$$
$$= h \left(X_{[d]/A}, X_A = x_A \right) \left[X_{[d]/A} = x_{[d]/A} \right] .$$ ∎

Example 13.7 (Hadamard products as coordinatewise transforms) Hadamard products of tensors (see Example 0.15) are a special way of coordinate calculus, where the transform is the product and thus

$$\cdot \left(\tau^0, \ldots, \tau^{p-1} \right) \left[X_{[d]} \right] = \left\langle \{ \tau^l \left[X_{[d]} \right] : l \in [p] \} \right\rangle \left[X_{[d]} \right] .$$

These hadamard products are applied in the effective computation of conjunctions, as we will discuss in more detail in Sect. 14.6. ♢

Example 13.8 (Exponentiation of energies) In Def. 2.21 we introduced exponential families, based on the exponentiation of energies. For a statistic $\mathcal{S}$, a base measure ν and a canonical parameter θ we defined

$$\mathbb{P}^{(\mathcal{S},\theta,\nu)} = \frac{\left\langle \exp\left[\left\langle \sigma^{\mathcal{S}}, \theta \right\rangle \left[X_{[d]}\right], \nu\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}\right]}{\left\langle \exp\left[\left\langle \sigma^{\mathcal{S}}, \theta \right\rangle \left[X_{[d]}\right], \nu\left[X_{[d]}\right]\right]\right\rangle [\varnothing]} .$$

Both the nominator and the denominator involve a coordinatewise transform of the energy tensor $\phi^{(\mathcal{S},\theta,\nu)}[X_{[d]}]$ by the exponentiation. Thm. 2.24 provided a transform-free contraction expression by basis encodings, which is the central tool of basis calculus (see Chapter 14).

Let us note, that Lem. 13.6 enables the energy-based answering of conditional queries, as has been shown in Thm. 3.3. ♢

13.3 Directed Tensors

Directionality as defined in Def. 0.17 is a constraint on the structure of a tensor, namely that the contraction leaving only incoming variables open trivializes the tensor. We have motivated such constraints by conditional distributions, see Def. 2.13, and referred to Markov Networks (see Def. 2.41) satisfying these by Bayesian Networks (see Def. 2.46). To support our findings therein, we now discuss in more detail the connection between directed hypergraphs and directed tensors.

Definition 13.9 (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 τ^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 basis encodings of invertible functions.

Example 13.10 (Dirac Delta Tensors) Given a set of variables $X_{[d]} = X_0, \ldots, X_{d-1}$ with identical dimension m, Diracs Delta Tensor is the element

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

with coordinates

$$\delta^{[d],m}\left[X_{[d]} = x_{[d]}\right] = \begin{cases} 1 & \text{if} \quad x_0 = \ldots = x_{d-1} \\ 0 & \text{else} \end{cases} .$$

The contractions with respect to subsets $\tilde{\mathcal{V}} \subset [d]$ are

$$\left\langle \delta^{[d],m} \right\rangle [X_{\tilde{\mathcal{V}}}] = \begin{cases} m & \text{if} \quad \tilde{\mathcal{V}} = \varnothing \\ \mathbb{I}\,[X_{\tilde{\mathcal{V}}}] & \text{if} \quad \left|\tilde{\mathcal{V}}\right| = 1 \\ \delta^{\tilde{\mathcal{V}},m}\,[X_{\tilde{\mathcal{V}}}] & \text{else} \end{cases} .$$

Thus are directed for any orientation of the respective edge with exactly one incoming variable. ♢

We can use Diracs Delta Tensors to represent any contraction of a tensor network on a hypergraph by a tensor network on a graph, as we show next.

Lemma 13.11 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a hypergraph and $\tau^{\mathcal{G}}$ 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 $\tau^{\tilde{\mathcal{G}}}$ by*

- *Recolored Edges $\tilde{\mathcal{E}} = \{\tilde{e} : e \in \mathcal{E}\}$ where $\tilde{e} = \{v^e : v \in e\}$, which decoration tensor $\tau^{\tilde{e}}$ has same coordinates as τ^e*
- *Nodes $\tilde{\mathcal{V}} = \bigcup_{e \in \mathcal{E}} \tilde{e}$*
- *Delta Edges $\Delta^{\mathcal{G}} = \{\{v\} \cup \{v^e : e \ni v\} : v \in \mathcal{V}\}$ each of which decorated by a delta tensor $\delta^{\{v^e : e \ni v\}}$*

Then we have

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

Proof. For any $x_{\mathcal{V}}$ we have

$$\begin{aligned}
\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] &= \left\langle \{\tau^{\tilde{e}}[X_{\{v^e : v \in e\}}] : e \in \mathcal{E}\} \cup \{\delta^{\{v\} \cup \{v^e : e \ni v\}}[X_{\{v^e : e \ni v\}}, X_v = x_v] : v \in \mathcal{V}\} \right\rangle [\varnothing] \\
&= \left\langle \{\tau^{\tilde{e}}[X_{\{v^e : v \in e\}} = x_{\{v : v \in e\}}] : e \in \mathcal{E}\} \right\rangle [\varnothing] \\
&= \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \ ,
\end{aligned}$$

which establishes the claim. ∎

13.3.1 Normalization

Normalized tensors (see Def. 0.18) are directed and directed tensors invariant under normalization wrt their incoming and outgoing variable, as we show next.

Theorem 13.12 *For any tensor network $\tau^{\mathcal{G}}$ on variables $\mathcal{V}$ that can be normalized with respect to $\mathcal{V}^{\text{in}}$ and $\mathcal{V}^{\text{out}}$, the normalization 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

$$\left\langle \left\langle \tau^{\mathcal{G}} \right\rangle \left[\mathcal{V}^{\text{in}} | \mathcal{V}^{\text{out}}\right], \epsilon_{x_{\mathcal{V}^{\text{in}}}} \right\rangle [\varnothing] = \frac{\left\langle \tau^{\mathcal{G}} \cup \{\epsilon_{x_{\mathcal{V}^{\text{in}}}}\} \right\rangle [\varnothing]}{\left\langle \tau^{\mathcal{G}} \cup \{\epsilon_{x_{\mathcal{V}^{\text{in}}}}\} \right\rangle [\varnothing]} \ .$$

By Def. 0.17, $\left\langle \tau^{\mathcal{G}} \right\rangle \left[\mathcal{V}^{\text{out}} | \mathcal{V}^{\text{in}}\right]$ is thus directed. ∎

The normalization operation coincides in cases of non-negative tensors with the conditioning of a Markov Network representing a probability distribution.

13.3.2 Normalization Equations

Normalization equations capture certain properties of normalizations of tensors. We first show that any normalizable tensor is the contraction of its normalization and an accompanying contraction, which generalizes the Bayes Thm. 2.9 towards more generic normalizable tensors.

Theorem 13.13 (normalization as a Contraction Equation) *For any on $\mathcal{V}^{\text{in}}$ normalizable tensor $\tau\left[X_{\mathcal{V}}\right]$, where $\mathcal{V}^{\text{in}} \dot{\cup} \mathcal{V}^{\text{out}} = \mathcal{V}$, we have*

$$\langle\tau\rangle\left[X_{\mathcal{V}}\right] = \left\langle \langle\tau\rangle\left[X_{\mathcal{V}^{\text{out}}} | X_{\mathcal{V}^{\text{in}}}\right], \langle\tau\rangle\left[X_{\mathcal{V}^{\text{in}}}\right]\right\rangle\left[X_{\mathcal{V}}\right] .$$

Proof. Let us choose indices $x_{\mathcal{V}^{\text{in}}}$ and $x_{\mathcal{V}^{\text{out}}}$. We have that

$$\langle\tau\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}} | X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right] = \frac{\langle\tau\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}, X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right]}{\langle\tau\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}\right]}$$

and therefor

$$\langle\tau\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}, X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right] = \langle\tau\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}} | X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right] \cdot \langle\tau\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}\right]$$

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 Thm. 2.10.

Theorem 13.14 (Generic Chain Rule) *For any Tensor $\tau\left[X_{\mathcal{V}}\right]$ and any total order $\prec$ on the nodes $\mathcal{V}$ we have*

$$\tau\left[X_{\mathcal{V}}\right] = \left\langle \left\{ \langle\tau\rangle\left[X_v | X_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}}\right] \,:\, v \in \mathcal{V} \right\} \right\rangle\left[X_{\mathcal{V}}\right] ,$$

provided that the normalizations exist.

Proof. We apply Thm. 13.13 on the tensor

$$\langle\tau\rangle\left[X_v, X_{\{\tilde{v}\,:\,v\prec\tilde{v}, \tilde{v}\neq v\}} | X_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}} = x_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}}\right] ,$$

where $v \in \mathcal{V}$ and $x_{\mathcal{V}}$ are chosen arbitrarly. For any $v \in \mathcal{V}$ we get

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

Applying this equation iteratively and making use of the commutation of contractions we get for any $v \in \mathcal{V}$

$$\langle\tau\rangle\left[X_v, X_{\{\tilde{v}\,:\,v\prec\tilde{v}, \tilde{v}\neq v\}} | X_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}}\right] = \left\langle \langle\tau\rangle\left[X_{\tilde{v}} | X_{\{\bar{v}:\bar{v}\prec\tilde{v}, \bar{v}\neq\tilde{v}\}}\right] \,:\, v \prec \tilde{v} \right\rangle\left[X_{\mathcal{V}}\right] .$$

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 13.15 *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 $\tau^{\mathcal{G}}$ respecting the direction of $\mathcal{G}$ we have that*

$$\left\langle \tau^{\mathcal{G}} \right\rangle\left[X_{\mathcal{V}^{\text{in}}}\right] = \mathbb{I}\left[X_{\mathcal{V}^{\text{in}}}\right] ,$$

that is $\langle \tau^{\mathcal{G}} \rangle [X_{\mathcal{V}}]$ 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. 13.11 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 $\tau^{\mathcal{G}}$ consists of a single core.

$n - 1 \to n$: Let us assume, that the claim holds for graphs with $n-1$ hyperedges and let $\tau^{\mathcal{G}}$ 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 Thm. 17.1 and get

$$\begin{aligned}\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}^{\text{in}}}] &= \left\langle \tau^{(\mathcal{V},\mathcal{E}/\{e\})} \cup \{\tau^e [X_{e^{\text{in}}}, X_{e^{\text{out}}}]\} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &= \left\langle \tau^{(\mathcal{V},\mathcal{E}/\{e\})} \cup \{\langle \tau^e \rangle [X_{e^{\text{in}}}]\} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &= \left\langle \tau^{(\mathcal{V},\mathcal{E}/\{e\})} \cup \{\mathbb{I} [X_{e^{\text{in}}}]\} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &\left\langle \tau^{(\mathcal{V},\mathcal{E}/\{e\})} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \ .\end{aligned}$$

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

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}^{\text{in}}}] = \left\langle \tau^{(\mathcal{V},\mathcal{E}/\{e\})} \right\rangle [X_{\mathcal{V}^{\text{in}}}] = \mathbb{I} [X_{\mathcal{V}^{\text{in}}}] \ . \qquad \blacksquare$$

13.4 Proof of Hammersley-Clifford Theorem

Let us now proof the Hammersley-Clifford theorem formulated in Chapter 2 as Thm. 2.44. Different to the original statement (see [CH71]), we here proof the analogous statement for hypergraphs, where we have to demand the property of clique-capturing defined in Def. 2.43. We start with showing the following Lemmata to be exploited in the proof.

Lemma 13.16 *Let $\tau [X_{\mathcal{V}}]$ be a positive tensor and $y_{\mathcal{V}}$ an arbitrary index. Then we have*

$$\tau [X_{\mathcal{V}}] = \left\langle \left(\langle \tau \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} : \bar{\mathcal{V}} \subset \tilde{\mathcal{V}} \subset \mathcal{V} \right\rangle [X_{\mathcal{V}}] \ ,$$

where the exponentiation is performed coordinatewise and positivity of τ ensures the well-definedness.

Proof. It suffices to show, that for an arbitrary index $x_{\mathcal{V}}$ be an arbitrary index we have

$$\tau [X_{\mathcal{V}} = x_{\mathcal{V}}] = \prod_{\bar{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle \tau \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} .$$

We do this by applying a logarithm on the right hand side and grouping the terms by $\bar{\mathcal{V}}$ as

$$\ln \left[\prod_{\bar{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \langle \tau \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} \Bigg]$$

$$= \sum_{\bar{\mathcal{V}} \subset \mathcal{V}} \ln\left[\langle\tau\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right]\right] \left(\sum_{\tilde{\mathcal{V}} \subset \mathcal{V} \,:\, \bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} (-1)^{|\tilde{\mathcal{V}}| - |\bar{\mathcal{V}}|} \right)$$

$$= \sum_{\bar{\mathcal{V}} \subset \mathcal{V}} \ln\left[\langle\tau\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right]\right] \left(\sum_{i \in [|\mathcal{V}| - |\bar{\mathcal{V}}|]} (-1)^i \binom{|\mathcal{V}| - |\bar{\mathcal{V}}|}{i} \right)$$

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

$$\begin{aligned} &\ln\left[\prod_{\bar{\mathcal{V}} \subset \mathcal{V}} \prod_{\tilde{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle\tau\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right] \right)^{(-1)^{|\tilde{\mathcal{V}}| - |\bar{\mathcal{V}}|}} \right] \\ &= \ln\left[\tau\left[X_{\mathcal{V}} = x_{\mathcal{V}}\right]\right] \left(\sum_{i \in [0]} (-1)^i \binom{0}{i} \right) \\ &= \ln\left[\tau\left[X_{\mathcal{V}} = x_{\mathcal{V}}\right]\right] . \end{aligned}$$

Applying the exponential function on both sides establishes the claim. ∎

Lemma 13.17 *Let τ 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*

$$\langle\tau\rangle\left[X_{A,B} | X_{\mathcal{V}/\{A,B\}}\right] = \left\langle \langle\tau\rangle\left[X_A | X_{\mathcal{V}/\{A,B\}}\right], \langle\tau\rangle\left[X_B | X_{\mathcal{V}/\{A,B\}}\right] \right\rangle \left[X_{\tilde{\mathcal{V}}}\right]$$

then

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle\tau\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right] \right)^{(-1)^{|\tilde{\mathcal{V}}| - |\bar{\mathcal{V}}|}} = 1 .$$

Proof. We abbreviate

$$Z_{\bar{\mathcal{V}}} = \langle\tau\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right] .$$

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

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} (Z_{\bar{\mathcal{V}}})^{(-1)^{|\tilde{\mathcal{V}}| - |\bar{\mathcal{V}}|}} = \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}/\{A,B\}} \left(\frac{Z_{\bar{\mathcal{V}}} \cdot Z_{\bar{\mathcal{V}} \cup \{A,B\}}}{Z_{\bar{\mathcal{V}} \cup \{A\}} \cdot Z_{\bar{\mathcal{V}} \cup \{B\}}} \right)^{(-1)^{|\tilde{\mathcal{V}}| - |\bar{\mathcal{V}}|}} . \tag{13.1}$$

From the independence assumption it follows that for any index x

$$\begin{aligned} &\langle\tau\rangle\left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B\right] \\ &\quad = \langle\tau\rangle\left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right] \\ &\quad = \langle\tau\rangle\left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B\right] \end{aligned}$$

Applying this in each squares bracket term of (13.1) we get

$$\begin{aligned}\frac{Z_{\bar{\mathcal{V}}}}{Z_{\bar{\mathcal{V}}\cup\{A\}}} &= \frac{\langle\tau\rangle\left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B\right]}{\langle\tau\rangle\left[X_A = y_A | X_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B\right]} \\ &= \frac{\langle\tau\rangle\left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B\right]}{\langle\tau\rangle\left[X_A = y_A | X_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}}\cup\{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B\right]} \\ &= \frac{Z_{\bar{\mathcal{V}}\cup\{B\}}}{Z_{\bar{\mathcal{V}}\cup\{A,B\}}}\,.\end{aligned}$$

Thus, each factor in (13.1) is trivial, which establishes the claim. ∎

We are finally ready to prove the Hammersley-Clifford Thm. 2.44 based on the Lemmata above.

Proof of Thm. 2.44. By Lem. 13.16 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. 13.17 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

$$\tau^e\left[X_e\right] = \prod_{\tilde{\mathcal{V}}\,:\,\alpha(\tilde{\mathcal{V}})=e} \prod_{\bar{\mathcal{V}}\subset\tilde{\mathcal{V}}} \left(\mathbb{P}\left[X_{\bar{\mathcal{V}}} = x_{\bar{\mathcal{V}}}, X_{\mathcal{V}/\bar{\mathcal{V}}} = y_{\mathcal{V}/\bar{\mathcal{V}}}\right]\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}}\,.$$

and get, that

$$\begin{aligned}\mathbb{P}\left[X_{\mathcal{V}}\right] &= \langle\{\tau^e\left[X_e\right] : e \in \mathcal{E}\}\rangle\left[X_{\mathcal{V}}\right] \\ &= \langle\{\tau^e\left[X_e\right] : e \in \mathcal{E}\}\rangle\left[X_v|\varnothing\right]\,.\end{aligned}$$

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 3 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 13.18 *For any tensor network $\tau^{\mathcal{G}}$ with positive τ^e we have*

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

Proof. By multilinearity of tensor network contractions we have

$$\frac{\partial}{\partial \tau^e[Y_e]} \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}] = \langle \{\delta[Y_e, X_e]\} \cup \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \neq e\} \rangle [Y_e, X_{\mathcal{V}}]$$

and thus

$$\frac{\partial}{\partial \tau^e[Y_e]} \left\langle \tau^{\mathcal{G}} \right\rangle [\varnothing] = \langle \{\delta[Y_e, X_e]\} \cup \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \neq e\} \rangle [Y_e] .$$

Using both we get

$$\begin{aligned}
\frac{\partial}{\partial \tau^e[Y_e]} \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}|\varnothing] &= \frac{\partial}{\partial \tau^e[Y_e]} \frac{\langle \tau^{\mathcal{G}} \rangle [X_{\mathcal{V}}]}{\langle \tau^{\mathcal{G}} \rangle [\varnothing]} \\
&= \frac{\frac{\partial}{\partial \tau^e[Y_e]} \langle \tau^{\mathcal{G}} \rangle [X_{\mathcal{V}}]}{\langle \tau^{\mathcal{G}} \rangle [\varnothing]} - \frac{\langle \tau^{\mathcal{G}} \rangle [X_{\mathcal{V}}] \frac{\partial}{\partial \tau^e[Y_e]} \langle \tau^{\mathcal{G}} \rangle [\varnothing]}{(\langle \tau^{\mathcal{G}} \rangle [\varnothing])^2} \\
&= \frac{\langle \{\delta[Y_e, X_e]\} \cup \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \neq e\} \rangle [Y_e, X_{\mathcal{V}}]}{\langle \tau^{\mathcal{G}} \rangle [\varnothing]} \\
&\quad - \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}|\varnothing] \cdot \frac{\langle \{\delta[Y_e, X_e]\} \cup \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \neq e\} \rangle [Y_e]}{\langle \tau^{\mathcal{G}} \rangle [\varnothing]} \qquad (13.2)
\end{aligned}$$

For the first term we get with a normalization equation (see Thm. 13.13) that

$$\begin{aligned}
\frac{\langle \{\delta[Y_e, X_e]\} \cup \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \neq e\} \rangle [Y_e, X_{\mathcal{V}}]}{\langle \tau^{\mathcal{G}} \rangle [\varnothing]} &= \frac{\langle \{\delta[Y_e, X_e]\} \cup \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \in \mathcal{E}\} \rangle [Y_e, X_{\mathcal{V}}]}{\tau^e[X_e] \cdot \langle \tau^{\mathcal{G}} \rangle [\varnothing]} \\
&= \frac{\langle \delta[Y_e, X_e], \langle \tau^{\mathcal{G}} \rangle [X_{\mathcal{V}}|\varnothing] \rangle [Y_e, X_{\mathcal{V}}]}{\tau^e[X_e]} \\
&= \frac{\langle \delta[Y_e, X_e], \langle \tau^{\mathcal{G}} \rangle [X_e|\varnothing], \langle \tau^{\mathcal{G}} \rangle [X_{\mathcal{V}/e}|X_e] \rangle [Y_e, X_{\mathcal{V}}]}{\tau^e[X_e]} .
\end{aligned}$$

Analogously, we have

$$\frac{\langle \{\delta[Y_e, X_e]\} \cup \{\tau^{\tilde{e}}[X_{\tilde{e}}] : \tilde{e} \neq e\} \rangle [Y_e]}{\langle \tau^{\mathcal{G}} \rangle [\varnothing]} = \frac{\langle \delta[Y_e, X_e], \langle \tau^{\mathcal{G}} \rangle [X_e|\varnothing] \rangle [Y_e]}{\tau^e[X_e]} .$$

With (13.2), we arrive at the claim

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

Lemma 13.19 *For any function $q(\tau^e)[X_{\mathcal{V}}]$ we have*

$$\begin{aligned}
&\frac{\partial}{\partial \tau^e\left[Y_e\right]}\left\langle\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], q(\tau^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] \\
&\quad=\frac{\left\langle\tau^{\mathcal{G}}\right\rangle\left[X_e=x_e|\varnothing\right]}{\tau^e\left[X_e=x_e\right]}\Big(\left\langle\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}/e}|X_e=x_e\right], q(\tau^e)[X_{\mathcal{V}}, Y_e]\right\rangle[\varnothing] \\
&\qquad\qquad-\left\langle\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], q(\tau^e)[X_{\mathcal{V}}]\right\rangle[\varnothing]\Big) \\
&\qquad+\left\langle\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right]\frac{\partial q(\tau^e)[X_{\mathcal{V}}]}{\partial \tau^e\left[Y_e\right]}\right\rangle[\varnothing]
\end{aligned}$$

Proof. By product rule of differentiation we have

$$\begin{aligned}
\frac{\partial}{\partial \tau^e\left[X_e=x_e\right]}\left\langle\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], q(\tau^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] &= \left\langle\frac{\partial}{\partial \tau^e\left[X_e=x_e\right]}\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], q(\tau^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] \\
&\quad+\left\langle\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}|\varnothing\right], \frac{\partial}{\partial \tau^e\left[X_e=x_e\right]} q(\tau^e)[X_{\mathcal{V}}]\right\rangle[\varnothing] .
\end{aligned}$$

The claim now follows with the application of Lem. 13.18 on the first term. ∎

14

Basis Calculus

Basis Calculus stores information 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.

14.1 Basis 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 14.1 (Basis encoding of subsets) We say that an arbitrary finite 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 $\epsilon_{\mathcal{V}}[O]$ defined for $o \in [|\mathcal{U}|]$ as

$$\epsilon_{\mathcal{V}}[O = o] = \begin{cases} 1 & \text{if} \quad I(o) \in \mathcal{V} \\ 0 & \text{else} \end{cases} .$$

In a one-hot basis decomposition we have

$$\epsilon_{\mathcal{V}}[O] \coloneqq \sum_{o \in [r_{\mathcal{U}}] \,:\, I(o) \in \mathcal{V}} \epsilon_o[O] .$$

The inclusion of subsets is represented by the partial ordering of tensors. Let us first define this property for arbitrary tensors.

Definition 14.2 (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 14.3 *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

$$\epsilon_{\mathcal{U}^0}[O] \prec \epsilon_{\mathcal{U}^1}[O] .$$

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

$$\forall o \in [|\mathcal{U}|]\big(\epsilon_{\mathcal{U}^0}[O = o] = 1\big) \Rightarrow \big(\epsilon_{\mathcal{U}^0}[O = o] = 1\big) .$$

Since subset encodings are boolean tensors, this is equivalent to

$$\epsilon_{\mathcal{U}^0}[O] \prec \epsilon_{\mathcal{U}^1}[O] .$$

■

14.1.1 Binary Relations

Since relations are subsets of cartesian products between two sets, their encoding is a straightforward generalization of Def. 14.1.

Definition 14.4 (Basis encoding of binary relations) A relation between two finite sets $\mathcal{U}^{\text{in}}$ and $\mathcal{U}^{\text{out}}$ is a subset of their cartesian product

$$\mathcal{R} \subset \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{out}} .$$

Given an enumeration of $\mathcal{U}^{\text{in}}$ and $\mathcal{U}^{\text{out}}$ by the categorical variables O_{in} and O_{out} and interpretation maps I_{in}, I_{out}, we define the basis encoding of this subset as the tensor $\epsilon_{\mathcal{R}}[O_{\text{in}}, O_{\text{out}}]$ with the coordinates

$$\epsilon_{\mathcal{R}}[O_{\text{in}} = o_{\text{in}}, O_{\text{out}} = o_{\text{out}}] = \begin{cases} 1 & \text{if} \quad (I_{\text{in}}(o_{\text{in}}), I_{\text{out}}(o_{\text{out}})) \in \mathcal{R} \\ 0 & \text{else} \end{cases} .$$

The basis encoding has a decomposition into one-hot encodings as

$$\epsilon_{\mathcal{R}}[O_{\text{in}}, O_{\text{out}}] = \sum_{o_{\text{in}}, o_{\text{out}} \,:\, (I_{\text{in}}(o_{\text{in}}), I_{\text{out}}(o_{\text{out}})) \in \mathcal{R}} \epsilon_{o_{\text{in}}}[O_{\text{in}}] \otimes \epsilon_{o_{\text{out}}}[O_{\text{out}}] .$$

basis encodings have a matrix structure by the cartesian product, which can be further folded to tensors, when the sets itself are cartesian products. The basis 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 14.5 (Basis encoding of d-ary relations) 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 basis encoding of the relation as the tensor $\epsilon_{\mathcal{R}}[O_{[d]}]$ with coordinates

$$\epsilon_{\mathcal{R}}\left[O_{[d]} = o_{[d]}\right] = \begin{cases} 1 & \text{if} \quad (I_0(o_0), \ldots, I_{d-1}(o_{d-1})) \in \mathcal{R} \\ 0 & \text{else} \end{cases} .$$

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. 4.2 by their basis encoding.

Theorem 14.6 *The encoding of any d-ary relation*

$$\mathcal{R} = \{x^i_{[d]} \,:\, i \in [n]\} \subset \bigtimes_{k\in[d]} \{\text{False}, \text{True}\}$$

where the objects in $\{\text{False}, \text{True}\}$ *are enumerated by* X_k *with the standard index interpretation function (see Sect. 4.1)*

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

coincides with the propositional formula

$$f\left[X_{[d]}\right] = \bigvee_{i\in[n]} Z^{\wedge}_{x^i_{[d]}},$$

Proof. By definition, the encoding $\epsilon_{\mathcal{R}}$ is decomposed as

$$\epsilon_{\mathcal{R}}\left[X_{[d]}\right] = \sum_{i\in[n]} \epsilon_{x^i_{[d]}}\left[X_{[d]}\right] .$$

By Thm. 4.14 this is equal to

$$\tau\left[X_{[d]}\right] = \left(\bigvee_{x_{[d]} \,:\, \tau\left[X_{[d]} = x_{[d]}\right] = 1} Z^{\wedge}_{\{k\,:\,x_k=0\},\{k\,:\,x_k=1\}} \right)[X_{[d]}] = \bigvee_{i\in[n]} Z^{\wedge}_{x^i_{[d]}},$$ ∎

Example 14.7 (Relational Databases) Relational Databases can be encoded as tensors using the relation encoding scheme. Each column is thereby understood as an enumeration variable, which values form the sets $\mathcal{U}^k$. ◊

Let us notice, that the dimensionality of the tensor space used for representing a relation is

$$\prod_{k\in[d]} \left|\mathcal{U}^k\right|$$

and therefore growing exponentially with the number of variables. Relations are however often

sparse, in the sense that

$$|\mathcal{R}| << \prod_{k\in[d]} \left|\mathcal{U}^k\right| .$$

It is therefore often beneficially to choose sparse encoding schemes, for example by restricted CP formats (see Chapter 15) to represent $\epsilon_{\mathcal{R}}$.

14.2 Basis 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 Definition

Definition 14.8 (Basis encoding of maps) Any map

$$q : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$$

can be represented by a relation

$$\mathcal{R}^q := \left\{ \left(x, q(x)\right) \,:\, x \in \mathcal{U}^{\text{in}} \right\} \subset \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{out}} .$$

Given a enumeration of the sets by O_{in} and O_{out} we define the basis encoding of q as the tensor

$$\beta^q \left[O_{\text{out}}, O_{\text{in}}\right] = \epsilon_{\mathcal{R}^q} \left[O_{\text{in}}, O_{\text{out}}\right] .$$

Remark 14.9 (Reduction to images) When q maps into a set of infinite cardinality, we restrict $\mathcal{U}^{\text{out}}$ to the image of q and enumerate the image by a variable O_q. This scheme is applied, when q is itself a tensor, i.e. $\mathcal{U}^{\text{out}} = \mathbb{R}$. While the variable O_q can in general be of the same cardinality as the domain set $\mathcal{U}^{\text{in}}$, it will be valued in $[2]$ when considering boolean tensors. ◊

We notice that any basis 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 basis encoding of functions. This is shown in the next theorem, by the claim that any boolean tensor which is directed is the basis representation of a function.

Theorem 14.10 *Let $\mathcal{U}^{\text{in}}, \mathcal{U}^{\text{out}}$ be sets and $\mathcal{R} \subset \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{out}}$ a relation. If and only if there exists a map $q : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ such that $\mathcal{R} = \mathcal{R}^q$, the basis encoding β^q is a directed tensor with O_{in} incoming and O_{out} outgoing.*

Proof. "⇒": When q is a function, we have for any $o_{\text{in}} \in [r_{\text{in}}]$

$$\sum_{o_{\text{out}}\in[r_{\text{out}}]} \beta^q \left[O_{\text{out}} = o_{\text{out}}, O_{\text{in}} = o_{\text{in}}\right] = \beta^q \left[O_{\text{out}} = I_{\text{out}}^{-1}(q(I_{\text{in}}(o_{\text{in}}))), O_{\text{in}} = o_{\text{in}}\right] = 1 .$$

Thus, $\beta^q \left[O_{\text{out}}, O_{\text{in}}\right]$ is a directed tensor with variables O_{in} incoming and O_{out} outgoing.

"⇐": Conversely let there be a relation $\mathcal{R}$, such that $\beta^{\mathcal{R}}$ is directed. To this end, we observe that for any $o_{\text{in}} \in [r_{\text{in}}]$ the tensor

$$\epsilon_{\mathcal{R}} \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}\right]$$

is a boolean tensor with coordinate sum one and therefore a basis vector. It follows that the function $q : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ defined for $x \in \mathcal{U}^{\text{in}}$ as

$$q(x) = I_{\text{out}}(\epsilon^{-1}(\epsilon_{\mathcal{R}}\left[O_{\text{in}} = I_{\text{in}}^{-1}(x), O_{\text{out}}\right]))$$

is well-defined. We then have by construction

$$\begin{aligned}\beta^q\left[O_{\text{out}}, O_{\text{in}}\right] &= \sum_{o_{\text{in}} \in [r_{\text{in}}]} \epsilon_{q(o_{\text{in}})}\left[O_{\text{out}}\right] \otimes \epsilon_{o_{\text{in}}}\left[O_{\text{in}}\right] \\ &= \sum_{o_{\text{in}} \in [r_{\text{in}}]} \epsilon_{\mathcal{R}}\left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}\right] \otimes \epsilon_{o_{\text{in}}}\left[O_{\text{in}}\right] \\ &= \epsilon_{\mathcal{R}}\left[O_{\text{out}}, O_{\text{in}}\right]\end{aligned}$$

and therefore by Def. 14.8 $\mathcal{R} = \mathcal{R}^q$. ∎

We are specially interested in sets of states of a factored system, which amounts to the case in Def. 0.20. 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. 14.8 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 14.11 (Function evaluation in Basis Calculus) *Retrieving the value of the function q 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*

$$\epsilon_{I_{\text{out}}^{-1}(q(u))}\left[O_{\text{out}}\right] = \left\langle \beta^f\left[O_{\text{out}}, O_{\text{in}}\right], \epsilon_{I_{\text{in}}^{-1}(u)}\left[O_{\text{in}}\right] \right\rangle \left[O_{\text{out}}\right] .$$

Thus, we can retrieve the function evaluation by the inverse one-hot mapping as

$$q(u) = \epsilon^{-1}(\left\langle \beta^f\left[O_{\text{out}}, O_{\text{in}}\right], \epsilon_{I_{\text{in}}^{-1}(u)}\left[O_{\text{in}}\right] \right\rangle \left[O_{\text{out}}\right]) .$$

Proof. From the representation

$$\beta^q\left[O_{\text{out}}, O_{\text{in}}\right] = \sum_{o_{\text{in}} \in [r_{\text{in}}]} \epsilon_{(I_{\text{out}}^{-1} \circ q \circ I_{\text{in}}) o_{\text{in}}}\left[O_{\text{in}}\right] \otimes \epsilon_{o_{\text{in}}}\left[O_{\text{in}}\right]$$

and the orthonormality of the one-hot encodings of the input enumeration we get

$$\left\langle \beta^f\left[O_{\text{out}}, O_{\text{in}}\right], \epsilon_{I_{\text{in}}^{-1}(u)}\left[O_{\text{in}}\right] \right\rangle \left[O_{\text{out}}\right] = \epsilon_{I_{\text{out}}^{-1}(q(u))}\left[O_{\text{out}}\right] .$$ ∎

In comparison with the Coordinate Calculus scheme (see Thm. 13.3), the Basis Calculus produces basis vectors of a functions evaluation instead of scalars. While this seems to produce unnecessary redundancy in representing a function, we will see in the following section, that this scheme is efficient in representing compositions of functions.

14.3 Decomposition of Basis Encodings

We now show the utility of basis 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 Def. 4.7), that combination of propositional formulas by connectives can be represented by contractions. We now show in a more general perspective, that in basis calculus, any composition of functions in its basis encoding the contraction of the encoded functions.

Theorem 14.12 (Composition of Functions) *Let there be two maps between factored systems*

$$q : \bigtimes_{v\in\mathcal{V}^1} [m_v] \to \bigtimes_{v\in\mathcal{V}^2} [m_v]$$

and

$$g : \bigtimes_{v\in\mathcal{V}^2} [m_v] \to \bigtimes_{v\in\mathcal{V}^3} [m_v]$$

with the image system of q is the domain system of g. Then the basis encoding of the composition

$$g \circ q : \bigtimes_{v\in\mathcal{V}^1} [m_v] \to \bigtimes_{v\in\mathcal{V}^3} [m_v]$$

is the contraction

$$\beta^{g\circ q}\left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] = \left\langle \beta^g\left[X_{\mathcal{V}^3}, X_{\mathcal{V}^2}\right], \beta^q\left[X_{\mathcal{V}^2}, X_{\mathcal{V}^1}\right], \right\rangle \left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] .$$

Proof. By definition we have the basis encoding of the composition as

$$\beta^{g\circ q}\left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] = \sum_{x_{\mathcal{V}^1}\in\times_{v\in\mathcal{V}^1}[m_v]} \epsilon_{(g\circ q)(x_{\mathcal{V}^1})}\left[X_{\mathcal{V}^3}\right] \otimes \epsilon_{x_{\mathcal{V}^1}}\left[X_{\mathcal{V}^1}\right] .$$

By using a similar representation for β^g and β^q we now show, that this coincides with the contraction of these basis encodings with closed variables $X_{\mathcal{V}^2}$. By the linearity of the contraction operation we get

$$\begin{aligned}
\left\langle \beta^q, \beta^g \right\rangle \left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] &= \sum_{x_{\mathcal{V}^1}\in\times_{v\in\mathcal{V}^1}[m_v]} \sum_{x_{\mathcal{V}^2}\in\times_{v\in\mathcal{V}^2}[m_v]} \left\langle \left(\epsilon_{g(x_{\mathcal{V}^2})}\left[X_{\mathcal{V}^3}\right] \otimes \epsilon_{x_{\mathcal{V}^2}}\left[X_{\mathcal{V}^2}\right]\right), \right. \\
&\qquad \left. \left(\epsilon_{q(x_{\mathcal{V}^1})}\left[X_{\mathcal{V}^2}\right] \otimes \epsilon_{x_{\mathcal{V}^1}}\left[X_{\mathcal{V}^1}\right]\right) \right\rangle \left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] \\
&= \sum_{x_{\mathcal{V}^1}\in\times_{v\in\mathcal{V}^1}[m_v]} \delta_{x_{\mathcal{V}^2}, x_{\mathcal{V}^1}} \cdot \epsilon_{g(x_{\mathcal{V}^2})}\left[X_{\mathcal{V}^3}\right] \otimes \epsilon_{x_{\mathcal{V}^1}}\left[X_{\mathcal{V}^1}\right] \\
&= \sum_{x_{\mathcal{V}^1}\in\times_{v\in\mathcal{V}^1}[m_v]} \epsilon_{(g\circ q)(x_{\mathcal{V}^1})}\left[X_{\mathcal{V}^3}\right] \otimes \epsilon_{x_{\mathcal{V}^1}}\left[X_{\mathcal{V}^1}\right] \\
&= \beta^{g\circ q}\left[X_{\mathcal{V}^3}, X_{\mathcal{V}^1}\right] ,
\end{aligned}$$

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. ∎

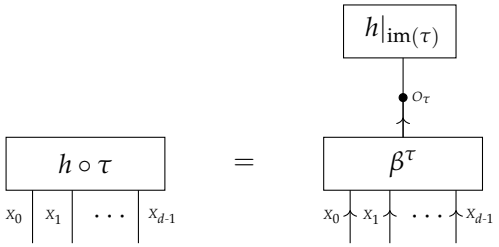


Figure 14.1: Representation of the composition of a tensor τ with a real function h.

We can use Thm. 14.12 iteratively to further decompose the function g. In this way, the basis encoding of a function consistent of multiple compositions can be represented as the contractions of all the functions. This has been applied in syntactical compositions (see Def. 4.7) 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

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

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 $\tau\left[X_{[d]}\right]$ as a map of the states $\bigtimes_{k\in[d]}[m_k]$ onto its by a variable O_τ and index interpretation I enumerated image $\mathrm{im}\,(\tau)$. We then define the restriction of h onto $\mathrm{im}\,(\tau)$ as the tensor $h|_{\mathrm{im}(\tau)}\,[O_\tau]$ with coordinates o_τ

$$h|_{\mathrm{im}(\tau)}\,[O_\tau = o_\tau] = (h \circ I)(o_\tau)\,.$$

Let us now show, how contractions with these vectors represents compositions with tensors.

Theorem 14.13 *The coordinatewise transform of any tensor τ (see Def. 13.5) by a real function h is the contraction (see Figure 14.1)*

$$h(\tau)[X_{[d]}] = \left\langle \beta^\tau \left[O_\tau, X_{[d]}\right], h|_{\mathrm{im}(\tau)}\,[O_\tau]\right\rangle \left[X_{[d]}\right]\,.$$

Proof. By the basis calculus Thm. 14.11 we have for any state $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$, that

$$\begin{aligned}\left\langle \beta^\tau \left[O_\tau, X_{[d]}\right], h|_{\mathrm{im}(\tau)}\,[O_\tau]\right\rangle \left[X_{[d]} = x_{[d]}\right] &= \left\langle \beta^\tau \left[O_\tau, X_{[d]} = x_{[d]}\right], h|_{\mathrm{im}(\tau)}\,[O_\tau]\right\rangle [\varnothing] \\ &= \left\langle \epsilon_{I_{\tau\left[X_{[d]}=x_{[d]}\right]}}\,[O_\tau], h|_{\mathrm{im}(\tau)}\,[O_\tau]\right\rangle [\varnothing] \\ &= h(\tau)[X_{[d]} = x_{[d]}]\,.\end{aligned}$$

Since both tensors coincide on all coordinates, they are equal. ∎

Corollary 14.14 *For any tensor $\tau\left[X_{[d]}\right]$ we have*

$$\tau\left[X_{[d]}\right] = \left\langle \beta^{\tau}\left[O_{\tau}, X_{[d]}\right], \mathrm{Id}|_{\mathrm{im}(\tau)}[O_{\tau}]\right\rangle \left[X_{[d]}\right].$$

Proof. This follows from Thm. 14.13 using $h = \mathrm{Id}$ and by noticing that

$$\tau\left[X_{[d]}\right] = \mathrm{Id}(\tau)[X_{[d]}]. \qquad \blacksquare$$

Remark 14.15 (Transform of basis into coordinate encodings) Cor. 14.14 states in particular the transformation of the basis encoding of a function into its coordinate encoding. Given a real function $q : \bigtimes_{k\in[d]}[m_k] \to \mathbb{R}$ we have

$$\chi^{q}\left[X_{[d]}\right] = \left\langle \beta^{q}\left[Y_q, X_{[d]}\right], I_q(Y_q)\right\rangle \left[X_{[d]}\right],$$

where Y_q is a variable, which enumerates the image of q with the interpretation $I_q(Y_q)$. $\lozenge$

Corollary 14.16 *For any tensor τ, which is directed with $X_{[d]}$ incoming, we have*

$$\mathbb{I}\left[X_{[d]}\right] = \langle \beta^{\tau}\rangle \left[X_{[d]}\right].$$

Proof. This follows from Thm. 14.13 using $h = \mathbb{I}$ and by noticing that

$$\mathbb{I}\left[X_{[d]}\right] = \mathbb{I}(\tau)[X_{[d]}]. \qquad \blacksquare$$

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 basis encoding can be represented by a contraction of basis encodings to each image coordinate.

Theorem 14.17 *Let q be a function between factored systems*

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

and denote by

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

the image coordinate restrictions of q, that is we have $q = (q_0, \ldots, q_{d-1})$. Let us assign the variable X to the factored system in the domain system of q and the variables X_k for $k \in [d]$ to the image system of q. We can then decompose the basis encoding of q into the basis encodings of its image coordinate

restrictions, that is

$$\beta^q \left[X_{[d]}, X\right] = \langle \{\beta^{q_k}\left[X_k, X\right] : k \in [d]\}\rangle \left[X_{[d]}, X\right] .$$

Proof. For any $x \in [m]$ we have

$$\begin{aligned}
\beta^q \left[X_{[d]}, X = x\right] &= \epsilon_{q(x)} \left[X_{[d]}\right] \\
&= \bigotimes_{k\in[d]} \beta^{q_k}\left[X_k, X = x\right] \\
&= \langle \{\beta^{q_k}\left[X_k, X = x\right] : k \in [d]\}\rangle \left[X_{[d]}\right] \\
&= \langle \{\beta^{q_k}\left[X_k, X\right] : k \in [d]\}\rangle \left[X_{[d]}, X = x\right]
\end{aligned}$$

and therefore equality of both tensors. ∎

In Chapter 15 we will apply Thm. 14.17 in Thm. 15.20 to show sparse basis CP decompositions to β^q. These decompositions are then applied for efficient representation of the empirical distribution, which involve the basis encoding of data maps (see Example 15.21), and for exponential families, which statistics have images, which are included in cartesian products of the images to each coordinate (see Example 15.22).

14.4 Selection Encodings

Selection encodings as introduced in Def. 0.22 are best understood in terms of linear mapping interpretations of tensors. We will first provide by basis representations 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 Representations of Linear Maps

Basis representations are standard linear algebra tools, where matrices are understood as linear maps between vector spaces. The state sets $\bigtimes_{k\in[d]}[m_k]$ can be interpreted as an enumeration of basis elements ϵ_x of the tensor space $\bigotimes_{k\in[d]}\mathbb{R}^{m_k}$. Along this interpretation, tensors have an interpretation as maps between tensor spaces. Any tensor and any partition of its variables into two sets can be interpreted as the basis elements of a linear map between the tensor spaces of the respective variables. Tensor valued functions on state sets $\bigtimes_{k\in[d]}[m_k]$ are an intermediate representation.

Definition 14.18 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 representation of any linear map $F \in \mathbb{L}(V_1, V_2)$ is then the tensor

$$\tau^q\left[O_1, O_2\right] \in \mathbb{R}^{r_1} \otimes \mathbb{R}^{r_2}$$

defined for $o_1 \in [r_1]$ and $o_2 \in [r_2]$ by

$$\tau^F\left[O_1 = o_1, O_2 = o_2\right] = \left\langle F^{I_1(o_1)}, I_2(o_2)\right\rangle [\varnothing] .$$

Basis representations for compositions of linear functions can be computed via contractions of the respective basis representations, as we show next.

Theorem 14.19 *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*

$$\tau^{F^2 \circ F^1}[O_1, O_3] = \left\langle \tau^{F^2}[O_2, O_3], \tau^{F^1}[O_1, O_2] \right\rangle [O_1, O_3] .$$

Proof. For arbitrary $o_1 \in [r_1]$ and $o_3 \in [r_3]$ we have to show that

$$\tau^{F^2 \circ F^1}[O_1 = o_1, O_3 = o_3] = \left\langle \tau^{F^2}[O_2, O_3 = o_3], \tau^{F^1}[O_1 = o_1, O_2] \right\rangle [\varnothing] .$$

By definition we have

$$\tau^{F^2 \circ F^1}[O_1 = o_1, O_3 = o_3] = \left\langle F^2 \circ F^1(I_1(o_1)), I_3(o_3) \right\rangle [\varnothing] .$$

Decomposing the linear maps using their basis representation we get

$$\begin{aligned}
\left\langle F^2 \circ F^1(I_1(o_1)), I_3(o_3) \right\rangle [\varnothing] &= \left\langle F^2 \left(\sum_{o_2 \in [r_2]} \tau^{F^1}[O_1 = o_1, O_2 = o_2] \cdot I_2(o_2) \right), I_3(o_3) \right\rangle [\varnothing] \\
&= \sum_{o_2 \in [r_2]} \left\langle F^2 \left(\tau^{F^1}[O_1 = o_1, O_2 = o_2] \cdot I_2(o_2) \right), I_3(o_3) \right\rangle [\varnothing] \\
&= \sum_{o_2 \in [r_2]} \left\langle \tau^{F^1}[O_1 = o_1, O_2 = o_2] \cdot \tau^{F^2}[O_2 = o_2, O_1 = o_1] \right\rangle [\varnothing] \\
&= \left\langle \tau^{F^2}[O_2, O_3 = o_3], \tau^{F^1}[O_1 = o_1, O_2] \right\rangle [\varnothing] .
\end{aligned}$$

Therefore, both tensors are equivalent. ■

For basis representations we thus have a similar composition theorem as for basis encodings of arbitrary functions (see Thm. 14.12). What is more, one can understand each basis encodings as a basis representation of a linear function. Along this line, the composition theorem Thm. 14.19 as the principle of linear algebra, which underlies Thm. 14.12. A typical interpretation of Thm. 14.19 is matrix multiplication, where matrices understood since matrices are basis representations of linear maps.

Example 14.20 (Basis encodings as basis representations) Let us justify that we referred to the contractions of basis encodings by basis calculus, by describing that basis encodings are a special case of basis representations. 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

$$F(\epsilon_i) = \sum_{j \in [r_{\text{out}}] \,:\, (i,j) \in \mathcal{R}} \epsilon_j .$$

Comparing the coefficients of the basis representation of F and the basis encoding $\mathcal{R}$ we get

$$\tau^F[O_{\text{out}} = o_{\text{out}}, O_{\text{in}} = o_{\text{in}}] = \epsilon_{\mathcal{R}}[O_{\text{out}} = o_{\text{out}}, O_{\text{in}} = o_{\text{in}}] .$$

◊

14.4.2 Selection Encodings as Basis Representations

Selection encodings (see Def. 0.22) are related to basis representations of linear maps as we show in the next theorem.

Theorem 14.21 *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 ϵ. Given a function*

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

we define a linear map $F^q \in \mathbb{L}(\bigotimes_{k\in[d]}\mathbb{R}^{m_k}, \bigotimes_{s\in[n]}\mathbb{R}^{p_s})$ by the action on the basis elements to $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ as the tensors

$$F^q(\epsilon_{x_{[d]}}) \coloneqq q(x_{[d]})$$

carrying the variables $L_{[n]}$. We then have

$$\sigma^q\left[X_{[d]}, L_{[n]}\right] = \tau^{F^q}\left[X_{[d]}, L_{[n]}\right] .$$

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^q that

$$\sigma^q\left[X_{[d]} = x_{[d]}, L_0, \ldots, L_{n-1}\right] = q(x_{[d]})[L_{[n]}] = F^q(\epsilon_{x_{[d]}})[L_{[n]}] .$$

We further have

$$\begin{aligned} F^q(\epsilon_{x_{[d]}})[L_{[n]}] &= \sum_{l_{[n]}} \left\langle F^q(\epsilon_{x_{[d]}})[L_{[n]} = l_{[n]}], \epsilon_{l_{[n]}}\left[L_{[n]}\right]\right\rangle [\varnothing] \cdot \epsilon_{l_{[n]}}\left[L_{[n]}\right] \\ &= \sum_{l_{[n]}} \tau^{F^q}\left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]}\right] \cdot \epsilon_{l_{[n]}}\left[L_{[n]}\right] \\ &= \tau^{F^q}\left[X_{[d]} = x_{[d]}, L_{[n]}\right] . \end{aligned}$$

For arbitrary $x_{[d]}$ the slices of σ^q and τ^{F^q} thus coincide, which proves the equivalence of both tensors. ∎

While basis encoding works for maps from $\bigtimes_{k\in[d]}[m_k]$ to arbitrary sets (which are enumerated), selection encodings as introduced in Def. 0.22 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

$$q(x) = \sigma^q\left[X = x, L\right] .$$

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 14.22 *Let $\tau\left[X_{\mathcal{V}}\right]$ 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

$$q : \bigtimes_{v \in A} [m_v] \to \bigotimes_{v \in B} \mathbb{R}^{m_v}$$

defined for $x_A \in \bigtimes_{v \in A} [m_v]$ as

$$q(x_A) \coloneqq \tau\left[X_A = x_A, X_B\right]$$

obeys

$$\sigma^q\left[X_A, X_B\right] = \tau\left[X_{\mathcal{V}}\right],$$

where we understand the variables X_B as selection variables.

Proof. We have for any x_B that

$$\begin{aligned}
\sigma^q\left[X_A, X_B = x_B\right] &= \sum_{x_A \in \bigtimes_{v \in A}[m_v]} \epsilon_{x_A}\left[X_A\right] \otimes q(x_A)[X_B = x_B] \\
&= \sum_{x_A \in \bigtimes_{v \in A}[m_v]} \epsilon_{x_A}\left[X_A\right] \otimes \tau\left[X_A = x_A, X_B = x_B\right] \\
&= \tau\left[X_A, X_B = x_B\right]
\end{aligned}$$

and the equivalence follows. ∎

Example 14.23 (Markov Logic Networks and Proposal Distributions) While the statistic of MLN (namely $\mathcal{F}$) and the proposal distribution (namely $\mathcal{F}^T$) have a common selection encoding, both result from the inverse selection encoding described in Thm. 14.22. We can construct $\mathcal{F}^T$ by first building the selection encoding to $\mathcal{F}$ and then applying the construction of Thm. 14.22 with $A = L$ and $B = X_{[d]}$. ◊

We use selection encodings to represent weighted sums of functions, based on the next theorem.

Theorem 14.24 (Weighted formula sums by selection encodings) *Let $\mathcal{S}$ be a tensor valued function from $\bigtimes_{k \in [d]} [m_k]$ to $\mathbb{R}^p$ with image coordinates s_l and let $\theta\left[L\right]$ be a tensor. Then*

$$\left(\sum_{l \in [p]} \theta\left[L = l\right] \cdot s_l\right)\left[X_{[d]}\right] : \bigtimes_{k \in [d]} [m_k] \to \mathbb{R}$$

is represented as

$$\left(\sum_{l \in [p]} \theta\left[L = l\right] \cdot s_l\right)\left[X_{[d]}\right] = \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \theta\left[L\right] \right\rangle \left[X_{[d]}\right].$$

Proof. The representation holds, since for any $x_{[d]} \in \bigtimes_{k \in [d]} [m_k]$ we have

$$\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \theta\left[L\right] \right\rangle \left[X_{[d]} = x_{[d]}\right] = \sum_{l \in [p]} q(L = l) \cdot \mathcal{S}_l\left[X_{[d]} = x_{[d]}\right].$$ ∎

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 $\mathcal{S}$ and q in Thm. 14.24 as basis decompositions of linear maps.

14.5 Indicator Features to Functions

We here provide a subspace perspective for the sparse representation of decomposable functions as tensor networks in the $\beta^{\cdot}$ basis encoding scheme of basis calculus.

Definition 14.25 Given any function $q : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ and index interpretation functions $I_{\text{in}}, I_{\text{out}}$ enumerating $\mathcal{U}^{\text{in}}$ and $\mathcal{U}^{\text{out}}$ we define the corresponding indicator subspace of $\mathbb{R}^{|\mathcal{U}^{\text{in}}|}$ as

$$V^q = \{\langle \beta^q [O_{\text{out}}, O_{\text{in}}], \xi [O_{\text{out}}] \rangle [O_{\text{out}}, O_{\text{in}}] : \xi [O_{\text{out}}] \in \mathbb{R}^{r_{\text{out}}}\} .$$

For any function q we have $\mathbb{I}\left[X_{[d]}\right] \in V^q$, when choosing $\xi [O_{\text{out}}] = \mathbb{I} [O_{\text{out}}]$.

We now characterize the indicator subspace as a span of boolean indicator features, indicating whether the function value coincides with an element $y \in \text{im}(q)$. Each indicator feature is a tensor $\mathbb{I}_{q=y} [O_{\text{in}}]$ with coordinates

$$\mathbb{I}_{q=y} [O_{\text{in}} = o_{\text{in}}] = \begin{cases} 1 & \text{if} \quad q(I_{\text{in}}(o_{\text{in}})) = y \\ 0 & \text{else} \end{cases} .$$

Lemma 14.26 *A value subspace is spanned by the boolean indicator features of the underlying function, that is*

$$V^q = \text{span}\left(\mathbb{I}_{q=y} [O_{\text{in}}] : y \in \text{im}(q)\right) .$$

Proof. By linearity of contractions we have

$$V^q = \text{span}\left(\langle \beta^q [O_{\text{out}}, O_{\text{in}}], \epsilon_{o_{\text{out}}} [O_{\text{out}}] \rangle [O_{\text{in}}] : o_{\text{out}} \in [r_{\text{out}}]\right) .$$

It further holds that

$$\mathbb{I}_{q=I_{o_{\text{out}}}} [O_{\text{in}}] = \langle \beta^q [O_{\text{out}}, O_{\text{in}}], \epsilon_{o_{\text{out}}} [O_{\text{out}}] \rangle [O_{\text{in}}] .$$

The claim follows as a combination of both equations. ∎

14.5.1 Connections with Computable Families

We now apply indicator spaces to provide further intuition into computable families (see Def. 2.20).

Lemma 14.27 *For any statistic $\mathcal{S} : \bigtimes_{k \in [d]} [m_k] \to \mathbb{R}^p$ we have*

$$\Lambda^{\mathcal{S},\text{MAX}} = V^{\mathcal{S}} \cup \mathbb{S}$$

Proof. For any non-negative $\mathbb{P}\left[X_{[d]}\right]$ we have $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{S},\text{MAX}}$ if and only if $\left\langle \mathbb{P}\left[X_{[d]}\right] \right\rangle [\varnothing] = 1$ and there is an activation core $\xi\left[Y_{[p]}\right]$ with

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \beta^{\mathcal{S}}\left[Y_{[p]}\right] \right\rangle \left[X_{[d]}\right] .$$

This is equivalent to $\mathbb{P}\left[X_{[d]}\right] \in V^{\mathcal{S}} \cup \mathbb{S}$. ∎

In the other extreme of tensor network formats for the activation tensor, we have the elementary graph EL. To provide a characterization of $\Lambda^{\mathcal{S},\mathrm{EL}}$, we understand any statistic as a cartesian product of its features s_l. For cartesian products we can show that the distributions realizable with elementary activation tensors coincide with the subspace contraction of the spaces V^{s_l} to be introduced next.

Definition 14.28 Given two subspaces V^1, V^2 of tensors with variables $X_{\mathcal{V}^1}$, $X_{\mathcal{V}^2}$, their contraction is

$$\left\langle V^1, V^2 \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\{ \left\langle \tau^1 [X_{\mathcal{V}^1}], \tau^2 [X_{\mathcal{V}^2}] \right\rangle [X_{\tilde{\mathcal{V}}}] \,:\, \tau^1 [X_{\mathcal{V}^1}] \in V^1,\, \tau^2 [X_{\mathcal{V}^2}] \in V^2 \right\}.$$

We notice that the contraction of two subspaces is in general not a subspace. This fact will in the following become clearer, where we characterize contractions of subspaces with elementarily computable distributions, which are known to not be linear subspaces.

The cartesian product of functions on the same input set $\mathcal{U}^{\mathrm{in}}$ and output sets $\mathcal{U}^{1,\mathrm{out}}, \mathcal{U}^{2,\mathrm{out}}$ is the function

$$(q, g) \,:\, \mathcal{U}^{\mathrm{in}} \to \mathcal{U}^{1,\mathrm{out}} \times \mathcal{U}^{2,\mathrm{out}}$$

with

$$(q, g)(z) = (q(z), g(z)).$$

Its indicator subspace is the contraction of indicator subspaces

$$\begin{aligned} \langle V^q, V^g \rangle [O_{\mathrm{in}}] &= \left\langle \mathrm{span}\left(\{\mathbb{I}_{q=y} [O_{\mathrm{in}}] \,:\, y \in \mathrm{im}\,(q)\}\right), \mathrm{span}\left(\{\mathbb{I}_{g=y} [O_{\mathrm{in}}] \,:\, y \in \mathrm{im}\,(g)\}\right) \right\rangle [O_{\mathrm{in}}] \\ &= \Lambda^{\{q,g\},\mathrm{EL}}. \end{aligned}$$

We generalize this in the next lemma to cartesian products of multiple features using that any statistic $\mathcal{S}$ is the cartesian product of its features s_l.

Lemma 14.29 *For any statistic $\mathcal{S} : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R}^p$ we have*

$$\Lambda^{\mathcal{S},\mathrm{EL}} = \langle V^{s_l} \,:\, l \in [p] \rangle \left[X_{[d]} \right] \cup \mathbb{S}.$$

where $\mathbb{S}$ is the sphere of normalized tensors in $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$.

Proof. For any non-negative tensor $\mathbb{P}\left[X_{[d]}\right]$ we have $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{S},\mathrm{EL}}$ if and only if

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

and there exist activation cores $\xi^l [Y_l]$ such that

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \bigcup_{l\in[p]} \{\beta^{s_l}\left[Y_l, X_{[d]}\right], \xi^l [Y_l]\} \right\rangle \left[X_{[d]}\right].$$

Since for any $l \in [p]$ we have

$$\left\{ \left\langle \beta^{s_l}\left[Y_l, X_{[d]}\right], \xi^l [Y_l] \right\rangle \left[X_{[d]}\right] \,:\, \xi^l [Y_l] \in \mathbb{R}^{n_l} \right\} = V^{s_l}$$

$\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\mathcal{S},\mathrm{EL}}$ is equal to

$$\mathbb{P}\left[X_{[d]}\right] \in \langle V^{s_l} \,:\, l \in [p]\rangle \left[X_{[d]}\right] . \qquad \blacksquare$$

Since always $\mathbb{I}\left[X_{[d]}\right] \in V^q$, we have for any subspace V^1

$$\left\langle V^1 \right\rangle [X_{\tilde{\mathcal{V}}}] \subset \left\langle V^1, V^q \right\rangle [X_{\tilde{\mathcal{V}}}] .$$

We can therefore understand the addition of a feature to a set of feature as a monotonously increasing set of elementarily computable distributions, that is

$$\Lambda^{\mathcal{S},\mathrm{EL}} \subset \Lambda^{\mathcal{S}\cup\{q\},\mathrm{EL}} .$$

We now relate the by a function q computable distributions with those, which are computable by its indicator features with elementary activation cores.

Lemma 14.30 *For any function $q : \bigtimes_{k\in[d]} [m_k] \to \mathcal{U}^{\mathrm{out}}$ we have*

$$\Lambda^{q,\mathrm{MAX}} = \Lambda^{\{\mathbb{I}_{q=y} \,:\, y\in \mathrm{im}(q)\},\mathrm{EL}}$$

Proof. "$\subset$": For any $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{q,\mathrm{MAX}}$ we find an activation core $\xi\,[Y]$ such that

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \beta^q \left[Y, X_{[d]}\right], \xi\,[Y] \right\rangle \left[X_{[d]}\right] .$$

Given this activation tensor $\xi\,[Y]$ we construct an elementary activation tensor

$$\bigotimes_{l\in[p]} \xi^l\,[Y_l]$$

which reproduces $\mathbb{P}\left[X_{[d]}\right]$ by contraction with the basis encoding of the indicator features to q. To this end, we define for $l \in [p]$ two-dimensional leg vectors

$$\xi^l\,[Y_l] = \begin{bmatrix} \xi\,[Y = l] \\ 1 \end{bmatrix} [Y_l] .$$

For these head variables we have for any index tuple $x_{[d]}$

$$\begin{aligned}
\left\langle \bigcup_{l\in[p]} \{\beta^{\mathbb{I}_{q=l_l}}\,[Y_l], \xi\,[Y = l]\} \right\rangle \left[X_{[d]} = x_{[d]}\right] &= \xi^{q(x_{[d]})}\,[Y_l = 1] \cdot \prod_{l\in[p]\,:\, q(x_{[d]})\neq l} \xi^{q(x_{[d]})}\,[Y_l = 0] \\
&= \xi^{q(x_{[d]})}\,[Y_l = 1] \\
&= \left\langle \beta^q \left[Y, X_{[d]}\right], \xi\,[Y] \right\rangle \left[X_{[d]} = x_{[d]}\right] \\
&= \mathbb{P}\left[X_{[d]} = x_{[d]}\right] .
\end{aligned}$$

"$\supset$": Conversely, given $\mathbb{P}\left[X_{[d]}\right] \in \Lambda^{\{\mathbb{I}_{q=y} \,:\, y\in \mathrm{im}(q)\},\mathrm{EL}}$ we find an elementary activation tensor

$$\bigotimes_{l\in[p]} \xi^l\,[Y_l]$$

such that

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \bigcup_{l\in[p]} \{\beta^{\mathbb{I}_{q=I_l}}[Y_l], \xi[Y=l]\} \right\rangle \left[X_{[d]}\right]$$

If $\xi^l[Y_l = 0] = 0$ for an l, then we have $\mathbb{P}\left[X_{[d]}\right] = \left\langle \mathbb{I}_{q=I_l}\left[X_{[d]}\right]\right\rangle \left[X_{[d]}|\varnothing\right]$, which is an element of $\Lambda^{q,\mathrm{MAX}}$ since then

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \beta^q\left[Y, X_{[d]}\right], \epsilon_l[Y]\right\rangle \left[X_{[d]}|\varnothing\right].$$

In all other cases we multiply scalars to the leg tensors such that $\xi^l[Y_l = 0] = 1$. Notice, that the product of all such scalars needs to be 1 since $\left\langle \mathbb{P}\left[X_{[d]}\right]\right\rangle[\varnothing] = 1$. We then construct an activation core $\xi[Y]$

$$\xi[Y = l] = \xi^l[Y_l = 1]$$

and have with a similar argument as in the converse proof direction

$$\mathbb{P}\left[X_{[d]}\right] = \left\langle \beta^q\left[Y, X_{[d]}\right], \xi[Y]\right\rangle \left[X_{[d]}\right]. \qquad \blacksquare$$

14.5.2 Composition of Functions

Let $q : \mathcal{U}^1 \to \mathcal{U}^2$ and $h : \mathcal{U}^2 \to \mathcal{U}^3$ be arbitrary functions, then the indicator of their composition obeys

$$V^{h\circ q} \subset V^q.$$

Example 14.31 (Propositional Formulas) Each formula defines by its basis 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

$$V^{\circ(f,h)} \subset V^{(f,h)}.$$

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

$$\bigotimes_{k\in[d]} \mathbb{R}^2 = \mathrm{span}\left(\left\langle V^{f^k} : k \in [d]\right\rangle \left[X_{[d]}\right]\right).$$

$\Diamond$

14.5.3 Effective Representation of Partition Statistics

Definition 14.32 We call a statistic $\mathcal{S} : \bigtimes_{k\in[d]}[m_k] \to \bigtimes_{l\in[p]}[2]$ a partition statistic if

$$\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle \left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right].$$

We now show, that partition statistics are exactly those statistics, which are collections of indicator features to a function.

Lemma 14.33 *A statistic $\mathcal{S}$ is a partition statistic, if and only if there exists a function $q : \bigtimes_{k\in[d]}[m_k] \to \mathcal{U}^{\text{out}}$ with an image interpretation $I : [p] \to \mathcal{U}^{\text{out}}$ such that for all $l \in [p]$*

$$\sigma^{\mathcal{S}}\left[X_{[d]}, L = l\right] = \mathbb{I}_{q=I(l)}\left[X_{[d]}\right] .$$

Proof. "⇐": Given any function $q : \bigtimes_{k\in[d]}[m_k] \to \mathcal{U}^{\text{out}}$ we have

$$\sum_{l\in[p]} \mathbb{I}_{q=I(l)}\left[X_{[d]}\right] = 1$$

and the selection tensor of indicator features is thus a partition statistic.

"⇒": Conversely, given a partition statistic $\mathcal{S}$, we can construct a function $q_{\mathcal{S}}$ such that the partition statistic coincides with the collection of indicator features to the function. To this end we notice that for any index tuple $x_{[d]}$ there is a unique $l \in [p]$ such that

$$\sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L = l\right] = 0 .$$

This follows from $\sigma^{\mathcal{S}}\left[X_{[d]}, L\right]$ being boolean by assumption and $\left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right]\right\rangle \left[X_{[d]} = x_{[d]}\right] = 1$. We define a function $q_{\mathcal{S}} : \bigtimes_{k\in[d]}[m_k] \to [p]$ with image interpretation by the identity on $[p]$ coordinatewise by

$$q_{\mathcal{S}}(X_{[d]} = x_{[d]}) = l$$

and have for each $l \in [p]$

$$\sigma^{\mathcal{S}}\left[X_{[d]}, L = l\right] = \mathbb{I}_{q_{\mathcal{S}}=l}\left[X_{[d]}\right]$$ ∎

Example 14.34 (Edge statistics of Markov Networks) Each edge statistics s_e of a Markov Network (see Thm. 2.50) defines a partition statistic

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

This holds since for any $x_{\mathcal{V}}$ we have

$$\begin{aligned}\langle \sigma^{s_e}\left[X_{\mathcal{V}}, L_e\right]\rangle \left[X_{\mathcal{V}} = x_{\mathcal{V}}\right] &= \sum_{l_e\in[m_e]} \sigma^{s_e}\left[X_{\mathcal{V}} = x_{\mathcal{V}}, L_e = l_e\right] \\ &= \sigma^{s_e}\left[X_{\mathcal{V}} = x_{\mathcal{V}}, L_e = X_e\right] \\ &= 1 .\end{aligned}$$

The corresponding indicators stated in Lem. 14.33 are enumerated by the image elements $l_e \in m_e$ and given by

$$\mathcal{S}_{e,l_e}\left[x_{\mathcal{V}}\right] = \begin{cases} 1 & \text{if} \quad x_e = l_e \\ 0 & \text{else} \end{cases} .$$

◊

We have already observed in Chapter 2, that Markov Networks have a tensor-network representation involving the selection encodings of their edge statistics. This efficiency gain compared with fea-

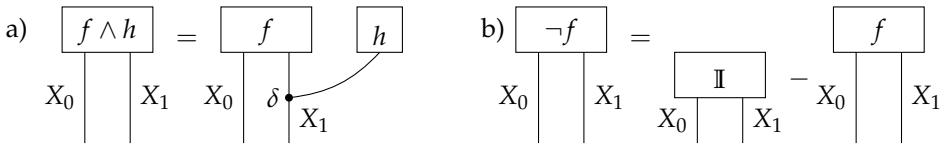


Figure 14.2: Decomposition schemes by hybrid calculus, using coordinatewise transforms of tensors (see Def. 13.5). a) Conjunction performed by coordinatewise multiplications. b) Negations performed by coordinatewise subtraction from one.

turewise releational encodings is in the following generalized to arbitrary partition statistics.

Theorem 14.35 *For any partition statistic and any elementary activation tensor* $\bigotimes_{l\in[p]} \xi^l\left[Y_l\right]$ *we have*

$$\left\langle \bigcup_{l\in[p]} \left\{\beta^{s_l}\left[Y_l, X_{[d]}\right], \xi^l\left[Y_l\right]\right\}\right\rangle \left[X_{[d]}\right] = \left\langle \sigma^{\mathcal{S}}\left[X_{[d]}, L\right], \xi\left[Y\right]\right\rangle \left[X_{[d]}\right]$$

where

$$\xi\left[Y\right] = \begin{cases} \left(\prod_{l\in[p]} \xi^l\left[Y_l=0\right]\right) \sum_{l\in[p]} \xi^l\left[Y_l=1\right]\cdot \epsilon_l\left[Y\right] & \text{if} \quad \forall l\in[p]: \xi^l\left[Y_l=0\right]\neq 0 \\ \xi^{\tilde{l}}\left[Y_{\tilde{l}}=1\right]\cdot \prod_{l\neq\tilde{l}} \xi^{\tilde{l}}\left[Y_{\tilde{l}}=0\right] & \text{if} \quad \exists \tilde{l}\in[p]: \xi^{\tilde{l}}\left[Y_{\tilde{l}}=0\right]=0 \end{cases}.$$

We notice that by definition $\xi\left[Y\right] = 0\left[Y\right]$ *if there is more than one* $l\in[p]$ *with* $\xi^l\left[Y_l=0\right]=0$.

Proof. Lem. 14.33 implies, that there is a function q such that the features of the partition statistics are the indicator features to q. Now, as in the proof of Lem. 14.30 we transform the activation cores to arrive at the statement. ∎

Thm. 14.35 is especially useful, when instantiating the distribution of a Hybrid Logic Network with a subset $\tilde{\mathcal{F}}\subset\mathcal{F}$ of features satisfying

$$\sum_{f\in\tilde{\mathcal{F}}} f\left[X_{[d]}\right] \prec \mathbb{I}\left[X_{[d]}\right].$$

If $\sum_{f\in\tilde{\mathcal{F}}} f\left[X_{[d]}\right] \neq \mathbb{I}\left[X_{[d]}\right]$ we can add a dummy feature $\mathbb{I}\left[X_{[d]}\right] - \sum_{f\in\mathcal{F}} f\left[X_{[d]}\right]$ to $\tilde{\mathcal{F}}$ with trivial activation core to get a partition statistics. Now, given a partition statistic $\tilde{\mathcal{F}}$ we can instantiate the to $\tilde{\mathcal{F}}$ corresponding tensors in the tensor network representation of Thm. 2.24 by the selection encoding $\sigma^{\tilde{\mathcal{F}}}\left[X_{[d]}, L\right]$ as

$$\left\langle \sigma^{\tilde{\mathcal{F}}}\left[X_{[d]}, L\right], \exp\left[\theta\left[L\right]\right]\right\rangle \left[X_{[d]}\right] = \left\langle \bigcup_{l\in[p]} \left\{\beta^{f_l}\left[Y_l, X_{[d]}\right], \exp\left[\theta\left[L=l\right]\cdot I_l(Y_l)\right]\right\}\right\rangle \left[X_{[d]}\right].$$

14.6 Hybrid Basis and 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. 13.5). As we show here, these include conjunctions, which correspond with coordinatewise multiplication, and negation, which correspond with coordinatewise subtraction from the trivial tensor. Such schemes are applied for example in [TAP24] in batchwise logical inference.

Theorem 14.36 *For any formulas f, h we have*

$$\left\langle \beta^{\wedge}\left[Y_{f\wedge h}, Y_f, Y_h\right], \epsilon_1\left[Y_{f\wedge h}\right]\right\rangle \left[Y_f, Y_h\right] = \epsilon_1\left[Y_f\right] \otimes \epsilon_1\left[Y_h\right] .$$

In particular, it holds that (see Figure 14.2a)

$$(f \wedge h)[X_{[d]}] = \langle f, h\rangle \left[X_{[d]}\right] .$$

Proof. We decompose

$$\beta^{\wedge}\left[Y_{f\wedge h}, Y_f, Y_h\right] = \epsilon_1\left[Y_{f\wedge h}\right] \otimes \epsilon_1\left[Y_f\right] \otimes \epsilon_1\left[Y_h\right] + \epsilon_0\left[Y_{f\wedge h}\right] \left(\mathbb{I}\left[Y_f, Y_h\right] - \epsilon_1\left[Y_f\right] \otimes \epsilon_1\left[Y_h\right]\right)$$

and get the first claim as

$$\begin{aligned}\left\langle \beta^{\wedge}\left[Y_{f\wedge h}, Y_f, Y_h\right], \epsilon_1\left[Y_{f\wedge h}\right]\right\rangle \left[Y_f, Y_h\right] &= \left\langle \epsilon_1\left[Y_{f\wedge h}\right] \otimes \epsilon_1\left[Y_f\right] \otimes \epsilon_1\left[Y_h\right], \epsilon_1\left[Y_{f\wedge h}\right]\right\rangle \left[Y_f, Y_h\right] \\ &= \epsilon_1\left[Y_f\right] \otimes \epsilon_1\left[Y_h\right] .\end{aligned}$$

To show the second claim we use

$$\begin{aligned}(f \wedge h)[X_{[d]}] &= \left\langle \beta^{f}\left[Y_f, X_{[d]}\right], \beta^{h}\left[Y_h, X_{[d]}\right], \beta^{\wedge}\left[Y_{f\wedge h}, Y_f, Y_h\right], \epsilon_1\left[Y_{f\wedge h}\right]\right\rangle \left[X_{[d]}\right] \\ &= \left\langle \beta^{f}\left[Y_f, X_{[d]}\right], \beta^{h}\left[Y_h, X_{[d]}\right], \left(\epsilon_1\left[Y_f\right] \otimes \epsilon_1\left[Y_h\right]\right)\right\rangle \left[X_{[d]}\right] \\ &= \langle f, h\rangle \left[X_{[d]}\right] .\end{aligned}$$

■

A similar decomposition holds for negations, as we show next.

Theorem 14.37 *For any formula f we have*

$$\left\langle \beta^{\neg}\left[Y_{\neg f}, Y_f\right], \epsilon_1\left[Y_{\neg f}\right]\right\rangle \left[Y_f\right] = \epsilon_0\left[Y_f\right] = \mathbb{I}\left[Y_f\right] - \epsilon_1\left[Y_f\right] .$$

and

$$\left\langle \beta^{\neg}\left[Y_f, Y_{\neg f}\right], \epsilon_0\left[Y_{\neg f}\right]\right\rangle \left[Y_f\right] = \epsilon_1\left[Y_f\right] .$$

In particular, it holds that (see Figure 14.2b)

$$(\neg f)[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - f\left[X_{[d]}\right] .$$

Proof. We have

$$\beta^{\neg}\left[Y_{\neg f}, Y_f\right] = \epsilon_1\left[Y_{\neg f}\right] \otimes \epsilon_0\left[Y_f\right] + \epsilon_0\left[Y_{\neg f}\right] \otimes \epsilon_1\left[Y_f\right]$$

and therefore get the second claim by contraction with $\epsilon_0\left[Y_{\neg f}\right]$

$$\begin{aligned}\left\langle \beta^{\neg}\left[Y_f, Y_{\neg f}\right], \epsilon_0\left[Y_{\neg f}\right]\right\rangle \left[Y_f\right] &= \left\langle \epsilon_1\left[Y_{\neg f}\right] \otimes \epsilon_0\left[Y_f\right], \epsilon_0\left[Y_{\neg f}\right]\right\rangle \left[Y_f\right] \\ &\quad + \left\langle \epsilon_0\left[Y_{\neg f}\right] \otimes \epsilon_1\left[Y_f\right], \epsilon_0\left[Y_{\neg f}\right]\right\rangle \left[Y_f\right] \\ &= \epsilon_1\left[Y_f\right] .\end{aligned}$$

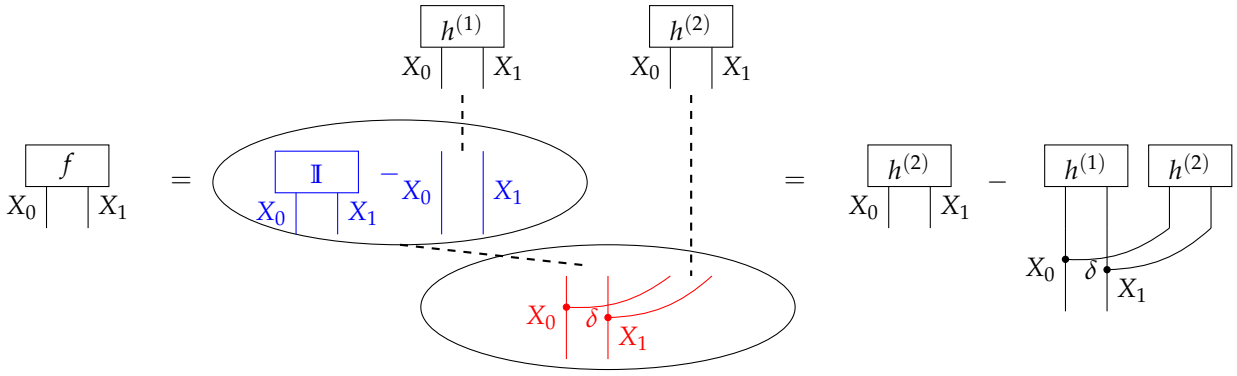


Figure 14.3: Example of a decomposition by hybrid calculus of a formula $f[X_1, X_2] = \neg h^{(1)}[X_1, X_2] \wedge h^{(2)}[X_1, X_2]$ into a sum of contractions.

The first equation of the first claim follows similarly by contraction with $\epsilon_1\left[Y_{\neg f}\right]$ as

$$\left\langle \beta^{\neg}\left[Y_{\neg f}, Y_f\right], \epsilon_1\left[Y_{\neg f}\right]\right\rangle\left[Y_f\right] = \epsilon_0\left[Y_f\right]$$

and the second equation using that $\mathbb{I}[X] = \epsilon_0[X] + \epsilon_1[X]$ and hence

$$\epsilon_0\left[Y_f\right] = \mathbb{I}\left[Y_f\right] - \epsilon_1\left[Y_f\right] .$$

To show the third claim, we contract the computation tensor $\beta^f\left[Y_f, X_{[d]}\right]$ to the formula f on both sides of the first claim and get

$$\begin{aligned}&\left\langle \beta^f\left[Y_f, X_{[d]}\right], \beta^{\neg}\left[Y_{\neg f}, Y_f\right], \epsilon_1\left[Y_{\neg f}\right]\right\rangle\left[Y_f\right]\\ &\quad = \left\langle \beta^f\left[Y_f, X_{[d]}\right], \mathbb{I}\left[Y_f\right]\right\rangle\left[Y_f\right] - \left\langle \beta^f\left[Y_f, X_{[d]}\right], \epsilon_1\left[Y_f\right]\right\rangle\left[Y_f\right] .\end{aligned}$$

We simplify this equation using the trivial head contraction identity of directed tensors of Cor. 14.16 stating that

$$f\left[X_{[d]}\right] = \left\langle \beta^f\left[Y_f, X_{[d]}\right], \epsilon_1\left[Y_f\right]\right\rangle\left[X_{[d]}\right] \quad \text{and} \quad \neg f\left[X_{[d]}\right] = \left\langle \beta^f\left[Y_f, X_{[d]}\right], \epsilon_0\left[Y_f\right]\right\rangle\left[X_{[d]}\right]$$

and arrive at the third claim

$$(\neg f)[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - f\left[X_{[d]}\right] . \qquad \blacksquare$$

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 14.3 we provide an example of this decomposition.

14.7 Applications in Machine Learning

Basis calculus provides a tool suited to represent decompositions of function by efficient tensor networks. The decomposition of function into smaller components, called neurons, is often referred to as the neural paradigm of machine learning. 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. Inference is then performed by contractions with basis tensors representing the input, as shown in Thm. 14.11. These contractions can further be executed neuron-wise.

What is more, basis calculus provides an efficient scheme to represent symbols such as logical connectives by their basis encodings. Basis calculus is therefore a tool for neuro-symbolic AI [Gar+19; Sar+22; Mar+24].

15

Sparse Representation

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.

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 $\lambda[I]$. With the corresponding singular vectors by $\rho^0[X_0, I]$ and $\rho^1[X_1, I]$, the singular value decomposition of M is

$$M[X_0, X_1] = \sum_{i \in [n]} \lambda[I = i] \cdot \rho^0[X_0, I = i] \otimes \rho^1[X_1, I = i] \,.$$

Here the smallest n such that this decomposition exists, is the matrix rank rank (M). In contraction notation we abbreviate this to

$$M[X_0, X_1] = \left\langle \lambda[I], \rho^0[X_0, I], \rho^1[X_1, I] \right\rangle [X_0, X_1] \,.$$

Given a tensor of higher order, a generalization of this decomposition is a tensor product over multiple vectors, as we define next.

Definition 15.1 A CP decomposition of size n of a tensor $\tau\left[X_{[d]}\right] \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ is a collections of a scalar core $\lambda[I]$ and leg cores $\rho^k[I, X_k]$ for $k \in [d]$, where I is an enumeration variable taking values in $[n]$, such that

$$\tau\left[X_{[d]}\right] = \left\langle \{\lambda[I]\} \cup \{\rho^k[X_k, I] \,:\, k \in [d]\} \right\rangle \left[X_{[d]}\right] \,.$$

We say that the CP Decomposition is

- directed, when for each k the core ρ^k is directed with I incoming and X_k outgoing.
- boolean, when for each k the core ρ^k is boolean.
- basis, where we demand both properties, that is for each $k \in [d]$ and $i \in [n]$

$$\rho^k[X_k, I = i] \in \{\epsilon_{x_k}[X_k] \,:\, x_k \in [m_k]\} \,.$$

- basis+, when for each $k \in [d]$ and $i \in [n]$

$$\rho^k[X_k, I = i] \in \{\epsilon_{x_k}[X_k] \,:\, x_k \in [m_k]\} \cup \{\mathbb{I}[X_k]\} \,.$$

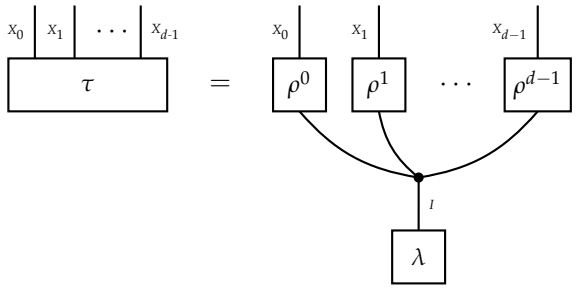


Figure 15.1: Tensor Network diagram of a generic CP decomposition (see Def. 15.1)

We denote by $\operatorname{rank}(\tau)$, respectively $\operatorname{rank}^{\text{bin}}(\tau)$, $\operatorname{rank}^{\text{bas}}(\tau)$ and $\operatorname{rank}^{\text{bas+}}(\tau)$ the minimal cardinality such that τ 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)

$$\tau\left[X_{[d]}\right] = \sum_{x_{[d]} \in \bigtimes_{k\in[d]}[m_k]} \tau\left[X_{[d]} = x_{[d]}\right] \cdot \bigotimes_{k\in[d]} \epsilon_k\left[X_k\right] .$$

If we construct i as an enumeration of the coordinates in $\bigtimes_{k\in[d]}[m_k]$, 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 [BM05] and approximation problems are often ill posed [SL08]. Since as a consequence their numerical treatment comes with many problems [Esp+12], alternative formats have gained popularity. Common formats are the TUCKER-format originally introduced in [Hit27], and often referred to as higher-order singular value decomposition, and the more recently developed TT and HT decomposition formats (see Chapter). Given a HT the best approximation of a tensor always exists (Theorem 11.58 in [Hac12]).

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 15.2 *For any tensor $\tau\left[X_{[d]}\right]$ we have*

$$\operatorname{rank}(\tau) = \operatorname{rank}^{\text{dir}}(\tau) .$$

Proof. Let there be a CP decomposition of τ by

$$\tau\left[X_{[d]}\right] = \left\langle \{\lambda[I]\} \cup \{\rho^k\left[X_k, I\right] : k \in [d]\} \right\rangle \left[X_{[d]}\right] .$$

We then transform the scalar core to another core $\tilde{\lambda}[I]$ with coordinates to $i \in [n]$ by

$$\tilde{\lambda}[I = i] = \lambda[I = i] \cdot \prod_{k\in[d]} \left\langle \rho^k\left[X_k, I = i\right] \right\rangle [\varnothing] .$$

It follows for any $i \in [n]$, that

$$\lambda[I = i] \cdot \bigotimes_{k \in [d]} \rho^k [X_k, I = i] = \tilde{\lambda}[I = i] \cdot \bigotimes_{k \in [d]} \left\langle \rho^k [X_k, I = i] \right\rangle [X_k | \varnothing]$$

and thus

$$\tau \left[X_{[d]} \right] = \left\langle \{ \tilde{\lambda}[I] \cup \{ \left\langle \rho^k [X_k, I] \right\rangle [X_k, I | \varnothing] \ : \ k \in [d] \} \right\rangle \left[X_{[d]} \right] .$$

We have thus constructed a directed CP decomposition of same size n to an arbitrary CP decomposition and conclude that $\operatorname{rank}(\tau) = \operatorname{rank}^{\mathrm{dir}}(\tau)$. ∎

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 15.3 The ℓ_0-norm counts the nonzero coordinates of a tensor by

$$\ell_0(\tau) = \#\{ x_0, \ldots, x_{d-1} \ : \ \tau_{x_0, \ldots, x_{d-1}} \neq 0 \} .$$

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 basis 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 $\ell_0(\tau)$.

Theorem 15.4 *For any tensor* $\tau \left[X_{[d]} \right]$ *we have*

$$\operatorname{rank}^{\mathrm{bas}}(\tau) = \ell_0(\tau) .$$

Proof. Let us first show, that $\operatorname{rank}^{\mathrm{bas}}(\tau) \leq \ell_0(\tau)$. We find a map

$$D : [\ell_0(\tau)] \to \bigtimes_{k \in [d]} [m_k]$$

which image is the set of non-vanishing coordinates of $\tau \left[X_{[d]} \right]$. Denoting its image coordinate maps by D_k we have

$$\tau \left[X_{[d]} \right] = \sum_{j \in [m]} \lambda[D(()j)] \left(\bigotimes_{k \in [d]} \epsilon_{D_k(j)} [X_k] \right) .$$

This is a basis CP decomposition of size $\ell_0(\tau)$ and we thus have $\operatorname{rank}^{\mathrm{bas}}(\tau) \leq \ell_0(\tau)$.

Conversely, let us show $\operatorname{rank}^{\mathrm{bas}}(\tau) \geq \ell_0(\tau)$. Any basis CP decomposition of τ with size r would has at most r coordinates different from zero and thus $\ell_0(\tau) \leq r$. Thus, there cannot be a CP decomposition with a dimension $r \leq \ell_0(\tau)$. ∎

The next theorem relates the basis CP decomposition with encodings of d-ary relations (see Def. 14.5).

Theorem 15.5 *Any boolean tensor* $\tau\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ *is the encoding of a d-ary relation* $\mathcal{R} \subset \bigtimes_{k\in[d]}[m_k]$ *with cardinality*

$$|\mathcal{R}| = \mathrm{rank}^{\mathrm{bas}}(\tau) \ .$$

Proof. We fine a basis CP decomposition of $\tau\left[X_{[d]}\right]$ with $n = \mathrm{rank}^{\mathrm{bas}}(\tau)$. Since $\tau\left[X_{[d]}\right]$ is boolean, and since each i labels a disjoint non-vanishing coordinate (see proof of Thm. 15.4), the decomposition has a trivial scalar core $\lambda[I] = \mathbb{I}[I]$. It follows, that

$$\tau\left[X_{[d]}\right] = \sum_{i\in[n]} \left(\bigotimes_{k\in[d]} \rho^k\left[X_k, I = i\right] \right)$$

Since the CP decomposition is basis, the slice $\rho^k[X_k, I = i]$ is for any $k \in [d]$ and $i \in [n]$ a basis vector. We then define

$$x_k^i = \epsilon^{-1}(\rho^k[X_k, I = i])$$

and notice, that for the relation

$$\mathcal{R} = \{x^i_{[d]} : i \in [n]\} \subset \bigtimes_{k\in[d]} [m_k]$$

we have

$$\epsilon_{\mathcal{R}}\left[X_{[d]}\right] = \sum_{i\in[n]} \left(\bigotimes_{k\in[d]} \rho^k\left[X_k, I = i\right] \right) .$$

This coincides with the above CP decomposition of $\tau\left[X_{[d]}\right]$ and the claim is established. ∎

Remark 15.6 (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 $\epsilon_\cdot$, 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

$$M[I = i, L = k] = \begin{cases} \lambda[I = i] & \text{if} \quad k = d \\ \epsilon^{-1}(\rho^k[X_k, I = i]) & \text{else} \end{cases} .$$

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 15.7 A monomial decomposition of a tensor $\tau\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ 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

$$\tau\left[X_{[d]}\right] = \sum_{(\lambda,A,x_A)\in\mathcal{M}} \lambda \cdot \langle \epsilon_{x_A}[X_A]\rangle \left[X_{[d]}\right] . \tag{15.1}$$

For any tensor $\tau\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we define its polynomial sparsity of order r as

$$\text{rank}^{\text{bas+},r}(\tau) = \min\left\{ |\mathcal{M}| \,:\, \tau\left[X_{[d]}\right] = \sum_{(\lambda,A,x_A)\in\mathcal{M}} \lambda \cdot \langle \epsilon_{x_A}[X_A]\rangle \left[X_{[d]}\right] , \forall_{(\lambda,A,x_A)\in\mathcal{M}} |A| \le r . \right\}$$

We refer to the terms in a decomposition (15.1) in Def. 15.7 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

$$\mathbb{I}_{X_k=x_k}[X_k] = \epsilon_{x_k}[X_k] .$$

The monomial of multiple such boolean features indicates, whether all variables labelled by a set A are in the state X_A. We have

$$\mathbb{I}_{\forall k\in A:\, X_k=x_k}[X_A] = \epsilon_{x_A}[X_A] = \bigotimes_{k\in A} \epsilon_{x_k}[X_k] .$$

The states of the variables labeled by $k \in [d]/A$ are not specified in the monomial and the indicators are trivially extended to

$$\langle \epsilon_{x_A}[X_A]\rangle \left[X_{[d]}\right] = \epsilon_{x_A}[X_A] \otimes \mathbb{I}\left[X_{[d]/A}\right] .$$

Since we are working with boolean features, there is no need to consider higher-order powers of individual features, since for any $n \in \mathbb{N}$, $n \ge 1$ and any boolean value $z \in \{0, 1\}$ we have $z^n = z$.

For some monomial orders $r < d$ there are tensors $\tau\left[X_{[d]}\right]$, which do not have a monomial decomposition of order r. In that case the minimum is over an empty set and we define $\text{rank}^{\text{bas+},r}(\tau) = \infty$. We characterize in the next theorem the set of tensors with monomial decompositions of order r.

Theorem 15.8 *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*

$$\dim(V^{d,r}) \le \sum_{s\in[r]} m^s \binom{d}{s} .$$

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 $\langle \epsilon_{x_A} \rangle \left[X_{[d]} \right]$. 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 [CKP13]. The number of elements in the frame is therefore in general a loose upper bound on the dimension. ■

Thm. 15.8 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 $\text{rank}^{\text{bas+},r}(\tau)$ furthermore coincides with the basis+ CP rank $\text{rank}^{\text{bas+}}(\tau)$.

Theorem 15.9 *For any tensor $\tau \left[X_{[d]} \right] \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ we have*

$$\text{rank}^{\text{bas+},d}(\tau) = \text{rank}^{\text{bas+}}(\tau) .$$

In case of two-dimensional legs (i.e. $m_k = 2$ for all $k \in [d]$) we also have

$$\text{rank}^{\text{bin}}(\tau) = \text{rank}^{\text{bas+},d}(\tau) .$$

Proof. To prove the first claim, we construct a basis+ CP decomposition given a monomial decomposition and vice versa. To show $\text{rank}^{\text{bas+},d}(\tau) \geq \text{rank}^{\text{bas+}}(\tau)$, let there be an arbitrary tensor $\tau \left[X_{[d]} \right]$ 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

$$\rho^k [I, X_k] = \left(\sum_{i \in [n] : k \in A} \epsilon_i [I] \otimes \epsilon_{x^i_k} [X_k] \right) + \left(\sum_{i \in [n] : k \notin A} \epsilon_i [I] \otimes \mathbb{I} [X_k] \right)$$

and

$$\lambda[I] = \sum_{i \in [n]} \lambda^i \cdot \epsilon_i [I]$$

and notice that

$$\begin{aligned} \tau \left[X_{[d]} \right] &= \sum_{i \in [n]} \lambda^i \cdot \left\langle \epsilon_{x^i_A} \right\rangle \left[X_{[d]} \right] \\ &= \sum_{i \in [n]} \left(\lambda[I = i] \cdot \bigotimes_{k \in [d]} \rho^k [I = i, X_k] \right) \\ &= \left\langle \{\lambda[I]\} \cup \{\rho^k [I, X_k] \,:\, k \in [d]\} \right\rangle \left[X_{[d]} \right] . \end{aligned}$$

By construction this is a basis+ CP decomposition with rank n. Since any monomial decomposition can be transformed into a basis+ CP decomposition with same rank we have

$$\mathrm{rank}^{\mathrm{bas+},d}(\tau) \geq \mathrm{rank}^{\mathrm{bas+}}(\tau) .$$

To show $\mathrm{rank}^{\mathrm{bas+},d}(\tau) \leq \mathrm{rank}^{\mathrm{bas+}}(\tau)$, let there now be a basis+ CP decomposition of an arbitrary $\tau\left[X_{[d]}\right]$. We define for each $i \in [n]$

$$A^i = \{k \in [d] : \rho^k[I = i, X_k] \neq \mathbb{I}[X_k]\} \quad \text{and} \quad x^i_A = \{\epsilon^{-1}(\rho^k[I = i, X_k]) : k \in A\}$$

where by $\epsilon^{-1}(\cdot)$ we denote the inverse of the one-hot encoding.

We notice that this is a monomial decomposition of $\tau\left[X_{[d]}\right]$ to the tuple set

$$\mathcal{M} = \{(\lambda[I = i], A^i, x^i_{A^i}) : i \in [n]\} .$$

It follows from this that

$$\mathrm{rank}^{\mathrm{bas+},d}(\tau) \leq \mathrm{rank}^{\mathrm{bas+}}(\tau)$$

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 15.10 (Sparse representation of propositional formulas) When all leg dimensions of a boolean tensor τ are 2, we can further interpret τ as a logical formula. We can use the boolean CP decomposition of any tensor $\tilde{\tau}$ with $\mathbb{I}_{\neq 0}(\tilde{\tau}) = \tau$ as a CNF of τ. Finding the sparsest CNF thus amounts to finding the $\tilde{\tau}$ with minimal $\mathrm{rank}^{\mathrm{bas+},d}(\tilde{\tau})$ such that $\mathbb{I}_{\neq 0}(\tilde{\tau}) = \tau$. ◊

Remark 15.11 (Matrix Storage of basis+ CP decompositions) We can adapt the storage format of Remark 15.6 from basis to basis+ CP decompositions. To this end, let there be a basis+ CP decomposition of a tensor with scalar core $\lambda[I]$ and leg cores $\{\rho^k[X_k, I] : k \in [d]\}$. We use a value $z \in \mathbb{R}/\mathrm{im}(\lambda)$ distinguished from the coordinates of the scalar core and define a matrix

$$M^z[I, L]$$

where L takes values in $[d]$, coordinatewise as

$$M^z[I = i, L = k] = \begin{cases} \lambda[I = i] & \text{if} \quad k = d \\ z & \text{if} \quad \rho^k[X_k, I = i] = \mathbb{I}[X_k] \\ \epsilon^{-1}(\rho^k[X_k, I = i]) & \text{else} \end{cases} .$$

◊

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. 15.1.

Theorem 15.12 *For any tensor* $\tau\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ *we have*

$$\operatorname{rank}(\tau) = \operatorname{rank}^{\mathrm{dir}}(\tau) \leq \operatorname{rank}^{\mathrm{bin}}(\tau) \leq \operatorname{rank}^{\mathrm{bas}+}(\tau) \leq \operatorname{rank}^{\mathrm{bas}}(\tau) .$$

Proof. The equality $\operatorname{rank}(\tau) = \operatorname{rank}^{\mathrm{dir}}(\tau)$ has been established in Lem. 15.2. 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 $\tau\left[X_{[d]}\right]$ 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 $\tau\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$, for which we have

$$\operatorname{rank}^{\mathrm{bas}}(\tau) = \ell_0(\tau) = \prod_{k\in[d]} m_k .$$

Since in the other restricted CP formats we can choose trivial slices to the leg cores, we have

$$\operatorname{rank}^{\mathrm{bas}+}(\tau) = 1 = \operatorname{rank}^{\mathrm{bin}}(\tau) = \operatorname{rank}^{\mathrm{dir}}(\tau) = \operatorname{rank}(\tau) .$$

The trivial tensor serves thus as an example, where the demand of the the storage format in Remark 15.6 has an exponential overhead compared to the storage format in Remark 15.11.

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 normalizations of the represented tensors. Let us here investigate, how these operations influence the decomposition.

15.2.2.1 Summation

We start with the sum of tensors in a CP decomposition, which can be captured by a concatenation of the slices.

Theorem 15.13 *For any collections of tensors* $\{\tau^l[X_{\mathcal{V}}] : l \in [p]\}$ *with identical variables and scalars* $\lambda^l \in \mathbb{R}$ *for* $l \in [p]$ *we have*

$$\operatorname{rank}\left(\sum_{l\in[p]} \lambda^l \cdot \tau^l\right) \leq \sum_{l\in[p]} \operatorname{rank}\left(\tau^l\right) .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\mathrm{bin}}(\cdot)$*, by* $\operatorname{rank}^{\mathrm{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\mathrm{bas}+}(\cdot)$.

Proof. Products with scalars do not change the rank, since they just rescale the core λ. 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.

15.2.2.2 Contraction

We continue to show rank bounds for arbitrary contractions by the product of the ranks of contracted tensors.

Theorem 15.14 *For any tensor network $\tau^{\mathcal{G}}\left[X_{\mathcal{V}}\right]$ on a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, we have for any subset $\tilde{\mathcal{V}}\subset\mathcal{V}$*

$$\operatorname{rank}\left(\left\langle \tau^{\mathcal{G}}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\right)\leq\prod_{e\in\mathcal{E}\,:\,\tilde{\mathcal{V}}\cap e\neq\varnothing}\operatorname{rank}\left(\tau^{e}\right).$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\mathrm{bin}}(\cdot)$*, by* $\operatorname{rank}^{\mathrm{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\mathrm{bas}+}(\cdot)$.

Remarkably, in Thm. 15.14 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 15.15 *For any tensors $\tau^{1}\left[X_{\mathcal{V}^{1}}\right]$ and $\tau^{2}\left[X_{\mathcal{V}^{2}}\right]$ and any set of variables $\tilde{\mathcal{V}}\subset\mathcal{V}^{1}\cup\mathcal{V}^{2}$ we have*

$$\operatorname{rank}\left(\left\langle \tau^{1},\tau^{2}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\right)\leq\operatorname{rank}\left(\tau^{1}\right)\cdot\operatorname{rank}\left(\tau^{2}\right).$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\mathrm{bin}}(\cdot)$*, by* $\operatorname{rank}^{\mathrm{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\mathrm{bas}+}(\cdot)$.

Proof. Let there be CP decompositions of $\tau^{1}\left[X_{\mathcal{V}^{1}}\right]$ and $\tau^{2}\left[X_{\mathcal{V}^{2}}\right]$ by

$$\tau^{1}\left[X_{\mathcal{V}^{1}}\right]=\left\langle\{\lambda^{1}[I_{1}]\}\cup\{\rho^{1,k}\left[X_{k},I_{1}\right]\,:\,k\in\mathcal{V}^{1}\}\right\rangle\left[X_{\mathcal{V}^{1}}\right]$$

and

$$\tau^{2}\left[X_{\mathcal{V}^{1}}\right]=\left\langle\{\lambda^{2}[I_{2}]\}\cup\{\rho^{2,l}\left[X_{l},I_{2}\right]\,:\,l\in\mathcal{V}^{2}\}\right\rangle\left[X_{\mathcal{V}^{2}}\right].$$

By linearity of contractions we have

$$\begin{aligned}\left\langle\tau^{1},\tau^{2}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]&=\sum_{i_{1}\in n_{1}}\sum_{i_{2}\in n_{2}}\lambda^{1}[I_{1}=i_{1}]\cdot\lambda^{2}[I_{2}=i_{2}]\\&\quad\cdot\left\langle\{\rho^{1,k}\left[X_{k},I_{1}=i_{1}\right]\,:\,k\in\mathcal{V}^{1}\}\cup\{\rho^{2,l}\left[X_{l},I_{2}=i_{2}\right]\,:\,l\in\mathcal{V}^{2}\}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\\&=\sum_{i_{1}\in n_{1}}\sum_{i_{2}\in n_{2}}\lambda^{1}[I_{1}=i_{1}]\cdot\lambda^{2}[I_{2}=i_{2}]\cdot\\&\quad\left\langle\{\rho^{1,k}\left[X_{k},I_{1}=i_{1}\right]\,:\,k\in\mathcal{V}^{1}/\tilde{\mathcal{V}}\}\cup\{\rho^{2,l}\left[X_{l},I_{2}=i_{2}\right]\,:\,l\in\mathcal{V}^{2}/\tilde{\mathcal{V}}\}\right\rangle\left[\varnothing\right]\cdot\\&\quad\bigotimes_{k\in\tilde{\mathcal{V}}}\rho^{k}\left[X_{k},I_{1}=i_{1},I_{2}=i_{2}\right],\end{aligned}$$

where we denote

$$\rho^{k}\left[X_{k},I_{1}=i_{1},I_{2}=i_{2}\right]=\begin{cases}\rho^{1,k}\left[X_{k},I_{1}=i_{1}\right] & \text{if}\quad k\notin\mathcal{V}^{2}\\ \rho^{2,k}\left[X_{k},I_{2}=i_{2}\right] & \text{if}\quad k\notin\mathcal{V}^{1}\\ \left\langle\rho^{1,k}\left[X_{k},I_{1}=i_{1}\right],\rho^{2,k}\left[X_{k},I_{2}=i_{2}\right]\right\rangle\left[X_{k}\right] & \text{else}\end{cases}.$$

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 $\left\langle \tau^1, \tau^2 \right\rangle [X_{\tilde{\mathcal{V}}}]$ of size $n = n_1 \cdot n_2$. This shows the claim in the case of rank $(\cdot)$.

When the CP decompositions of τ^1 and τ^2 are boolean, basis or basis+, then the property is preserved in the constructed CP decomposition, since the constructed slices $\rho^k [X_k, I_1 = i_1, I_2 = i_2]$ 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 $\text{rank}^{\text{bin}}(\cdot)$, $\text{rank}^{\text{bas}}(\cdot)$ and $\text{rank}^{\text{bas+}}(\cdot)$. ∎

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. 15.15 as we show next.

Lemma 15.16 *For any two tensors $\tau^1 [X_{\mathcal{V}^1}]$, $\tau^2 [X_{\mathcal{V}^2}]$ and any set $\tilde{\mathcal{V}}$ with $\tilde{\mathcal{V}} \cap \mathcal{V}^2 = \varnothing$ we have*

$$\text{rank}\left(\left\langle \tau^1, \tau^2 \right\rangle [X_{\tilde{\mathcal{V}}}]\right) \leq \text{rank}\left(\tau^1\right).$$

The bound still holds, when we replace on both sides $\text{rank}(\cdot)$ *by* $\text{rank}^{\text{bin}}(\cdot)$, *by* $\text{rank}^{\text{bas}}(\cdot)$ *or by* $\text{rank}^{\text{bas+}}(\cdot)$.

Proof. As in the proof of Lem. 15.15 we assume a CP decomposition of $\tau^1 [X_{\mathcal{V}^1}]$ and $\tau^2 [X_{\mathcal{V}^2}]$ and use the linearity of contractions to get

$$\begin{aligned}\left\langle \tau^1, \tau^2 \right\rangle [X_{\tilde{\mathcal{V}}}] = \sum_{i_1 \in n_1} \sum_{i_2 \in n_2} \lambda^1[I_1 = i_1] \cdot \lambda^2[I_2 = i_2] \cdot \\ \left\langle \{\rho^{1,k} [X_k, I_1 = i_1] : k \in \mathcal{V}^1/\tilde{\mathcal{V}}\} \cup \{\rho^{2,l} [X_l, I_2 = i_2] : l \in \mathcal{V}^2/\tilde{\mathcal{V}}\} \right\rangle [\varnothing] \cdot \\ \bigotimes_{k \in \tilde{\mathcal{V}}} \rho^{1,k} [X_k, I_1 = i_1] ,\end{aligned}$$

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

$$\begin{aligned}\lambda[I_1 = i_1] = \sum_{i_2 \in n_2} \lambda^1[I_1 = i_1] \cdot \lambda^2[I_2 = i_2] \cdot \\ \left\langle \{\rho^{1,k} [X_k, I_1 = i_1] : k \in \mathcal{V}^1/\tilde{\mathcal{V}}\} \cup \{\rho^{2,l} [X_l, I_2 = i_2] : l \in \mathcal{V}^2/\tilde{\mathcal{V}}\} \right\rangle [\varnothing] .\end{aligned}$$

This shows the rank bound for rank $(\cdot)$. The properties of the CP decomposition are trivially inherited by the constructed decomposition, since the leg cores of the decomposition of $\tau^1 [X_{\mathcal{V}^1}]$ are chosen. Thus, the rank bounds hold also for any other rank in the claim. ∎

Proof of Thm. 15.14. 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

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \left\langle \{\tau^e [X_e] : e \in \mathcal{E}^1\} \right\rangle \left[X_{\bigcup_{e \in \mathcal{E}^1} e}\right], \left\langle \{\tau^e [X_e] : e \in \mathcal{E}^2\} \right\rangle \left[X_{\bigcup_{e \in \mathcal{E}^2} e}\right] \right\rangle [X_{\tilde{\mathcal{V}}}] \quad (15.2)$$

By an iterative application of Lem. 15.15 when including the cores to $e \in \mathcal{E}^1$ after each other to the contraction, we get the bound

$$\text{rank}\left(\left\langle \{\tau^e [X_e] : e \in \mathcal{E}^1\} \right\rangle \left[X_{\bigcup_{e \in \mathcal{E}^1} e}\right]\right) \leq \prod_{e \in \mathcal{E}^1} \text{rank}(\tau^e) .$$

With the decomposition (15.2) and Lem. 15.16 we then arrive at the claim

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] \leq \prod_{e \in \mathcal{E}^1} \operatorname{rank}(\tau^e) \,.$$

Since the applied lemmata hold also for the restricted CP ranks in the claim, the derived bound is also for those valid. ∎

Example 15.17 (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 $\beta^{\neg}$ which has polynomial sparsity of 2. This is not optimal, since $\neg f$ has at most an absolute polynomial 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 15.18 (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 basis 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 15.19 *For any function*

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

between factored systems we have

$$\operatorname{rank}^{\mathrm{bas}}(\beta^q) = \prod_{k \in [d]} m_k \,.$$

Proof. The bound follows from Thm. 15.4, using that $\ell_0(\beta^q) = \prod_{k \in [d]} m_k$. ∎

Let us further provide a construction scheme to find a basis CP decomposition of β^q of size $\prod_{k \in [d]} m_k$. We notice that

$$\beta^q \left[Y_{[p]}, X_{[d]} \right] = \sum_{x_{[d]} \in \bigtimes_{k \in [d]} [m_k]} \epsilon_{x_{[d]}} \left[X_{[d]} \right] \otimes \epsilon_{q(X_{[d]} = x_{[d]})} \left[Y_{[p]} \right] \,.$$

We build for $k \in [d]$ decomposition variables $I_{[d]}$ with $n_k = m_k$ and define leg cores

$$\rho^k \left[X_k, I_{[d]} \right] = \delta \left[X_k, I_k \right]$$

and for $\tilde{k} \in [r]$ and $i_{[d]}$

$$\rho^{\tilde{k}} \left[Y_{\tilde{k}}, I_{[d]} = i_{[d]} \right] = \epsilon_{q_{\tilde{k}}[X_{[d]} = i_{[d]}]} Y_{\tilde{k}} \,.$$

We then have with a trivial scalar core

$$\beta^q \left[Y_{[p]}, X_{[d]} \right] = \left\langle \{ \mathbb{I} \left[I_{[d]} \right] \} \cup \{ \rho^k \left[X_k, I_{[d]} \right] : k \in [d] \} \cup \{ \rho^{\tilde{k}} \left[Y_{\tilde{k}}, I_{[d]} \right] : \tilde{k} \in [r] \} \right\rangle \left[Y_{[p]}, X_{[d]} \right] \,.$$

This is a basis CP decomposition of size $\prod_{k\in[d]} m_k$.

In combination with Thm. 15.12, Thm. 15.19 also provides bounds on all other CP ranks defined in Def. 15.1. This is obvious, since basis leg slices are the most restrictive properties compared with boolean, directed or basis+.

We restate Thm. 14.17 as a basis CP decomposition bound.

Theorem 15.20 *Let q and be a function between factored systems*

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

and q_k as in Thm. 14.17. Then $\beta^q \left[X_{[d]}, X\right]$ has a basis CP *decomposition with decomposition index X, trivial slices $\mathbb{I}[X]$ leg vectors $\beta^{q_k}[X_k, X]$, that is*

$$\beta^q \left[X, X_{[d]}\right] = \langle \{\mathbb{I}[X]\} \cup \{\beta^{q_k}[X_k, X] \,:\, k \in [d]\} \rangle \left[X_{[d]}\right]$$

Proof. The claimed decomposition directly follows from Thm. 14.17, since the trivial scalar core $\lambda[X] = \mathbb{I}[X]$ does not influence the contraction and can be omitted. ∎

Basis CP decompositions can be constructed by understanding the variable O_{in} of the basis encoding of a function $q : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ as the slice selection variable.

Example 15.21 (Empirical distributions, see Thm. 3.5) Let there be a data map

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

We can use Thm. 15.20 to find a tensor network representation fo β^D as

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

This representation is a basis CP decomposition, when adding trivial scalar core. This provides also a basis CP decomposition for the empirical distribution, since normalization can be done by setting a slice core to $\frac{1}{m}\mathbb{I}[J]$. ♢

Example 15.22 Exponential families The statistic has a CP decomposition with rank by the cardinality of states, that is

$$\beta^{\mathcal{S}} \left[Y_{[p]}, X_{[d]}\right] = \left\langle \{\beta^{s_l} \left[Y_l, X_{[d]}\right] \,:\, l \in [p]\} \right\rangle \left[Y_{[p]}, X_{[d]}\right] .$$

♢

While Thm. 15.19 and Thm. 15.20 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 15.23 *Let $q : \mathcal{U}^{\text{in}} \to \mathcal{U}^{\text{out}}$ be an arbitrary function and let us consider for each $y \in \text{im}(q)$ the indicator*

$$\mathbb{I}^{q=y}[O_{\text{in}}] = \begin{cases} 1 & \text{if} \quad q(O_{\text{in}} = o_{\text{in}}) = y \\ 0 & \text{else} . \end{cases}$$

The basis+ rank of the basis encoding of q then obeys the bound

$$\operatorname{rank}\left(\beta^{q}\right) \leq \sum_{y \in \operatorname{im}(q)} \operatorname{rank}\left(\mathbb{I}^{q=y}\right) .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\mathrm{bin}}(\cdot)$*, by* $\operatorname{rank}^{\mathrm{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\mathrm{bas}+}(\cdot)$*.*

Proof. We have

$$\beta^{q}\left[O_{\mathrm{out}}, O_{\mathrm{in}}\right] = \sum_{y \in \operatorname{im}(q)} \epsilon_{I(y)}\left[O_{\mathrm{out}}\right] \otimes \mathbb{I}^{q=y}\left[O_{\mathrm{in}}\right] .$$

For any $y \in \operatorname{im}(q)$ it is obvious that

$$\operatorname{rank}\left(\epsilon_{I(y)}\left[O_{\mathrm{out}}\right] \otimes \mathbb{I}^{q=y}\left[O_{\mathrm{in}}\right]\right) = \operatorname{rank}\left(\mathbb{I}^{q=y}\left[O_{\mathrm{in}}\right]\right) ,$$

which also holds true for the other bounds in the claim. We then use the summation bound of Thm. 15.13 to get

$$\begin{aligned}\operatorname{rank}\left(\beta^{q}\left[O_{\mathrm{out}}, O_{\mathrm{in}}\right]\right) &\leq \sum_{y \in \operatorname{im}(q)} \operatorname{rank}\left(\epsilon_{I(y)}\left[O_{\mathrm{out}}\right] \otimes \mathbb{I}^{q=y}\left[O_{\mathrm{in}}\right]\right) \\ &\leq \sum_{y \in \operatorname{im}(q)} \operatorname{rank}\left(\mathbb{I}^{q=y}\left[O_{\mathrm{in}}\right]\right) .\end{aligned}$$

Again, the bound still hold for the other ranks in the claim. ∎

The above claim still holds when replacing $\operatorname{rank}^{\mathrm{bas}+}(\cdot)$ with the ranks $\operatorname{rank}^{\mathrm{bas}}(\cdot)$ or $\operatorname{rank}^{\mathrm{bin}}(\cdot)$. For the rank $\operatorname{rank}^{\mathrm{bas}}(\cdot)$ it leads to the bound of Thm. 15.19, since summing the number of non zero coordinators of the indicators is the cardinality of the domain.

Example 15.24 (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 $\operatorname{im}(f) \subset \{0,1\}$ and the indicators

$$\mathbb{I}^{f=1}\left[X_{[d]}\right] = f\left[X_{[d]}\right] \quad \text{and} \quad \mathbb{I}^{f=0}\left[X_{[d]}\right] = \neg f\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] - f\left[X_{[d]}\right] .$$

For the conjunction $\wedge[X_0, X_1] = X_0 \wedge X_1$ we have

$$\beta^{\wedge}\left[X_0, X_1\right] = \epsilon_1\left[Y_{\wedge}\right] \otimes \epsilon_{1,1}\left[X_0, X_1\right] + \epsilon_0\left[Y_{\wedge}\right] \otimes \left(\mathbb{I}\left[X_0, X_1\right] - \epsilon_{1,1}\left[X_0, X_1\right]\right)$$

and thus

$$\operatorname{rank}^{\mathrm{bas}+}\left(\beta^{\wedge}\right) \leq 3$$

while $\operatorname{rank}^{\mathrm{bas}}\left(\beta^{\wedge}\right) = 4$.

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 4.10)

$$\wedge[X_{[d]}] = \bigotimes_{k \in [d]} \epsilon_1\left[X_k\right] \quad \text{and} \quad \neg\wedge[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - \bigotimes_{k \in [d]} \epsilon_1\left[X_k\right]$$

and thus

$$\beta^{\wedge}\left[X_{[d]}\right] = \epsilon_1\left[Y_{\wedge}\right] \otimes \left(\bigotimes_{k\in[d]} \epsilon_1\left[X_k\right]\right) + \epsilon_0\left[Y_{\wedge}\right] \otimes \left(\bigotimes_{k\in[d]} \epsilon_1\left[X_k\right]\right)$$

Thus, while the basis CP rank is $\mathrm{rank}^{\mathrm{bas}}\left(\beta^{\wedge}\right) = 2^d$, the basis+ rank is bounded by 3, independently of d. $\Diamond$

16

Sparse Optimization

We in this chapter study the search for the maximal coordinate in a tensor, which has a sparse basis+ CP decomposition. We then investigate methods to approximate tensors by sparse basis+ CP decomposition.

16.1 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 $\tau\left[X_{[d]}\right]$ we state this as the problem:

$$\text{argmax}_{x_{[d]} \in \times_{k\in[d]}[m_k]} \tau\left[X_{[d]} = x_{[d]}\right] \tag{$\mathrm{P}_{\tau}^{\max}$}$$

Problem ($\mathrm{P}_{\tau}^{\max}$) can be reformulated as optimization over the standard simplex

$$\mathcal{M}_{\wedge} = \text{conv}\left(\epsilon_{x_{[d]}}\left[X_{[d]}\right] \,:\, x_{[d]} \in \bigtimes_{k\in[d]} [m_k]\right)$$

as

$$\text{argmax}_{\mu[L_{[d]}] \in \mathcal{M}_{\wedge}} \langle \mu, \tau \rangle [\varnothing] \,.$$

Example 16.1 (Mode search in exponential families) Given a statistic $\mathcal{S}$, a canonical parameter θ and a boolean base measure ν, the mode search problem for the member $\mathbb{P}^{\mathcal{S},\theta,\nu}$ of the exponential family $\Gamma^{\mathcal{S},\nu}$ is

$$\max_{x_{[d]} \in \times_{k\in[d]}[2] \,:\, \nu\left[X_{[d]} = x_{[d]}\right]=1} \left\langle \sigma^{\mathcal{S}}\left[X_{[d]} = x_{[d]}, L\right], \theta\,[L] \right\rangle [\varnothing] = \max_{\mu \in \mathcal{M}_{\mathcal{S},\nu}} \langle \mu\,[L], \theta\,[L] \rangle [\varnothing] \,.$$

Such mode search problems have appeared as generic MAP queries (see Chapter 3). In Chapter 9 we have discussed them for the specific cases of Hybrid Logic Networks and grafting proposal distributions. ♢

16.1.1 Unconstrained Binary Optimization

For leg dimensions $m_k = 2$, Problem ($\mathrm{P}_{\tau}^{\max}$) is known as the unconstrained binary optimization. Problem ($\mathrm{P}_{\tau}^{\max}$) is a Higher-Order Unconstrained Binary Optimization (HUBO), when $\tau\left[X_{[d]}\right]$ has a when τ has a monomial decomposition (see Def. 15.7) with $\left|A^i\right| \leq r$ for all $i \in [n]$, that is when $\text{rank}^{\text{bas}+,r}(\tau) < \infty$.

Definition 16.2 Let $\tau\left[X_{[d]}\right]$ 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 ($\mathrm{P}_\tau^{\max}$) a r-Order Unconstrained Binary Optimization (HUBO), which we denote as

$$\operatorname{argmax}_{x_{[d]}\in\bigtimes_{k\in[d]}[2]} \quad \sum_{i\in[n]} \lambda^i \left\langle \epsilon_{x^i_{A^i}}\left[X_{A^i}\right]\right\rangle \left[X_{[d]} = x_{[d]}\right] . \tag{$\mathrm{P}_\tau^{\mathrm{HUBO}}$}$$

Remark 16.3 (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. 5.4.2). ♢

The sparsity $\operatorname{rank}^{\mathrm{bas}+,r}(\tau)$ is the minimal number of monomials, for which a weighted sum is equal to τ. Thus we interpret Problem ($\mathrm{P}_\tau^{\mathrm{HUBO}}$) as searching for the maximum in a polynomial consistent of $\operatorname{rank}^{\mathrm{bas}+,r}(\tau)$ monomial terms.

Problem ($\mathrm{P}_\tau^{\mathrm{HUBO}}$) 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 16.4 *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. ■

16.1.2 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 $v_{x_{[d]}}[L] \in \mathbb{R}^d$ with coordinates

$$v_{x_{[d]}}[L = k] = x_k .$$

Definition 16.5 The integer linear program (ILP) of $M[J, L] \in \mathbb{R}^{n\times d}$, $b[J] \in \mathbb{R}^n$ and $c \in \mathbb{R}^d$ is

the problem

$$\text{argmax}_{x_{[d]} \in \times_{k\in[d]}[m_k]} \left\langle c[L], v_{x_{[d]}}[L] \right\rangle [\varnothing] \quad \text{subject to} \quad \left\langle M[J,L], v_{x_{[d]}}[L] \right\rangle [\varnothing] \prec b[J] \,,$$

where by $\prec$ we denote partial ordering of tensors (see Def. 14.2).

We now show that any binary optimization problem of a tensor can be transformed into an integer linear program, given a monomial decomposition of the tensor $\tau\left[X_{[d]}\right]$ by $\mathcal{M} = \{(\lambda^i, A^i, x^i_{A^i}) : i \in [n]\}$. For this we choose state indices by vectors

$$y_{[d+n]} = x_0, \ldots, x_{d-1}, z_0, \ldots z_{n-1} \in \left(\bigtimes_{k\in[d]} [2] \right) \times \left(\bigtimes_{i\in[n]} [2] \right) ,$$

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

$$c^{\mathcal{M}}[L = l] = \begin{cases} \lambda^{l-d} & \text{if} \quad l > d \\ 0 & \text{else} \end{cases} . \tag{16.1}$$

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

$$M^{\mathcal{M}}[J = m_i + j, L = l] = \begin{cases} 1 - 2 \cdot x^i_{I(j)} & \text{if} \quad j < \left|A^i\right|, \; j = l \text{ and } l = I(j) \\ 1 & \text{if} \quad j < \left|A^i\right| \text{ and } l = d + i \\ -x^i_{I(j)} & \text{if } j = \left|A^i\right| \text{ and } l = I(j) \\ -1 & \text{if } j = \left|A^i\right| \text{ and } l = d + i \\ 0 & \text{else} \end{cases} . \tag{16.2}$$

Similarly, we define $b^{\mathcal{M}}[J]$ as the vector which nonvanishing coordinates are for $i \in [n]$ at

$$b^{\mathcal{M}}[J = m_i + j] = \begin{cases} 1 - x^i_{I(j)} & \text{if } j < \left|A^i\right| \\ -1 + \left|\{k \in A^i : x^i_k = 1\}\right| & \text{if } j = \left|A^i\right| \end{cases} . \tag{16.3}$$

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 16.6 *Given a monomial decomposition $\mathcal{M} = \{(\lambda^i, A^i, x^i_{A^i}) : i \in [n]\}$ of a tensor τ, let $y^{ILP,\mathcal{M}}$ be a solution of the integer linear program defined by the matrix and vectors in equations (16.1.2), (16.2) and (16.3). Then we have*

$$y^{ILP,\mathcal{M}}|_{[d]} \in \operatorname{argmax}_{x_{[d]} \in \bigtimes_{k\in[d]}[2]} \quad \tau\left[X_{[d]} = x_{[d]}\right] .$$

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

$$z_i = \left(\prod_{k\in A^i, x^i_k=0} (1 - x_k) \right) \cdot \left(\prod_{k\in A^i, x^i_k=1} x_k \right) \tag{16.4}$$

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 (16.4) 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 (16.4) 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 (16.4) 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

$$\left\langle c[L], v_{y_{[d+n]}}[L] \right\rangle [\varnothing] = \sum_{i\in[n]} \lambda^i \cdot z_i$$

$$= \sum_{i\in[n]} \lambda^i \cdot \left(\prod_{k\in A^i, x^i_k=0} (1 - x^i_k) \right) \cdot \left(\prod_{k\in A^i, x^i_k=1} x^i_k \right)$$
$$= \tau \left[X_{[d]} = x_{[d]} \right] .$$

Therefore, any solution of the ILP reduced to the first d indices corresponding with the axis of τ, is a solution of the binary optimization problem to τ. ∎

In order to achieve a sparse linear program it is benefitial to use a monomial decomposition with small order and rank. Beside this sparsity, the matrix $M^{\mathcal{M}}[J, L]$ is often ℓ_0-sparse, and has thus an efficient representation in a basis CP format. More precisely we have by the above construction

$$\ell_0 \left(M^{\mathcal{M}}[J, L] \right) \leq \sum_{i\in[n]} 3 \cdot \left| A^i \right| + 1 .$$

16.2 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 7). We first define a basis+ CP selecting tensor and then show its decomposition into a formula selecting neural network.

Definition 16.7 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

$$\sigma^{\mathcal{F}_{\wedge},d,r} \left[X_{[d]}, L_{0,0}, \ldots, L_{r-1,0}, L_{0,1}, \ldots, L_{r-1,1} \right]$$

with dimensions

$$p_{s,0} = m + 1, p_{s,1} = d \quad \text{for} \quad s \in [r]$$

and coordinates

$$\sigma^{\mathcal{F}_{\wedge},d,r} \left[X_{[d]} = x_{[d]}, L_{0,0} = l_{0,0}, \ldots, L_{r-1,0} = l_{r-1,0}, L_{0,1} = l_{0,1}, \ldots, L_{r-1,1} = l_{r-1,1} \right]$$
$$= \begin{cases} 1 & \text{if} \quad \forall k \in [d], s \in [r] : \left(l_{s,1} = k \wedge l_{s,0} \neq m \right) \Rightarrow \left(l_{s,0} = x_k \right) \\ 0 & \text{else} \end{cases} .$$

Intuitively, the selection variables $L_{s,1}$ of $\sigma^{\mathcal{F}_{\wedge},d,r}$ 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 7 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 16.8 *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*

$$\mathcal{M} = \{ i : i \in [n], \forall s, \tilde{s} \in [r] : \left(l^i_{s,1} = l^i_{\tilde{s},1} \wedge l^i_{s,1} \neq m \wedge l^i_{\tilde{s},1} \neq m \right) \Rightarrow \left(l^i_{s,1} = l^i_{\tilde{s},1} \right) \}$$

Then

$$\left\langle \sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle\left[X_{[d]}\right]$$
$$= \sum_{i\in\mathcal{M}} \theta\left[L_{[r]\times[m+1]} = l^i_{[r]\times[m+1]}\right]\left\langle\{\epsilon_{l_{s,0}}\left[X_{l_{s,1}}\right] : s\in[r],\, l_{s,0}\neq m\}\right\rangle\left[X_{[d]}\right]$$

Further we have

$$\operatorname{rank}^{\text{bas+}}\left(\left\langle \sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle\left[X_{[d]}\right]\right) \leq \ell_0(\theta)\ .$$

Proof. Let us notice, that whenever $i \notin \mathcal{M}$ then

$$\sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{[r]\times[m+1]} = l_{[r]\times[m+1]}\right] = 0\left[X_{[d]}\right]\ ,$$

since the condition for non-vanishing coordinates

$$\forall k\in[d], s\in[r] : \left(l_{s,1} = k \wedge l_{s,0}\neq m\right) \Rightarrow \left(l_{s,0} = x_k\right)$$

in Def. 16.7 cannot be satisfied for any $x_{[d]}$. For $i \notin \mathcal{M}$ we have

$$\sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{[r]\times[m+1]} = l_{[r]\times[m+1]}\right] = \left\langle\{\epsilon_{l_{s,0}}\left[X_{l_{s,1}}\right] : s\in[r],\, l_{s,0}\neq m\}\right\rangle\left[X_{[d]}\right]\ .$$

We now use these insights and linearity of contraction to get

$$\left\langle \sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle\left[X_{[d]}\right]$$
$$= \sum_{i\in[n]} \theta\left[L_{[r]\times[m+1]} = l^i_{[r]\times[m+1]}\right]\sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{[r]\times[m+1]} = l_{[r]\times[m+1]}\right]$$
$$= \sum_{i\in\mathcal{M}} \theta\left[L_{[r]\times[m+1]} = l^i_{[r]\times[m+1]}\right]\left\langle\{\epsilon_{l_{s,0}}\left[X_{l_{s,1}}\right] : s\in[r],\, l_{s,0}\neq m\}\right\rangle\left[X_{[d]}\right]$$

This shows the decomposition claim. We notice that this is a basis+ CP decomposition of size $\mathcal{M}$ and thus

$$\operatorname{rank}^{\text{bas+}}\left(\left\langle \sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{[r]\times[m+1]}\right], \theta\left[L_{[r]\times[m+1]}\right]\right\rangle\left[X_{[d]}\right]\right) \leq |\mathcal{M}| \leq \ell_0(\theta)\ .$$

■

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 16.9 *The* CP *selection tensor coincides with a formula selecting neural network with neurons (see Figure 16.2):*

- *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.

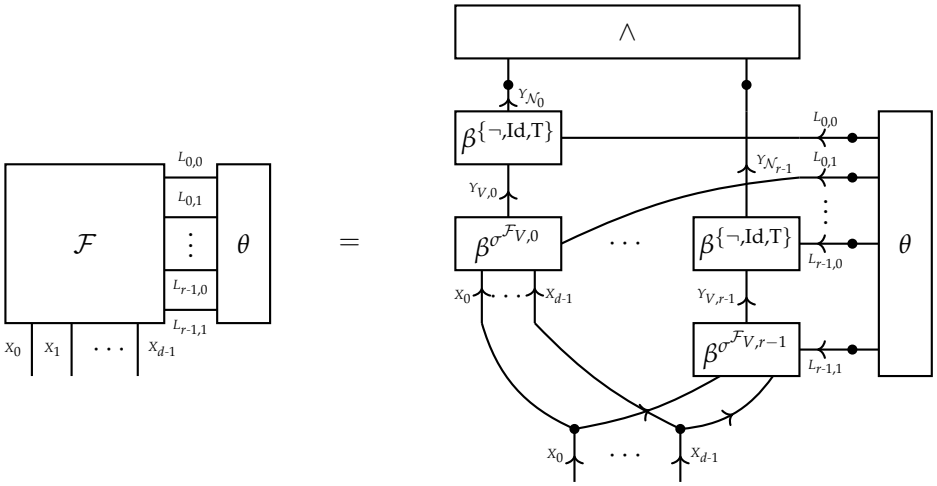


Figure 16.1: Representation of a basis+ Tensor by the contraction of a parameter tensor θ with a CP selecting architecture $\mathcal{F}$, which has a decomposition as a formula selecting neural network (see Lem. 16.9). The nonzero coordinates of θ represent slices of the CP decomposition.

16.2.1 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 16.10 (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 $\epsilon_0, \epsilon_1, \mathbb{I}\,[1]$. 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. $\lozenge$

16.3 Approximation in the Hilbert-Schmidt norm

The Hilbert-Schmidt norm of a tensor is the contraction of the coordinatewise transform with the square function

$$\|\tau\left[X_{[d]}\right]\|_2 = \sqrt{\left\langle \left(\tau\left[X_{[d]}\right]\right)^2 \right\rangle [\varnothing]}\,.$$

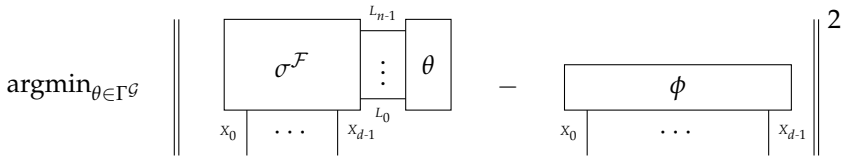
Approximation involving a selection architecture $\mathcal{F}$ is the problem

$$\mathrm{argmin}_{\theta\in\Gamma^{\mathcal{G}}} \|\phi - \left\langle \sigma^{\mathcal{F}}, \theta \right\rangle \left[X_{[d]}\right]\|^2\,.$$

As an example, using the universal selection architecture (corresponding with minterm formulas), the approximation problem is

$$\mathrm{argmin}_{\theta\in\Gamma^{\mathcal{G}}} \|\phi\left[X_{[d]}\right] - \theta\left[X_{[d]}\right]\|^2\,.$$

In a tensor network diagram we depict this as



16.3.1 Approximation for Mode Queries

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 Hybrid Logic Networks and the proposal distribution of grafting.

16.3.1.1 Weighted Squares Loss Trick

Lemma 16.11 *Let τ be a boolean tensor, that is* $\mathrm{im}\,(\tau) \subset \{0,1\}$. *Then*

$$\tau\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] - \left(\tau\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2$$

where $\mathbb{I}$ is a tensor with same shape as τ and all coordinates being 1.

Proof. Since for each $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have $\tau[X_{[d]} = x_{[d]}] \in \{0,1\}$, it holds that

$$\tau\left[X_{[d]} = x_{[d]}\right] = 1 - (\tau\left[X_{[d]} = x_{[d]}\right] - 1)^2$$

and thus in coordinatewise calculus

$$\tau\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] - \left(\tau\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2 .$$

■

We apply this property to reformulate optimization problems over boolean tensors into weighted least squares problems.

Theorem 16.12 (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*

$$\mathrm{argmax}_{\tau\in\Gamma}\,\langle I,\tau\rangle\,[\varnothing] = \mathrm{argmin}_{\tau\in\Gamma}\left\langle I, (\tau\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2\right\rangle[\varnothing]$$

Proof. Using the Lemma above, τ is identical to $\mathbb{I}\left[X_{[d]}\right] - (\tau\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2$ and we get

$$\langle I,\tau\rangle\,[\varnothing] = \left\langle I, \mathbb{I}\left[X_{[d]}\right]\right\rangle[\varnothing] - \left\langle I, (\tau\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2\right\rangle[\varnothing]$$

Since the first term does not depend on τ, it can be dropped in the maximization problem. The (-1) factor then turns the maximization into a minimization problem. ■

Thm. 16.12 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 Thm. 16.12 importance tensor, since it manipulates the relevance of each coordinate in the squares loss.

As a result, we interpret the objective

$$\left\langle I, (\tau\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2 \right\rangle [\varnothing]$$

as a weighted squares loss.

Example 16.13 (Proposal distribution maxima) The Problem $\mathrm{P}^{\mathrm{grad}}_{D,\tilde{\mathbb{P}},\mathcal{H}}$ of finding the maximal coordinate of the proposal distribution can thus be turned into

$$\begin{aligned}&\mathrm{argmax}_{l_{[n]}} \left\langle (\mathbb{P}^D - \tilde{\mathbb{P}}), \mathcal{F} \right\rangle \left[L_{[n]} = l_{[n]}\right] \\ &\quad = \mathrm{argmin}_{l_{[n]}} \left\langle (\mathbb{P}^D - \tilde{\mathbb{P}}), \left(\left\langle \mathcal{F}, \epsilon_{l_{[n]}} \left[L_{[n]}\right] \right\rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2 \right\rangle [\varnothing] \ .\end{aligned}$$

$\Diamond$

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 $\Gamma^{\mathcal{S}^{\mathcal{G}},\mathbb{I}}$ for efficient solutions by alternating algorithms.

However, for any hypergraph $\mathcal{G}$ we have $\mathbb{I}\left[X_{[d]}\right] \in \Gamma^{\mathcal{S}^{\mathcal{G}},\mathbb{I}}$. 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 16.14 (Fitting a boolean tensor by a formula tensor) Given a tensor τ, we want to find a formula $f \in \mathcal{F}$ such that it approximates τ.

If τ 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

$$\mathrm{argmax}_{f \in \mathcal{F}} \langle f \Leftrightarrow \tau \rangle [\varnothing] \ .$$

We can use the squares risk trick and get an equivalent problem

$$\mathrm{argmin}_{f \in \mathcal{F}} \| \langle f \Leftrightarrow \tau \rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right] \|^2 .$$

We have since τ and f are boolean

$$\| \langle f \Leftrightarrow \tau \rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right] \|^2 = \| \langle f \rangle \left[X_{[d]}\right] - \tau\left[X_{[d]}\right] \|^2$$

Now, when representing $\mathcal{F}$ in a formula selecting architecture we have

$$\mathrm{argmin}_{\theta \in \Gamma_1} \| \langle \mathcal{F}, \theta \rangle \left[X_{[d]}\right] - \tau\left[X_{[d]}\right] \|^2 .$$

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 $\tau\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right]$, then the solution would be $\mathbb{I}[L]$.

$\Diamond$

16.3.3 Alternating Solution of Least Squares Problems

When the parameter tensor θ is only restricted to have a decomposition as a tensor network on $\mathcal{G}$, we can iteratively update each core. The resulting algorithm is called Alternating Least Squares (ALS) (see Algorithm 11).

Algorithm 11 Alternating Least Squares (ALS)

Input: Target tensor $\phi[X_{[d]}]$, hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $[d] \subset \mathcal{V}$ specifying the approximation format

Output: Approximation of $\phi[X_{[d]}]$ by a tensor network $\langle \{\tau^e [X_e] : e \in \mathcal{E}\} \rangle \left[X_{[d]}\right]$

for all $e \in \mathcal{E}$ **do**
  Set $\tau^e [X_e]$ to a random element in $\bigotimes_{k \in e} \mathbb{R}^{m_k}$
end for
while Stopping criterion is not met **do**
  for all $e \in \mathcal{E}$ **do**
    Set $\tau^e [X_e]$ to a solution of the local problem, that is

$$\tau^e [X_e] \leftarrow \operatorname{argmin}_{\tau^e[X_e]} \left\langle I, (\langle \mathcal{F}, \theta \rangle \left[X_{[d]}\right] - \phi[X_{[d]}])^2 \right\rangle [\varnothing]$$

  end for
end while
return $\{\tau^e [X_e] : e \in \mathcal{E}\}$

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.

16.3.4 Regularization and Compressed Sensing

When regularizing the least squares problem by enforcing the sparsity of θ, we arrive at the compressed sensing problem

$$\operatorname{argmin}_{\theta[L]} \ell_0(\theta) \quad \text{subject to} \quad \left\| \left\langle \sigma^{\mathcal{S}}, \theta \right\rangle \left[X_{[d]}\right] - \phi \left[X_{[d]}\right] \right\|_2 \leq \eta$$

Here, the sensing matrix is the selection tensor.

Example 16.15 (Formula fitting to an example) Choosing the best formula fitting data (see Example 16.14) is the problem

$$\operatorname{argmin}_{\theta[L] : \ell_0(\theta)=1} \left\| \left\langle I, \sigma^{\mathcal{S}}, \theta \right\rangle \left[X_{[d]}\right] - \phi \right\|_2$$

where I has nonzero entries at marked coordinates and ϕ stores in Boolean coordinates whether the marked coordinates are positive or negative examples. When the number of positive and negative examples are identical, we can linearly transform the objective to that of a grafting instance, where the current model is the empirical distribution of negative examples and the data consists of the positive examples. ♢

The sparse tensor solving the problem then has a small number of nonzero coordinates and the selection tensor can be restricted to those. As a consequence, inference can be performed more efficiently.

The algorithmic solution of these problems can be done by greedy algorithms, thresholding based algorithms or optimization based algorithms [FR13].

Guarantees for the success of the algorithms depend on the properties of the sensing matrices. Here the sensing matrices are deterministic, since constructed as selection tensors, and concentration based approaches towards probabilistic bounds on these properties (see [Goe21]) are not applicable.

Example 16.16 (Sensing matrix for propositional Formulas) Let there be a set $\mathcal{F}$ of formulas, then we have

$$\left\langle \sigma^{\mathcal{F}}\left[X_{[d]}, L_{\text{in}}\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L_{\text{out}}\right]\right\rangle [L_{\text{in}} = l_{\text{in}}, L_{\text{out}} = l_{\text{out}}] = \left\langle f_{l_{\text{in}}}, f_{l_{\text{out}}}\right\rangle [\varnothing] \ .$$

If the formulas have disjoint model sets then

$$\left\langle \sigma^{\mathcal{F}}\left[X_{[d]}, L_{\text{in}}\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L_{\text{in}}\right]\right\rangle [L_{\text{in}} = l_{\text{in}}, L_{\text{out}} = l_{\text{out}}] = \begin{cases} \langle f_{\text{in}}\rangle [\varnothing] & \text{if } l_{\text{in}} = l_{\text{out}} \\ 0 & \text{else} \end{cases} \ .$$

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 $\langle f_{\text{in}}\rangle [\varnothing]$. ♢

Example 16.17 (Sensing matrix for slice selection networks) For the slice selection network

$$\begin{aligned} &\left\langle \sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{\text{in}}\right], \sigma^{\mathcal{F}_{\wedge,d,r}}\left[X_{[d]}, L_{\text{out}}\right]\right\rangle [L_{\text{in}} = l_{\text{in}}, L_{\text{out}} = l_{\text{out}}] \\ &\quad = \begin{cases} 0 & \text{if for a } \tilde{k} \in A^{l_{\text{in}}} \cap A^{l_{\text{out}}} \text{ we have } x^{l_{\text{in}}}_{\tilde{k}} \neq x^{l_{\text{out}}}_{\tilde{k}} \\ \prod_{\tilde{k} \notin A^{l_{\text{in}}} \cup A^{l_{\text{out}}}} m_{\tilde{k}} & \text{else} \end{cases} \ . \end{aligned}$$

Given a fixed l_{in}, the maximum value in the respective slice is thus taken at $l_{\text{in}} = l_{\text{out}}$. ♢

16.4 Discussion

The selection network described here have been taylored to the CP format. We can extend the 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 $\tau^e [X_e]$, where $e \subset \mathcal{V}$ is selected by variable selectors and the values of the tensor τ^e by activation selectors.

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 17.1 (Commutativity of Contractions) *Let $\tau^{\mathcal{G}}$ 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 $\tilde{\mathcal{V}} \subset \mathcal{V}$*

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \tau^{\mathcal{G}_1} [X_{\mathcal{V}^1}] \cup \{ \left\langle \tau^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2 \cap (\mathcal{V}^1 \cup \tilde{\mathcal{V}})} \right] \} \right\rangle [X_{\tilde{\mathcal{V}}}] \ .$$

Proof. For any index $x_{\tilde{\mathcal{V}}}$ we show that

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = \left\langle \tau^{\mathcal{G}_1} \cup \{ \left\langle \tau^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2 \cap (\mathcal{V}^1 \cup \tilde{\mathcal{V}})} \right] \} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \ .$$

By definition we have

$$\begin{aligned}
\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] &= \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}}} \prod_{e \in \mathcal{E}} \tau^e [X_e = x_e] \\
&= \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}}} \left(\prod_{e \in \mathcal{E}_1} \tau^e [X_e = x_e] \right) \cdot \left(\prod_{e \in \mathcal{E}_2} \tau^e [X_e = x_e] \right) \\
&= \sum_{x_{\mathcal{V}^1/\tilde{\mathcal{V}}}} \sum_{x_{\mathcal{V}^2/(\tilde{\mathcal{V}} \cup \mathcal{V}^1)}} \left(\prod_{e \in \mathcal{E}_1} \tau^e [X_e = x_e] \right) \cdot \left(\prod_{e \in \mathcal{E}_2} \tau^e [X_e = x_e] \right) \\
&= \sum_{x_{\mathcal{V}^1/\tilde{\mathcal{V}}}} \left(\prod_{e \in \mathcal{E}_1} \tau^e [X_e = x_e] \right) \cdot \left(\sum_{x_{\mathcal{V}^2/(\tilde{\mathcal{V}} \cup \mathcal{V}^1)}} \prod_{e \in \mathcal{E}_2} \tau^e [X_e = x_e] \right) \ .
\end{aligned}$$

When contracting the variables $X_{\mathcal{V}^2/(\tilde{\mathcal{V}} \cup \mathcal{V}^1)}$ on $\tau^{\mathcal{G}_2}$, the variables $X_{\mathcal{V}^2 \cap (\tilde{\mathcal{V}} \cup \mathcal{V}^1)}$ are left open. We

therefore have for any $x_{\mathcal{V}^2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}^1)}$

$$\left\langle \tau^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}^1)} = x_{\mathcal{V}^2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}^1)} \right] = \left(\sum_{x_{\mathcal{V}^2/(\tilde{\mathcal{V}}\cup\mathcal{V}^1)}} \prod_{e\in\mathcal{E}_2} \tau^e \left[X_e = x_e \right] \right) .$$

It follows with the above, that

$$\begin{aligned} \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] &= \sum_{x_{\mathcal{V}^1/\tilde{\mathcal{V}}}} \left(\prod_{e\in\mathcal{E}_1} \tau^e \left[X_e = x_e \right] \right) \cdot \left\langle \tau^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}^1)} = x_{\mathcal{V}^2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}^1)} \right] \\ &= \left\langle \tau^{\mathcal{G}_1} \cup \{ \left\langle \tau^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2\cap(\mathcal{V}^1\cup\tilde{\mathcal{V}})} \right] \} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] . \end{aligned}$$

■

We can interpret the inner contraction $\left\langle \tau^{\mathcal{G}_2} \right\rangle \left[X_{\mathcal{V}^2\cap(\mathcal{V}^1\cup\tilde{\mathcal{V}})} \right]$ 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 Thm. 17.1 to split a contraction into subcontractions, which are consecutively performed.

Contractions can be performed partially, and the result passed to the rest of the network as a message.

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 17.2 (Cluster Graph) Given a tensor network $\tau^{\mathcal{G}}$ 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 17.3 *Given a tensor network $\tau^{\mathcal{G}}$ 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

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \{ \left\langle \tau^{C_i} \right\rangle \left[X_{\mathcal{V}_i\cap(\tilde{\mathcal{V}}_i\cup\tilde{\mathcal{V}})} \right] : i \in [n] \} \right\rangle [X_{\tilde{\mathcal{V}}}] .$$

Proof. By Thm. 17.1 applied for each cluster seen as a subgraph. ■

17.2.2 Message Passing to calculate Contractions

Having a hypergraph $\mathcal{G}$, we iteratively apply Thm. 17.1 and call the $\mathcal{G}_2$ a cluster. When iterating until $\mathcal{G}$ is empty, we get a cluster graph, where all tensors are assigned to a cluster.

When the cluster are a polytree, that is a union of disjoint trees, we define messages between neighbored clusters C_i and C_j with $C_j \prec C_i$ by the contractions

$$\delta_{j\to i}\left[X_{\mathcal{V}^i\cap\mathcal{V}^j}\right] = \left\langle \{\delta_{\tilde{j}\to j}\left[X_{\mathcal{V}^{\tilde{j}}\cap\mathcal{V}^j}\right] \,:\, C_{\tilde{j}} \prec C_j\} \cup \tau^{C_j}\right\rangle \left[X_{\mathcal{V}^i\cap\mathcal{V}^j}\right] .$$

We note, that the messages are well defined by these recursive equations, exactly when the cluster graph is a polytree.

When the cluster graph is a tree, we can choose a root cluster and order the clusters by the topological order $\prec$.

Lemma 17.4 *When the cluster graph is a tree satisfying the running intersection property, we have for neighbored clusters C_i and C_j with $C_j \prec C_i$*

$$\delta_{i\to\tilde{i}}\left[X_{\mathcal{V}^i\cap\mathcal{V}^{\tilde{i}}}\right] = \left\langle \{\tau^{C_j} \,:\, C_j \prec C_i\}\right\rangle \left[X_{\mathcal{V}^i\cap\mathcal{V}^{\tilde{i}}}\right] .$$

Proof. By induction over the cardinality n of the preceding clusters. $\boldsymbol{n = 1}$: For a single preceding cluster the statement holds trivial, since the preceding cluster is the cluster itself.

$\boldsymbol{n \to n+1}$: 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 Thm. 17.3 on this constructed cluster graph and get

$$\left\langle \{\tau^{C_{\tilde{j}}}\left[X_{\mathcal{V}^{\tilde{j}}}\right] \,:\, \tilde{j} \neq i\}\right\rangle [X_{\mathcal{V}^i}] = \left\langle \left\{ \left\langle \{\tau^{C_{\tilde{j}}}\left[X_{\mathcal{V}^{\tilde{j}}}\right] \,:\, \tilde{j} \prec j\}\right\rangle [X_{\tilde{\mathcal{V}}^j}] \,:\, j \in \mathrm{N}(i)\right\}\right\rangle [X_{\mathcal{V}^i}]$$

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

$$\left\langle \{\tau^{C_{\tilde{j}}}\left[X_{\mathcal{V}^{\tilde{j}}}\right] \,:\, \tilde{j} \prec j\}\right\rangle [X_{\tilde{\mathcal{V}}^j}] = \delta_{j\to i}\left[X_{\mathcal{V}^j\cap\mathcal{V}^i}\right]$$

With the above, we arrive at

$$\delta_{i\to\tilde{i}}\left[X_{\mathcal{V}^i\cap\mathcal{V}^{\tilde{i}}}\right] = \left\langle \{\tau^{C_j} \,:\, C_j \prec C_i\}\right\rangle \left[X_{\mathcal{V}^i\cap\mathcal{V}^{\tilde{i}}}\right] . \qquad \blacksquare$$

Theorem 17.5 *When the cluster graph is a tree satisfying the running intersection property, then we have for each cluster C_i with neighbors $\mathrm{N}(i)$*

$$\left\langle \tau^{\mathcal{G}} \right\rangle \left[X_{\mathcal{V}_i} \right] = \left\langle \{ \delta_{j \to i} \left[X_{\mathcal{V}^i \cap \mathcal{V}^j} \right] \,:\, j \in N(i) \} \cup \{ \tau^{C_i} \} \right\rangle \left[X_{\mathcal{V}_i} \right] .$$

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 Thm. 17.3 and Lem. 17.4.

■

While we have defined message passing along the topological order of a graph, we can also define messages against the topological order, that is

$$\delta_{j \leftarrow i} = \left\langle \{ \delta_{i \leftarrow \tilde{j}} \,:\, C_i \prec C_{\tilde{j}} \} \cup \tau^{C_i} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right]$$

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 [WJ08].

17.2.3 Variable Elimination Cluster Graphs

Remark 17.6 (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 17.1).

Definition 17.7 Given a tensor network $\tau^{\mathcal{G}}$ 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 τ^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. 13.11 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).

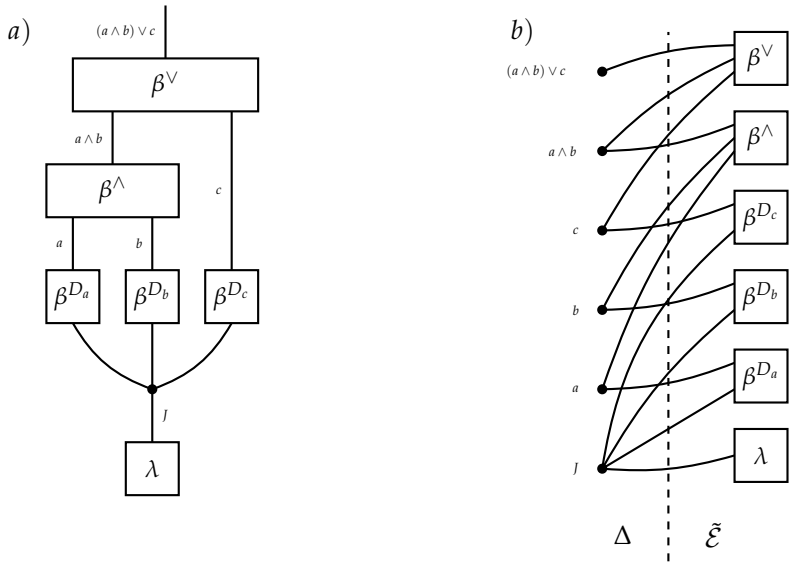


Figure 17.1: Example of a Bethe Cluster Graph. a) Example of a Tensor Network $\tau^{\mathcal{G}}$, which represents the by λ averaged evaluation of the formula $(a \wedge b) \vee c$ on data D. b) Corresponding Bethe Cluster Hypergraph, which dual is bipartite by the sets Δ and $\tilde{\mathcal{E}}$.

17.3 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 Monotonicity of Tensor Contraction

To state the next theorem we use the nonzero function $\mathbb{I}_{\neq 0} : \mathbb{R} \to [2]$ 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. 14.2.

Theorem 17.8 (Monotonicity of Tensor Contractions) *Let $\tau^{\mathcal{G}}, \tau^{\tilde{\mathcal{G}}}$ be tensor network of non-negative tensors and $X_{\tilde{\mathcal{V}}}$ an arbitrary set of random variables. Then we have*

$$\mathbb{I}_{\neq 0}\left(\left\langle \tau^{\mathcal{G}} \cup \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}]\right) \prec \mathbb{I}_{\neq 0}\left(\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}]\right).$$

Proof. It suffices to show that for any $x_{\tilde{\mathcal{V}}}$ with

$$\mathbb{I}_{\neq 0}\left(\left\langle \tau^{\mathcal{G}} \cup \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}]\right) = 1$$

we also have

$$\mathbb{I}_{\neq 0}\left(\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}]\right) = 1.$$

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

$$\left\langle \tau^{\mathcal{G}} \cup \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] > 0$$

and it follows that

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] > 0 \quad \text{and} \quad \left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] > 0 \,.$$

But this already implies, that

$$\mathbb{I}_{\neq 0} \left(\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \right) = 1 \,. \qquad \blacksquare$$

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 Thm. 5.23.

Theorem 17.9 (Invariance under adding subcontractions) *Let $\tau^{\mathcal{G}}$ be a tensor network of non-negative tensors with variables $X_{\mathcal{V}}$ and let $\tau^{\tilde{\mathcal{G}}}$ be a subset. Then we have for any subset $X_{\tilde{\mathcal{V}}}$ of $X_{\mathcal{V}}$*

$$\left\langle \tau^{\mathcal{G}} \cup \{ \mathbb{I}_{\neq 0} \left(\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] \right) \} \right\rangle [X_{\mathcal{V}}] = \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}] \,.$$

Proof. For any $x_{\mathcal{V}}$ with

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0$$

we also have

$$\left\langle \tau^{\mathcal{G}} \cup \{ \mathbb{I}_{\neq 0} \left(\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] \right) \} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0 \,.$$

For any $x_{\mathcal{V}}$ with

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \neq 0$$

we have for the reduction $x_{\tilde{\mathcal{V}}}$ of the index $x_{\mathcal{V}}$ that

$$\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \neq 0$$

and thus

$$\begin{aligned} \left\langle \tau^{\mathcal{G}} \cup \{ \mathbb{I}_{\neq 0} \left(\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] \right) \} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] &= \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \cdot \mathbb{I}_{\neq 0} \left(\left\langle \tau^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] \right) [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \\ &= \left\langle \tau^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \,. \end{aligned}$$

$\blacksquare$

Remark 17.10 Similar statements hold, when dropping the non-negativity assumption on the, but demanding that all variables are left open. $\lozenge$

17.3.3 Basis Calculus as a Message Passing Scheme

Message Passing of directed and boolean message by basis encoding of functions can be interpreted as function evaluation. Each subfunction evaluation is passed in its one-hot encoding.

This is because any basis encoding of a function, the decomposition

$$\beta^q = \sum_{y \in \mathrm{im}(q)} \Big(\sum_{i : q(i) = y} \epsilon_i \Big) \otimes \epsilon_y$$

is a SVD of the matrification of β^q with respect to incoming and outgoing legs.

Passing a message ϵ_i in direction thus gives the message $\epsilon_{q(i)}$.

Note, that this is exact, whenever the graph is directed and acyclic. We do not need acyclicity of the underlying undirected graph.

Remark 17.11 (Basis Calculus as Message Passing) Given a tensor network of directed and binary tensor cores, each representing a function q_e depending on variables e^{in}. When there are not directed cycles, we define the compositions of q_e to be the function q 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

$$\left\langle \{\beta^{q_e}\left[X_{e^{\text{out}}}, X_{e^{\text{in}}}\right] : e = (e^{\text{out}}, e^{\text{in}}) \in \mathcal{E}\} \right\rangle \left[\mathcal{V}^2\right] = \epsilon_{q(x_v : v \in \mathcal{V}^1)} \,.$$

$\lozenge$

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 17.12 (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. $\lozenge$

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, in the version 2.0.0

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 the repository

https://github.com/EnexaProject/enexa-tensor-reasoning

or by

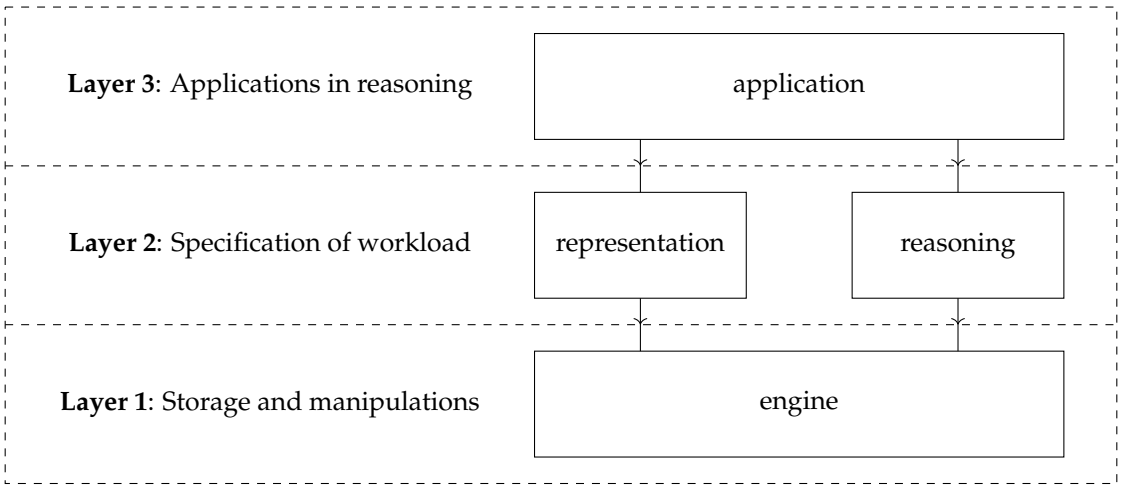
```
!pip install tnreason==2.0.0
```

A.1 Architecture

tnreason is structured in four subpackages and three layers

- Layer 1: Storage and numerical manipulations, by subpackage engine
- Layer 2: Specification of workload, subpackage representation specific for storage, subpackage reasoning specific for manipulations
- Layer 3: Applications in reasoning, by subpackage application

We sketch this structure by



A.2 Implementation of basic notation

First of all, we explain how the basis notation explained in Chapter 2.5 is reflected in the implementation.

A.2.1 Categorical Variables and Representations

Categorical Variables are identified by strings, which then appear as colors of the corresponding tensor axes. Their dimension is stored in shapeDicts, but most practically these shapes are stored in the tensors in which variables appear. Suffixes in the color string (defined in `representation.suffixes`) denote the type of the variable:

- Distributed variables with color suffix *"_dV"*: X.
- Computed variables with color suffix *"_cV"*: Y.
- Selection variables with color suffix *"_sV"*: L.
- Term variables with color suffix *"_tV"*: O.

A.2.2 Tensors

Tensors are objects of classes inheriting `engine.TensorCore` with main attributes

- `values`: Storing the coordinates of the tensors (individual realization for different cores)
- `colors`: List of the variables $[Y_f, X_0, X_1]$
- `name`: Reflecting the notation such as β^f
- `shape`: Storing the dimension of each appearing variable, as a list of integers with the same length as colors.

Suffixes in the name string (defined in `representation.suffixes`) highlight the origin and purpose of the tensor. Cores are named with suffixes based on their functionality

- Computation core with name suffix *"_cC"*: They represent the computation of a function in basis calculus, and are directed cores. Their colors are `[headColors] + [inputColors]`, where `[inputColors]` are either distributed variables or, if having a composition of formulas. When the function is a selection augmentation of other functions, selection colors are listed in the end of `[inputColors]`.
- Activation core with name suffix *"_aC"*: two-dimensional vectors representing of the activation core to a formula

Both the cores and the colors are further refined by infixes before the suffices to denote specific instantiations.

- *"_s"*: Involving a selection variable
- *"_e"*: Storing evidence about a variable
- *"_h"*: Head of a function, typically the variable computed at a activation selector
- *"_f"*: Function selection variables
- *"_p"+"i"*: Variable selection for argument at position i
- *"_d"*: Involving data (data cores and colors)

Further infixes are strings denoting atom names and neuron names.
Exploiting efficient representation tricks we further have the tensor name suffices:

- *"_atoC"*: Atomization core, for sparse representation of categorical constraints
- *"_vselC"*: Variable selection core: For sparse representation of variable selectors

Initialization Tensors are instantiated by

```
engine.getCore(coreType)(values, colors, name, shape)
```

where `coreType` is a string further specifying a specific implementation of tensors (see for more detail Sect. A.3). The default tensor implementation `NumpyCore`is chosen, when `coreType` is not specified.

One-hot encodings are specific tensors created in representation.

A.2.3 Contractions

Tensor networks $\tau^{\mathcal{G}} = \{\tau^e [X_e] : e \in \mathcal{E}\}$ are stored as dictionaries of tensors, where the keys coincide with the names of the corresponding tensors. The edges of the hypergraph $\mathcal{G}$ used in the definition of tensor networks in Def. 0.10 are here labeled by the names of the tensors and the affected variables by the list of colors, being an attribute of each tensor.

Contractions are implemented in the subpackage engine, orienting on Def. 0.12. Reflected in the notation

$$\left\langle \tau^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}]$$

a contraction is defined by

- Tensor Network $\tau^{\mathcal{G}}$, specified by a dictionary of tensor names as keys and valued by tensor cores.
- Open Variables $\tilde{\mathcal{V}}$, specified by a list of colors to the variables.

Contraction calls are implemented as

```
engine.contract(contractionMethod, coreDict, openColors,
 dimensionDict, evidenceColorDict)
```

where the arguments are

- `contractionMethod`: str, chooses one of the contraction providers. The default contraction method `NumpyEinsum`is chosen, when
- `coreDict`: Dictionary of TensorCores (of the above formats), representing the Tensor Network $\tau^{\mathcal{G}}$
- `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

Coordinates of tensors can be retrieved by

$$\langle \tau [X_{\mathcal{V}}] \rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \ .$$

We implement this by leaving `openColors` empty and passing $x_{\tilde{\mathcal{V}}}$ as the `evidenceColorDict`, as a dictionary with keys by the `str` colors to the variables and values by the corresponding `int` indices.

Graphical illustrations can be generated by

```
engine.draw_factor_graph(coreDict)
```

where `coreDict` is a tensor network to be visualized.

A.2.4 Function encoding schemes

Encoding schemes are implemented in the subpackage representation.

A.3 Subpackage engine

The engine subpackage is for the storage and numerical manipulation of tensors and tensor networks. We organize the subpackage as the lowest layer of tnreason, specializing in storage of Tensor Networks and performing the contractions.

A.3.1 Basis+ CP Decompositions storing `values`

Specification of basis+ elementary tensors We orient on basis+ sparse tensor decomposition in the initialization of tensor cores, as discussed in detail in Chapter 15. The basis+ elementary tensors have basis+ CP rank of 1 and admit a decomposition as (see Def. 15.7)

$$\tau\left[X_{\mathcal{V}}\right] = \lambda \cdot \langle \epsilon_{x_A}\left[X_A\right]\rangle \left[X_{\mathcal{V}}\right] .$$

Elementary basis+ tensors are in tnreason stored by a tuple

```
(value, posDict)
```

where `posDict` specifies the values to the variables, which do not have a trivial leg vector, and `value` a scalar scaling the basis vector. Comparing with the notation of Chapter 15, the keys of `posDict` correspond with A, the values of `posDict` with x_A and `value` corresponds with λ.

Elementary Iterators The initialization, coordinate retrieval and conversion operations of all tensor cores are oriented on basis+ CP Decompositions (15.1) of tensors. A tensor

$$\tau\left[X_{\mathcal{V}}\right] = \sum_{(\lambda, A, x_A)\in\mathcal{M}} \lambda \cdot \langle \epsilon_{x_A}\left[X_A\right]\rangle \left[X_{\mathcal{V}}\right] .$$

corresponds with an iterator over tuples `(value,posDict)`, each specifying a basis+ elementary tensor in the sum.

Core Arithmetics When subscribing an instance `exampleCore` of `engine.TensorCore` by

```
exampleCore[posDict] = value
```

a basis+ elementary tensor specified by `(value,posDict)` is added to its values, that is

$$\tau\left[X_{[d]}\right] \leftarrow \tau\left[X_{[d]}\right] + \langle \epsilon_{x_A}\left[X_A\right]\rangle \left[X_{[d]}\right] .$$

The linear structure of tensors spaces are more further reflected in sums of `engine.TensorCore` instances, which are implemented with the same `coreType`, as

```
summed = exampleCore1 + exampleCore2
```

and scalar multiplication, where a scalar `value` of type `int` or `float`

```
multiplied = value * exampleCore
```

Both operations are performed as manipulations of the tensors `values`. Contraction of two `engine.TensorCore` instances are performed by

```
contracted = exampleCore1.contract_with(exampleCore2)
```

and are used in corewise contraction, where `contractionMethod="CorewiseContractor"`.

coreType	Used Package	Storage of `values`	Sparse basis+ CP support
`NumpyCore`	`numpy`	`numpy.array`	No
`PandasCore`	`pandas`	`pandas.DataFrame`	Yes
`TentrisCore`	`tentris` [Big+20]	`tentris.hypertrie`	Yes
`PolynomialCore`	−−	`list` of `(value,posDict)`	Yes

Figure A.1: Derived classes from `engine.TensorCore`, differing in the implemented storage of `values`.

contractionMethod (str)	**Package**	**Applied procedure**
"NumpyEinsum"	`numpy`	Einstein summation `numpy.einsum`
"TentrisEinsum"	`tentris` [Big+20]	Einstein summation `tentris.einsum`
"PgmpyVariableEliminator"	`pgmpy`	Variable Elimination of `pgmpy.DiscreteFactor`
"CorewiseContractor"	−−	Contraction using `core.contract_with()`

Figure A.2: Implemented contraction methods in tnreason.

Initialization Any instance of `engine.TensorCore` is initialized as a vanishing tensor $\mathbb{I}[X_{\mathcal{V}}]$, when `values` is not specified. The values are then assigned by iteration over a `sliceIterator` over `(value,posDict)` tuples specifying elementary basis+ tensors, where the CP rank is the length of the iterator This initialization is by applied in the method

```
engine.create_from_slice_iterator(shape, colors, sliceIterator,
 coreType, name)
```

where `shape, colors, coreType, name` are used in the call of an empty core by `engine.get_core` and `sliceIterator` used to iterative add the basis+ elementary tensors to create the tensor.

Storage of basis+ CP decompositions The implemented tensor classes derived from `engine.TensorCore` differ in their implementation of `values`. Motivated from basis+ CP decompositions, most classes rely on a data base storing the `(value,posDict)` tuples. An overview over the derived classes is provided in Figure A.1. Here `PandasCore`, `TentrisCore` and `PolynomialCore` support sparse basis+ CP decompositions, by utilizing `pandas.DataFrame`, `tentris.hypertrie` and `list` as storage data base. These are implementations of the matrix representation of Remark 15.11. The `NumpyCore` class on the other hand is based relies on arrays as `numpy.array` as storage solution, which corresponds with the demand that each `posDict` contains all colors of the tensor. Effectively, this amounts to restricting to basis CP decomposition, which demanded ranks are always larger than basis+ CP decompositions (see Thm. 15.12).

A.3.2 Contractions

The supported contraction methods are listed in Figure A.2.

Einstein Summation is a syntax of specifying the contractions of arrays. Different possibilities are available to optimize over possible contraction paths, for example in `numpy` by the `numpy.einsum_path`.

$\circ$	$S(\circ)$	**Notes**	$\mathrm{rank}^{\mathrm{bas}+}(\circ)$	$\mathrm{rank}^{\mathrm{bas}+}(\beta^\circ)$
$\wedge$	*"and"*		1	3
$\vee$	*"or"*		2	3
$\Rightarrow$	*"imp"*	last variable as head, others premises	2	3
$\oplus$	*"xor"*	implemented as the negation of *"eq"*, i.e. *"neq"*	3	5
$\Leftrightarrow$	*"eq"*		2	5
X_k	*"pas"* + "k"	kth atom	1	2
$\neg$	*"not"*	negation of the first argument, i.e. *"npas0"*	1	2

Figure A.3: Supported connectives in the script language. The arity of all connectives is not restricted. We notice that the basis+ rank $\mathrm{rank}^{\mathrm{bas}+}(\beta^\circ)$ is independent of the arity and in most cases less than the naive bound of 2^d.

Variable Elimination contracts along a junction tree, build by variable elimination. We here use an implementation in the `pgmpy` package.

Corewise Contraction uses the `contract_with` method of tensor cores to contract in a given order.

A.4 Subpackage representation

The representation subpackage consists in a collection of core creation methods. We arrange the representation subpackage into the second layer of the tnreason architecture, since it specifies tensor cores which formats are specified in engine.

Coordinate Calculus Coordinatewise transformations (see Def. 13.5) are supported by

```
engine.coordinatewise_transform(coresList, transformFunction)
```

where `coresList` is a list of p tensors with identical variables and `transformFunction` a function $h : \mathbb{R}^p \to \mathbb{R}$.

Basis Calculus Basis encodings (see Def. 14.8) of functions $q : \bigtimes_{k\in[d]}[m_k] \to \bigtimes_{l\in[r]}[m_l]$ are created by

```
engine.create_relational_encoding_from_lambda(inshape, outshape,
 incolors, outcolors, indicesToIndices)
```

where `indicesToIndices` is a lambda-function representing q and `inshape, outshape, incolors, outcolors` specify the input and output variables. Let us notice, that this procedure produces sums of $\prod_{k\in[d]} m_k$ basis tensors corresponding with a basis CP decompositions. More involved initialization procedures based on basis+ elementary tensors calling `engine.create_from_iterator` might result in sparser representations.

Propositional Connectives are represented by strings. Figure A.3 lists the supported logical connectives, which are implemented in `representation.basisplus_calculus`. If the `str` to the connective starts with *"n"*, then the negated connective is encoded.

featureType	Purpose	Canonical Parameter	Activation Core
"SingleSoftFeature"	s_l of a statistic	Scalar (`int` or `float`)	$\exp\left[\theta \cdot I_{\mathrm{im}(s_l)}(Y_l)\right]$
"SoftPartitionFeature"	Partition statistics (see Def. 14.32) $\mathcal{S}$	$\theta\,[L]$	$\exp\left[\theta\,[L]\right]$
"HardPartitionFeature"	Partition statistics (see Def. 14.32)	Boolean tensor $\theta\,[L]$	$\theta\,[L]$

Figure A.4: Features derived from `representation.ComputedFeature`, which are implemented in `representation.features`.

Wolfram codes provide a classification scheme of propositional formuals by natural numbers, which is supported in the script language. The Wolfram code has been designed for the classification of cellular automaton rules [Wol83] and popularized in the book [Wol02]. Along this, the coordinate encodings of d-ary connectives $\circ$ are flattened and interpreted as a binary number, which is transformed into a decimal number and represented as a string $S\,(\circ)$. To be more precise, to each d-ary connective its Wolfram code is calculated by

$$N(\circ) = \sum_{x \in [2^d]} 2^{2^d - x - 1} \cdot \circ(x)\,.$$

In the script language, the connective is represented by the string concatenation of the arity and the Wolfram code as

$$S\,(\circ) = "d" + "," + "N(\circ)"\,.$$

A.4.1 Computation Activation Networks

Features are generation procedures of activation cores given canonical parameters. They are initialized by

```
ComputedFeature(featureColors, affectedComputationCores, shape,
 name)
```

where

- `featureColors` is a list of `str` variable colors, which are assigned to a created activation core
- `affectedComputationCores` is a list of `str` names of computation cores, which are required to compute the feature colors
- `shape` is a list of `int` dimensions to the variables

Mean parameters to features are computed by

```
exampleFeature.compute_meanParam(environmentMean)
```

where `environmentMean` is the contraction of a tensor network with open colors by the `featureColors`. They are further capable of computing local changes to canonical parameter in order to match mean parameters by

```
exampleFeature.local_update(environmentMean, meanParam)
```

To customize their purposes, individual feature classes are derived from `representation.ComputedFeature`, as listed in Figure A.4.

Computation Activation Networks are the most general models representable in tnreason. They generalize distributions such as those computable by a statistic (see Def. 2.20) as well as constraint satisfaction problems (see Def. 5.13). Computation Activation Networks are initialized by

```
representation.ComputationActivationNetwork(featureDict,
 computationCoreDict, baseMeasureCoreDict, canParamDict)
```

where

- `featureDict` is a dictionary of features, where the keys correspond with the names of the features
- `computationCoreDict` is a tensor network of computation cores, which needs to contain all names of computation cores required to compute all feature colors
- `baseMeasureCoreDict` is a tensor network representing of $\nu\left[X_{\mathcal{V}}\right]$
- `canParamDict` is a dictionary of canonical parameters (see Figure A.4)

Computation Activation Networks can be instantiated as cores by `exampleCANetwork.create_cores()` or as an energy dictionary by `exampleCANetwork.get_energy_dict()`. Energy dictionaries are stored as dictionaries with values by tuples `value, tensorNetwork`, representing a by `value` weighted sum of the `tensorNetwork`.

Example A.1 (Representation of a member of an exponential family) To represent $\mathbb{P}^{\mathcal{S},\theta,\nu}$ we initialize a

- `featureDict` as a dictionary of `representation.SingleSoftFeature` to each feature s_l
- `computationCoreDict` $\tau^{\mathcal{G}}$ of computation cores such that

$$\tau^{\mathcal{G}}\left[Y_{[p]}, X_{\mathcal{V}}\right] = \beta^{\mathcal{S}}\left[Y_{[p]}, X_{\mathcal{V}}\right] .$$

  Decompositions and redundancies between coordinates s_l can be exploited to find an efficient tensor network.
- `baseMeasureCoreDict` $\tau^{\tilde{\mathcal{G}}}$ to represent the base measure ν
- `canParamDict` of canonical parameters, valued by the coordinates $\theta\left[L = l\right]$

$\Diamond$

A.5 Subpackage reasoning

The reasoning subpackage implements contraction-based reasoning algorithm on representation.ComputationActiva
As the representation 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.5.1 Sampling

Sampling is performed by MCMC methods calling local sampling methods, which are derived classes from `reasoning.SampleCoreBase`.

The energy-based algorithms execute reasoning tasks solely on energy dictionaries, which are created by `representation.ComputationActivationNetwork.get_energy_dict()`.

```
reasoning.EnergyBasedGibbs
```

inferenceMethod (str)	**Applied procedure**	**Dependency**
"ForwardContractor"	Contraction of all cores keeping the feature colors open	−−
"BackwardAlternator"	Iterative local updates to match the mean parameters	Forward inferer to iteratively update the mean parameters
"ExpectationPropagator"	Iterative updates of messages between feature clusters	Forward and backward inferer used for the computation of messages

Figure A.5: Implemented inference methods in tnreason.

A.5.2 Variational Inference

An overview over the variational inference methods in presented in Figure A.5.

Forward mappings are implemented by `reasoning.ForwardContractor` as contraction of all cores, and in `reasoning.ExpectationPropagator` as a message-passing approach.

Backward mappings (see Sect. 3.8) are implemented by `reasoning.BackwardAlternator` as alternating algorithms iteratively updating the canonical parameters to single features (see Algorithm 4).

Mean field methods are approximation methods of energy tensors by tractable exponential families (see Sect. 3.7). Given an energy dictionary the naive mean field method is implemented as

```
reasoning.NaiveMeanField(energyDict)
```

and the more general markov network based mean field method by

```
reasoning.GenericMeanField(energyDict, edgeColorDict)
```

where `edgeColorDict` specifies the graph of the approximating markov network.

A.5.3 Optimization

Optimization is a reasoning task of finding a maximal coordinate given a tensor network. The supported methods are implemented in `reasoning.optimization_handling` and listed in Figure A.6.

A.6 Subpackage application

With the application subpackage we provide an interface for reasoning workload. It builds a third layer, since it used representation to represent knowledge by tensor networks and reasoning in the execution of reasoning tasks. A user-friendly high-level syntax of script language (logical formulas or neuro-symbolic architectures) for the specification of tensor networks creation, such as propositional formulas or categorical constraints, is introduced. 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.

optimizationMethod (str)	**Package**	**Applied procedure**
"numpyArgMax"	`numpy`	Transformation into a numpy core and solution by `numpy.argmax`
"gurobi"	`gurobipy`	Transformation into an ILP and solution by `gurobipy.optimize`
"gibbsSample"	−−	Simulated annealing based on gibbs sampling
"meanFieldSample"	−−	Mean field approximation combined with *"gibbsSample"*

Figure A.6: Implemented optimization methods in tnreason.

A.6.1 Representation of formulas

Propositional formulas f are represented in three schemes:

- Syntactical representation by a script language $S(f)$ as nested lists (see Sect. A.6.2).
- Syntactical representation by a `str` specifying a color to the categorical variables Y_f.
- Representation of formulas by tensor networks being contracted to $\beta^f \left[Y_f, X_{\mathcal{V}} \right]$

Conversions of the formats:

- $S(f)$ to color by

  ```
  application.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

  ```
  application.create_raw_cores(S(f))
  ```

  This creates the connective cores for the semantic representation of β^f. We encode them by iterative calls of `engine.create_from_iterators`.

A.6.2 Script Language

To specify propositional sentences, neuro-symbolic architectures and Markov Logic Networks, we developed a script language.

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 are represented by nested lists, where each sublist is either specifying an atomic formula (if string) or another composed formula. For example, a formula $f_1 \circ, f_2$ is represented by

d-ary connective	*"and"* | *"or"* | *"imp"* | *"xor"* | *"eq"* | *"not"* | $\cdots$ *"d"* + "_" + *"N"*, where $N < 2^{2^d-1}$
Atomic Formula	Set of strings not in Connectives
Complex Formula	Atomic Formula | [d-ary connective, d Complex Formulas]

Figure A.7: Backus-Naur form of the grammar producing the nested list expressions. The string connectives are either appearing in the list of Figure A.3, or represented by a Wolfram code.

$$S(f_1\circ, f_2) = [S(\circ), S(f_1), S(f_2)]$$

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 nested lists follows a grammar, which is provided in Figure A.7 in its Backus-Naur form.

Example A.2 (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"*]]

$\Diamond$

Knowledge Bases
We distinguish here formulas, with propositional logic interpretation and formulas which have a soft logic interpretation. The formulas with hard interpretation are called facts in a knowledge base $\mathcal{KB}$ and encoded by dictionaries

$$\{\text{key}(f) : S(f) \text{ for } f \in \mathcal{KB}\}$$

Markov Logic Networks
The formulas with soft interpretation are called weighted formulas and encoded by $\exp\left[\theta_f \cdot f\right]$. We thus require a specification of the weights, which we do by adding θ_f as a float or an int to the list $S(f)$. We then store Markov Logic Networks by dictionaries

$$\{\text{key}(f) : S(f) + [\theta_f] \text{ for } f \in \mathcal{F}\}$$

Neuro-Symbolic Architecture by Nested Lists
To specify neuro-symbolic architectures in terms of formula selecting maps, as has been the subject of Chapter 7 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. 7.2 and 7.7). More formally, the production rules are formalized in Figure A.8 by the extension of the Backus-Naur form in Figure A.7.

d-ary connectives	[d-ary connective] | [d-ary connective] + d-ary connectives
Dependency Choice	Atomic Formula | Neuron
Dependency Choices	[Dependency Choice] | [Dependency Choice] + Dependency Choices
Neuron	[d-ary connectives, d Dependency Choices]

Figure A.8: Extension of the grammar in Backus-Naur form in Figure A.7 to describe selections of functions.

Connective selectors (see Def. 7.2) are encoded by the list

$$S(\circ) = [S(\circ_0), \ldots, S\left(\circ_{p\text{-}1}\right)]$$

and a formula selector (see Def. 7.7) by

$$S(\mathcal{F}) = [S(\circ_0), \ldots, S\left(\circ_{p\text{-}1}\right)]$$

A logical neuron of order n (see Def. 7.8), defined by a connective selector $\circ$, and a formula selector $\mathcal{F}_k$ on each argument $k \in [n]$, is encoded by

$$S(\mathcal{N}) = [S(\circ), S(\mathcal{F}_0), \ldots, S(\mathcal{F}_{n-1})]$$

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 $S(\mathcal{N})$ is the encoding of a neuron, then any element $x \in S(\mathcal{N})$ 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

$$S(\mathcal{A}) = \{\text{key}(\mathcal{N}) : S(\mathcal{N}) \text{ for } f \in \mathcal{A}\}$$

where key($\mathcal{N}$) 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 a well-defined architecture.

Example A.3 (Neuro-Symbolic Architecture for the Wet Street) Following the wet street example, we can define a neuron by

$$S(\mathcal{N}) = [[\text{"imp"},\text{"eq"}],[\text{"Wet"},\text{"Sprinkler"}],[\text{"Street"}]]$$

from which the formulas

["imp", "Wet", "Street"]
["eq", "Wet", "Street"]
["imp", "Sprinkler", "Street"]
["eq", "Sprinkler", "Street"]

can be chosen. Combining this neuron with further neurons, e.g. by the architecture

$$S(\mathcal{A}) = \{\ \text{"neur1"}: [[\text{"imp"},\text{"eq"}],[\text{"neur2"}],[\text{"Street"}]]\ ,$$
$$\text{"neur2"}: [[\text{"lnot"},\text{"id"}],[\text{"Wet"},\text{"Sprinkler"}],[\text{"Street"}]]\ \}$$

the expressitivity increases. In this case, the further neuron provides the flexibility of the first atoms to be replaced by its negation. ♢

A.6.3 Distributions

Distributions are procedures to specify `representation.ComputationActivationNetworks` and are derived from the base class `representation.DistributionBase`. Each distribution needs to have a routine

```
exampleDistribution.create_caNetwork()
```

creating the corresponding network.

Markov Networks are distributions, which do not have computation cores. The positive coordinates of the factors are represented by `representation.SoftPartitionFactors` and the support of the factors `representation.HardPartitionFactors`.

Empirical Distributions are special instances of Markov Networks specifying distributions of sample data. We represent the values as a CP Format of data cores as specified in Sect. 3.3.1. They are initialized by

```
application.get_empirical_distribution(sampleDf, atomColumns,
 interpretation, dimensionsDict)
```

where

- `sampleDf` is a `pandas.DataFrame` specifying the data
- `atomColumns` is a list of column names in `sampleDf` to be extracted as variables of the distribution.
- `interpretation` is either *"atomic"* or *"categorical"*, specifying whether the entries in `sampleDf` are interpreted as uncertainties in the interval $[0, 1]$, or as assignments to

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.

```
application.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 $S(f)$

A.6.4 Inference

To simplify deductive inference on models a class

```
application.InferenceProvider
```

taking a `representation.ComputationActivationNetwork` or `application.Distribution` has been implemented.

Probabilistic queries as specified Def. 3.1) by

```
.query(variableList, evidenceDict)
```

Mode queries by

```
.exact_map_query()
```

Entailment from the distribution (Def. 5.8) 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.6.5 Learning

To learn instances of `application.HybridKnowledegBase` on data the class

```
application.HybridLearner
```

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

Formula Selecting Neural Networks (Def. 7.8) are specified to define a proposal distribution. They are encoded by creating all formula selecting neurons, each involving a

- a connective selection map (Def. 7.2)
- variable selection cores (Def. 7.3) to each argument using the decomposition of Thm. 7.4.

Each selection variable of each neuron comes with a control variable with suffix *"_sV"*.

Structure Learning is performed by

```
application.HybridLearner.propose_candidate()
```

where a proposal distribution is instantiated and then sampled given the specified inference method.

Weight Estimation is performed by

```
application.HybridLearner.infer_weights_on_data(empDistribution)
```

where `empDistribution` is used to infer the mean parameters to be matched by the canonical parameters of `knowledgeBase.weightedFormulas`.

B

Glossary

B.1 Tensors

Small greek letters are reserved for the notation of tensors:

Notation	Name	Reference
$\alpha^{\theta}\left[Y_{[p]}\right]$	Soft Activation Tensor	Thm. 2.24
$\alpha^{l,\theta}\left[Y_l\right]$	Leg of Soft Activation Tensor	Thm. 2.24
$\beta^{q}\left[Y_q, X_{[d]}\right]$	Basis Encoding of a function q	Def. 14.8
$\delta^{[d],m}\left[X_{[d]}\right]$	Diracs delta	Example 13.10
$\epsilon_{x_{[d]}}\left[X_{[d]}\right]$	One-hot Encoding	Def. 0.7
$\eta^{\mathcal{S},\mathbb{P}^*,m}[L]$	Noise tensor	Def. 10.3
$\theta\left[L\right]$	Canonical Parameter	Def. 2.21
$\kappa^{e}\left[X_e\right]$	Knowledge core	Def. 5.22
$\lambda[I]$	Scalar core in CP decompositions	Def. 15.1
$\mu\left[L\right]$	Mean Parameter	Def. 2.26
$\nu\left[X_{[d]}\right]$	Boolean base measure	Sect. 2.1.2
$\rho^{k}\left[X_k, I\right]$	Leg core in CP decompositions	Def. 15.1
$\sigma^{q}\left[X_{[d]}, L\right]$	Selection Encoding of a vector-valued function q	Def. 0.22
$\tau^{\mathcal{G}}\left[X_{[d]}\right]$	Tensor network of tensors $\tau\left[X_{[d]}\right]$	Def. 0.10
$\xi^{(A,y_A,\theta)}\left[Y_{[p]}\right]$	Activation tensor of a Hybrid Logic Network	Def. 8.10
$\phi\left[X_{[d]}\right]$	Energy tensor	Def. 2.21
$\chi^{q}\left[X_{[d]}\right]$	Coordinate Encoding of a function q	Def. 13.2, often abbreviated by q

Sets of tensors are represented by large greek letters:

Notation	Name	Reference
$\Gamma^{\mathcal{S},\nu}$	Exponential family	Def. 2.21
$\Lambda^{\mathcal{S},\mathcal{G}}$	Sets of by $\mathcal{S}$ and $\mathcal{G}$ computable distributions	Def. 2.20
$\Lambda^{\delta,\mathrm{MAX},\nu}$	Distributions realizable with base measure ν	Def. 2.2
$\mathcal{M}_{\mathcal{S},\nu}$	Polytope of mean parameters	Def. 2.26

In the implementation in tnreason,we distinguish between computation and activation cores. The coarse roles are the computation of a function using basis calculus and the activation of the prepared variable to shape a probability distribution.

Name	Notation	String Suffix
Computation Core	$\beta^{\cdot}$	"_cC"
Activation Core	$\alpha^{\cdot}$, $\kappa^{\cdot}$	"_aC"

B.2 Variables

Variables are denoted by large latin letters, their indices by the corresponding small letters. We distinguish between the coarse types of variables:

Name	Notation	String Suffix
Distributed Variable	$X.$	"_dV"
Computed Variable	$Y.$	"_cV"
Selection Variable	$L.$	"_sV"
Term Variable	$O.$	"_tV"

B.3 Maps

We have in this work encountered different maps, which have been encoded as tensors. Note, that in order to ease the notation, when not specified otherwise the coordinate encoding $\chi^{\cdot}$ has been used.

Notation	Name	Domain	Range	Reference
f	Propositional formula	$\bigtimes_{k\in[d]}[2]$	$\{0,1\}$	Def. 4.2
$\mathcal{F}$	Boolean statistic	$\bigtimes_{k\in[d]}[2]$	$\bigtimes_{f\in\mathcal{F}}[2]$	Def. 8.1
$\mathcal{F}_C$	d-ary connective selecting map	$\bigtimes_{k\in[d]}[2]$	$\bigtimes_{l\in[p_C]}[2]$	Def. 7.2
$\mathcal{F}_V$	Variable selecting map	$\bigtimes_{l\in[p_V]}[2]$	$\bigtimes_{l\in[p_V]}[2]$	Def. 7.3
$\mathcal{F}_S$	State selection map	$[m]$	$\bigtimes_{k\in[d]}[2]$	Def. 7.5
$\mathcal{KB}$	Knowledge Base (conjunction of formulas)	$\bigtimes_{k\in[d]}[2]$	$\{0,1\}$	
$\mathbb{P}\left[X_{[d]}\right]$	Probability distribution	$\bigtimes_{k\in[d]}[m_k]$	$[0,1]$	Def. 2.1
q	Function between states of factored representations	$\bigtimes_{k\in[d]}[m_k]$	$\bigtimes_{l\in[r]}[m_l]$	

B.4 Contraction equations

We here provide a summary for the application of contractions and normalization in probabilistic and logical reasoning.

Concept	**Contraction Equation**	**Reference**
Marginal probability	$\mathbb{P}\left[X_0\right] = \langle \mathbb{P} \rangle \left[X_0\right]$	Def. 2.4
Conditional probability	$\mathbb{P}\left[X_0 \middle| X_1\right] = \langle \mathbb{P} \rangle \left[X_0 | X_1\right]$	Def. 2.6
Markov Network Distribution	$\mathbb{P}^{\tau^{\mathcal{G}}} = \left\langle \tau^{\mathcal{G}} \right\rangle \left[\mathcal{V} | \varnothing\right]$	Def. 2.41
Partition Function	$\mathcal{Z}\left(\tau^{\mathcal{G}}\right) = \left\langle \tau^{\mathcal{G}} \right\rangle \left[\varnothing\right]$	Def. 2.41
Independence of X_0 and X_1	$\langle \mathbb{P} \rangle \left[X_0, X_1\right] = \langle \mathbb{P} \rangle \left[X_0\right] \otimes \langle \mathbb{P} \rangle \left[X_1\right]$	Def. 2.11, Thm. 2.12
Independence of X_0 and X_1 conditioned on X_2	$\langle \mathbb{P} \rangle \left[X_0, X_1 | X_2\right] = \langle \mathbb{P} \rangle \left[X_0 | X_2\right] \otimes \langle \mathbb{P} \rangle \left[X_1 | X_2\right]$	Def. 2.13, Thm. 2.14

Bibliography

[Aba+16] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. arXiv:1603.04467 [cs]. Mar. 2016. DOI: 10.48550/arXiv.1603.04467. URL: http://arxiv.org/abs/1603.04467 (visited on 04/02/2025).

[AH08] 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 (visited on 03/29/2025).

[Ant+12] Grigoris Antoniou et al. *A Semantic Web Primer, third edition*. English. third edition. Cambridge (Mass.): The MIT Press, Aug. 2012. ISBN: 978-0-262-01828-9.

[AZ99] Artur S. Avila Garcez and Gerson Zaverucha. "The Connectionist Inductive Learning and Logic Programming System". en. *Applied Intelligence* 11.1 (July 1999), 59–77. ISSN: 1573-7497. DOI: 10.1023/A:1008328630915. URL: https://doi.org/10.1023/A:1008328630915 (visited on 04/02/2025).

[Bad+22] Samy Badreddine et al. "Logic Tensor Networks". *Artificial Intelligence* 303 (Feb. 2022), 103649. ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103649. URL: https://www.sciencedirect.com/science/article/pii/S0004370221002009 (visited on 11/20/2023).

[BAH19] Ivana Balazevic, Carl Allen, and Timothy Hospedales. "TuckER: Tensor Factorization for Knowledge Graph Completion". en. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019). 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, 5184–5193. DOI: 10.18653/v1/D19-1522. URL: https://www.aclweb.org/anthology/D19-1522 (visited on 04/02/2025).

[Bar16] Albert-László Barabási. *Network Science*. English. Illustrated edition. Cambridge: Cambridge University Press, July 2016. ISBN: 978-1-107-07626-6.

[Bel61] Richard E. Bellman. *Adaptive Control Processes*. en. Publication Title: Adaptive Control Processes. New Jersey: Princeton University Press, 1961. ISBN: 978-1-4008-7466-8. (Visited on 02/07/2021).

[BM05] Gregory Beylkin and Martin J. Mohlenkamp. "Algorithms for Numerical Analysis in High Dimensions". en. *SIAM Journal on Scientific Computing* 26.6 (Jan. 2005), 2133–2159. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/040604959. (Visited on 12/11/2019).

[Big+20] Alexander Bigerl et al. "Tentris – A Tensor-Based Triple Store". en. *The Semantic Web – ISWC 2020*. Ed. by Jeff Z. Pan et al. Vol. 12506. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, 56–73. 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 (visited on 04/02/2025).

[CKP13] Peter G. Casazza, Gitta Kutyniok, and Friedrich Philipp. "Introduction to Finite Frame Theory". en. *Finite Frames: Theory and Applications*. Ed. by Peter G. Casazza and Gitta Kutyniok. Boston: Birkhäuser, 2013, 1–53. ISBN: 978-0-8176-8373-3. DOI: 10.1007/978-0-8176-8373-3_1. URL: https://doi.org/10.1007/978-0-8176-8373-3_1 (visited on 02/04/2025).

[Cha+12] Venkat Chandrasekaran et al. "The Convex Geometry of Linear Inverse Problems". en. *Foundations of Computational Mathematics* 12.6 (Dec. 2012), 805–849. ISSN: 1615-3375, 1615-3383. DOI: 10.1007/s10208-012-9135-7. (Visited on 08/01/2019).

[Cic14] Andrzej Cichocki. "Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions". *arXiv:1403.2048 [cs]* (Mar. 2014). arXiv: 1403.2048. (Visited on 03/16/2019).

[Cic+15] Andrzej Cichocki et al. "Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis". *IEEE Signal Processing Magazine* 32.2 (Mar. 2015). Conference Name: IEEE Signal Processing Magazine, 145–163. ISSN: 1558-0792. DOI: 10.1109/MSP.2013.2297439.

[CH71] P. Clifford and J. M. Hammersley. "Markov fields on finite graphs and lattices". English. *Unpublished* (1971). Publisher: University of Oxford. URL: https://ora.ox.ac.uk/objects/uuid:4ea849da-1511-4578-bb88-6a8d02f457a6 (visited on 03/11/2025).

[CYM20] William Cohen, Fan Yang, and Kathryn Rivard Mazaitis. "TensorLog: A Probabilistic Database Implemented Using Deep-Learning Infrastructure". en. *Journal of Artificial Intelligence Research* 67 (Feb. 2020), 285–325. ISSN: 1076-9757. DOI: 10.1613/jair.1.11944. URL: https://jair.org/index.php/jair/article/view/11944 (visited on 02/20/2024).

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. English. 2nd edition. Hoboken, N.J: Wiley-Interscience, Sept. 2006. ISBN: 978-0-471-24195-9.

[DeG16] Morris H. DeGroot. *Probability and Statistics*. English. PEARSON INDIA, Jan. 2016. ISBN: 978-93-325-7387-1.

[DN21] Caglar Demir and Axel-Cyrille Ngonga Ngomo. "DRILL- Deep Reinforcement Learning for Refinement Operators in ALC". eng. *CoRR* abs/2106.15373 (2021). URL: https://ris.uni-paderborn.de/record/25217 (visited on 11/06/2023).

[Esp+12] Mike Espig et al. "Variational calculus with sums of elementary tensors of fixed rank". en. *Numerische Mathematik* 122.3 (Nov. 2012), 469–488. ISSN: 0945-3245. DOI: 10.1007/s00211-012-0464-x. (Visited on 10/13/2021).

[FH12] Antonio Falco and Wolfgang Hackbusch. "On Minimal Subspaces in Tensor Representations". *Foundations of Computational Mathematics* 12 (Dec. 2012), 765–803. DOI: 10.1007/s10208-012-9136-6.

[FR13] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. en. Applied and Numerical Harmonic Analysis. Birkhäuser Basel, 2013. ISBN: 978-0-8176-4947-0. DOI: 10.1007/978-0-8176-4948-7. URL: https://www.springer.com/de/book/9780817649470 (visited on 12/11/2019).

[Fou25] Python Software Foundation. *Python Language Reference, version 3.13.2*. Apr. 2025. URL: https://docs.python.org/3/ (visited on 03/06/2025).

[Gal+13] Luis Antonio Galárraga et al. "AMIE: association rule mining under incomplete evidence in ontological knowledge bases". en. *Proceedings of the 22nd international conference on World Wide Web*. Rio de Janeiro Brazil: ACM, May 2013, 413–422. ISBN: 978-1-4503-2035-1. DOI: 10.1145/2488388.2488425. URL: https://dl.acm.org/doi/10.1145/2488388.2488425 (visited on 11/06/2023).

[Gan+08] Varun Ganapathi et al. “Constrained approximate maximum entropy learning of Markov random fields”. *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*. UAI’08. Arlington, Virginia, USA: AUAI Press, July 2008, 196–203. ISBN: 978-0-9749039-4-1. (Visited on 01/29/2025).

[Gar+19] Artur d’Avila Garcez et al. *Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning*. arXiv:1905.06088 [cs]. May 2019. DOI: 10.48550/arXiv.1905.06088. URL: http://arxiv.org/abs/1905.06088 (visited on 04/02/2025).

[Gel25] Patrick Gelß. *PGelss/scikit_tt*. original-date: 2018-11-23T13:10:49Z. June 2025. URL: https://github.com/PGelss/scikit_tt (visited on 06/23/2025).

[Gel+19] Patrick Gelß et al. “Multidimensional Approximation of Nonlinear Dynamical Systems”. *Journal of Computational and Nonlinear Dynamics* 14.6 (Apr. 2019), 061006–061006–12. ISSN: 1555-1415. DOI: 10.1115/1.4043148. (Visited on 05/01/2019).

[GT19] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. Englisch. MIT Press, Sept. 2019. ISBN: 978-0-262-53868-8.

[Gil07] Rafael Gillmann. “0/1-Polytopes: Typical and Extremal Properties”. English. PhD thesis. Feb. 2007. URL: https://depositonce.tu-berlin.de/items/urn:nbn:de:kobv:83-opus-14695 (visited on 03/04/2025).

[Gio17] Vincenzo Nicosia Giovanni Russo Vito Latora. *Complex Networks: Principles, Methods and Applications. With 58 exercises*. English. Cambridge, United Kingdom ; New York, NY: Cambridge University Press, Sept. 2017. ISBN: 978-1-107-10318-4.

[Gla+19] Ivan Glasser et al. “Expressive power of tensor-network factorizations for probabilistic modeling”. en. *Advances in Neural Information Processing Systems* 32 (2019). (Visited on 07/06/2021).

[Goe+20] Alex Goeßmann et al. “Tensor network approaches for data-driven identification of non-linear dynamical laws”. en. *Advances in Neural Information Processing Systems - First Workshop on Quantum Tensor Networks in Machine Learning*. 2020, 21.

[Goe21] Alex Christoph Goeßmann. “Uniform Concentration of Tensor and Neural Networks: An Approach towards Recovery Guarantees”. en. Accepted: 2021-12-30T15:00:58Z. PhD Thesis. Berlin: Technische Universität Berlin, 2021. URL: https://depositonce.tu-berlin.de/handle/11303/15990 (visited on 01/13/2022).

[Gra10] Lars Grasedyck. “Hierarchical Singular Value Decomposition of Tensors”. *SIAM J. Matrix Analysis Applications* 31 (Jan. 2010), 2029–2054. DOI: 10.1137/090764189.

[HK09] W. Hackbusch and S. Kühn. “A New Scheme for the Tensor Representation”. en. *Journal of Fourier Analysis and Applications* 15.5 (Oct. 2009), 706–722. ISSN: 1531-5851. (Visited on 06/18/2021).

[Hac12] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. en. Springer Series in Computational Mathematics. Berlin Heidelberg: Springer-Verlag, 2012. ISBN: 978-3-642-28026-9. DOI: 10.1007/978-3-642-28027-6. (Visited on 01/30/2020).

[HL93] Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. English. 1993rd edition. Berlin, Heidelberg: Springer, Oct. 1993. ISBN: 978-3-540-56852-0.

[Hit27] Frank L. Hitchcock. “The Expression of a Tensor or a Polyadic as a Sum of Products”. en. *Journal of Mathematics and Physics* 6.1-4 (1927), 164–189. ISSN: 1467-9590. DOI: https://doi.org/10.1002/sapm192761164. (Visited on 02/09/2021).

[Hoc22] Sepp Hochreiter. “Toward a broad AI”. en. *Communications of the ACM* 65.4 (Jan. 2022), 56–57. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3512715. URL: https://dl.acm.org/doi/10.1145/3512715 (visited on 02/22/2024).

[Hog+21] Aidan Hogan et al. *Knowledge Graphs*. English. 1st edition. Cham: Springer, Nov. 2021. ISBN: 978-3-031-00790-3.

[HRS12] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. "On manifolds of tensors of fixed TT-rank". en. *Numerische Mathematik* 120.4 (Apr. 2012), 701–731. ISSN: 0029-599X, 0945-3245. DOI: 10.1007/s00211-011-0419-7. URL: http://link.springer.com/10.1007/s00211-011-0419-7 (visited on 12/18/2019).

[Jou+23] Norm Jouppi et al. "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings". *Proceedings of the 50th Annual International Symposium on Computer Architecture*. ISCA '23. New York, NY, USA: Association for Computing Machinery, June 2023, 1–14. ISBN: 979-8-4007-0095-8. DOI: 10.1145/3579371.3589350. URL: https://dl.acm.org/doi/10.1145/3579371.3589350 (visited on 04/02/2025).

[KB09] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". en. *SIAM Review* 51.3 (Aug. 2009), 455–500. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/07070111X. (Visited on 01/28/2021).

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. English. 1. edition. Cambridge, Mass.: The MIT Press, July 2009. ISBN: 978-0-262-01319-2.

[Kou+22] N'Dah Jean Kouagou et al. *Neural Class Expression Synthesis*. en. arXiv:2111.08486 [cs]. Dec. 2022. URL: http://arxiv.org/abs/2111.08486 (visited on 11/06/2023).

[Kou+23] N'Dah Jean Kouagou et al. "Neural Class Expression Synthesis". en. *The Semantic Web*. Ed. by Catia Pesquita et al. Vol. 13870. Series Title: Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023, 209–226. ISBN: 978-3-031-33454-2 978-3-031-33455-9. DOI: 10.1007/978-3-031-33455-9_13. URL: https://link.springer.com/10.1007/978-3-031-33455-9_13 (visited on 11/06/2023).

[Lan11] J. Landsberg. *Tensors: Geometry and Applications*. en. Vol. 128. Graduate Studies in Mathematics. American Mathematical Society, Dec. 2011. ISBN: 978-0-8218-6907-9 978-0-8218-8481-2 978-0-8218-8483-6 978-1-4704-0923-4. (Visited on 02/10/2021).

[Leh+11] Jens Lehmann et al. "Class expression learning for ontology engineering". *Journal of Web Semantics* 9.1 (Mar. 2011), 71–81. ISSN: 1570-8268. DOI: 10.1016/j.websem.2011.01.001. URL: https://www.sciencedirect.com/science/article/pii/S1570826811000023 (visited on 11/06/2023).

[Mac03] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Englisch. Illustrated Edition. Cambridge: Cambridge University Press, Sept. 2003. ISBN: 978-0-521-64298-9.

[Mac77] Alan K. Mackworth. "Consistency in networks of relations". *Artificial Intelligence* 8.1 (Feb. 1977), 99–118. ISSN: 0004-3702. DOI: 10.1016/0004-3702(77)90007-8. URL: https://www.sciencedirect.com/science/article/pii/0004370277900078 (visited on 03/29/2025).

[Mar+24] Giuseppe Marra et al. "From statistical relational to neurosymbolic artificial intelligence: A survey". *Artificial Intelligence* 328 (Mar. 2024), 104062. ISSN: 0004-3702. DOI: 10.1016/j.artint.2023.104062. URL: https://www.sciencedirect.com/science/article/pii/S0004370223002084 (visited on 02/20/2024).

[McC59] John McCarthy. "Programs with Common Sense". *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*. London: Her Majesty's Stationary Office, 1959, 75–91. URL: http://www-formal.stanford.edu/jmc/mcc59.html (visited on 04/02/2025).

[Mot36] Theodore S. Motzkin. “Beiträge zur Theorie der linearen Ungleichungen”. ger. Book Title: Beiträge zur Theorie der linearen Ungleichungen. PhD thesis. Jerusalem: Buchdruckerei Azriel, 1936.

[MD94] Stephen Muggleton and Luc De Raedt. “Inductive logic programming: Theory and methods”. *The Journal of Logic Programming* 19 (1994). Publisher: Elsevier, 629–679. URL: https://www.sciencedirect.com/science/article/pii/0743106694900353 (visited on 11/06/2023).

[Mur22] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. English. Cambridge, Massachusetts London, England: The MIT Press, Mar. 2022. ISBN: 978-0-262-04682-4.

[NTK11] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. “A three-way model for collective learning on multi-relational data”. *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML’11. Madison, WI, USA: Omnipress, June 2011, 809–816. ISBN: 978-1-4503-0619-5. (Visited on 04/02/2025).

[Nic+16] Maximilian Nickel et al. “A Review of Relational Machine Learning for Knowledge Graphs”. en. *Proceedings of the IEEE* 104.1 (Jan. 2016), 11–33. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2015.2483592. URL: https://ieeexplore.ieee.org/document/7358050/ (visited on 11/06/2023).

[Nik+22] Goran S. Nikolić et al. “A Survey of Three Types of Processing Units: CPU, GPU and TPU”. *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*. June 2022, 1–6. DOI: 10.1109/ICEST55168.2022.9828625. URL: https://ieeexplore.ieee.org/document/9828625 (visited on 04/02/2025).

[Orú19] Román Orús. “Tensor networks for complex quantum systems”. en. *Nature Reviews Physics* 1.9 (Sept. 2019), 538–550. ISSN: 2522-5820. DOI: 10.1038/s42254-019-0086-7. (Visited on 07/05/2021).

[OT09] 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 (Jan. 2009). Publisher: Society for Industrial and Applied Mathematics, 3744–3759. ISSN: 1064-8275. DOI: 10.1137/090748330. (Visited on 12/12/2020).

[Pas+19] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. arXiv:1912.01703 [cs]. Dec. 2019. DOI: 10.48550/arXiv.1912.01703. URL: http://arxiv.org/abs/1912.01703 (visited on 04/02/2025).

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Englisch. s.l.: Morgan Kaufmann, Sept. 1988. ISBN: 978-1-55860-479-7.

[Pea09] Judea Pearl. *Causality: Models, Reasoning and Inference. Ausgezeichnet: ACM Turing Award for Transforming Artificial Intelligence 2011*. Englisch. 2nd ed. Cambridge New York, NY Port Melbourne New Delhi Singapore: Cambridge University Press, Nov. 2009. ISBN: 978-0-521-89560-6.

[Pen87] Roger Penrose. *Spinors and Space-Time: Volume 1, Two-Spinor Calculus and Relativistic Fields*. Englisch. Cambridge: Cambridge University Press, Feb. 1987. ISBN: 978-0-521-33707-6.

[Per+07] D. Perez-Garcia et al. “Matrix product state representations”. *Quantum Information & Computation* 7.5 (July 2007), 401–430. ISSN: 1533-7146.

[Pul+25] Ema Puljak et al. *tn4ml: Tensor Network Training and Customization for Machine Learning*. arXiv:2502.13090 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2502.13090. URL: http://arxiv.org/abs/2502.13090 (visited on 06/23/2025).

[RD06] Matthew Richardson and Pedro Domingos. “Markov logic networks”. en. *Machine Learning* 62.1-2 (Feb. 2006), 107–136. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/s10994-006-5833-1. URL: http://link.springer.com/10.1007/s10994-006-5833-1 (visited on 01/14/2023).

[RS19] Elina Robeva and Anna Seigal. "Duality of graphical models and tensor networks". *Information and Inference: A Journal of the IMA* 8.2 (June 2019), 273–288. ISSN: 2049-8772. DOI: 10.1093/imaiai/iay009. (Visited on 07/06/2021).

[Roc97] Ralph Tyrell Rockafellar. *Convex Analysis*. English. reprint edition. Princeton: Princeton University Press, Jan. 1997. ISBN: 978-0-691-01586-6.

[Rud11] Sebastian Rudolph. "Foundations of Description Logics". en. *Reasoning Web. Semantic Technologies for the Web of Data: 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures*. Ed. by Axel Polleres et al. Berlin, Heidelberg: Springer, 2011, 76–136. 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 (visited on 03/14/2025).

[RN21] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach, Global Edition: A Modern Approach, Global Edition*. Englisch. 4th ed. Boston: Pearson, May 2021. ISBN: 978-1-292-40113-3.

[SIS17] Chiaki Sakama, Katsumi Inoue, and Taisuke Sato. "Linear Algebraic Characterization of Logic Programs". en. *Knowledge Science, Engineering and Management*. Ed. by Gang Li et al. Vol. 10412. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, 520–533. ISBN: 978-3-319-63557-6 978-3-319-63558-3. DOI: 10.1007/978-3-319-63558-3_44. URL: http://link.springer.com/10.1007/978-3-319-63558-3_44 (visited on 11/06/2023).

[San+25] Aaron Sander et al. *Large-scale stochastic simulation of open quantum systems*. arXiv:2501.17913 [quant-ph]. Jan. 2025. DOI: 10.48550/arXiv.2501.17913. URL: http://arxiv.org/abs/2501.17913 (visited on 04/02/2025).

[Sar+22] Md Kamruzzaman Sarker et al. "Neuro-symbolic artificial intelligence: Current trends". *AI Communications* 34.3 (Mar. 2022), 197–209. ISSN: 18758452, 09217126. DOI: 10.3233/AIC-210084. URL: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/AIC-210084 (visited on 04/17/2024).

[Sat17] Taisuke Sato. "A linear algebraic approach to datalog evaluation". en. *Theory and Practice of Logic Programming* 17.3 (May 2017). Publisher: Cambridge University Press, 244–265. ISSN: 1471-0684, 1475-3081. DOI: 10.1017/S1471068417000023. URL: https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/linear-algebraic-approach-to-datalog-evaluation/CED3EEB903D9D8A16843CFCA5AC4D577 (visited on 11/06/2023).

[SG16] Luciano Serafini and Artur S. d'Avila Garcez. "Learning and Reasoning with Logic Tensor Networks". *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*. Berlin, Heidelberg: Springer-Verlag, Nov. 2016, 334–348. 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 (visited on 04/02/2025).

[SB14] Shalev-Schwartz, Shai and Ben-David, Shai. *Understanding Machine Learning: From Theory to Algorithms*. Englisch. New York, NY, USA: Cambridge University Press, July 2014. ISBN: 978-1-107-05713-5.

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". en. *Bell System Technical Journal* 27.3 (1948). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x, 379–423. 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 (visited on 04/04/2025).

[SL08] Vin de Silva and Lek-Heng Lim. "Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem". en. *SIAM Journal on Matrix Analysis and Applications* 30.3 (Jan. 2008), 1084–1127. ISSN: 0895-4798, 1095-7162. DOI: 10.1137/06066518X. (Visited on 09/17/2019).

[Sim05] Helmut Simonis. "Sudoku as a constraint problem". *CP Workshop on modeling and reformulating Constraint Satisfaction Problems*. Vol. 12. Citeseer Sitges, Spain, 2005, 13–27. URL: https://ai.dmi.unibas.ch/_files/teaching/fs21/ai/material/ai26-simonis-cp2005ws.pdf (visited on 03/29/2025).

[SS16] Edwin Stoudenmire and David J Schwab. "Supervised Learning with Tensor Networks". *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., 2016, 4799–4807. (Visited on 01/10/2020).

[SH17] Daniel Suess and Milan Holzäpfel. "mpnum: A matrix product representation library for Python". en. *Journal of Open Source Software* 2.20 (Dec. 2017), 465. ISSN: 2475-9066. DOI: 10.21105/joss.00465. URL: https://joss.theoj.org/papers/10.21105/joss.00465 (visited on 06/23/2025).

[Tal14] Michel Talagrand. *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*. en. Berlin, Heidelberg: Springer, 2014. ISBN: 978-3-642-54074-5. DOI: 10.1007/978-3-642-54075-2. (Visited on 05/05/2020).

[TS94] Geoffrey G. Towell and Jude W. Shavlik. "Knowledge-based artificial neural networks". *Artificial Intelligence* 70.1 (Oct. 1994), 119–165. ISSN: 0004-3702. DOI: 10.1016/0004-3702(94)90105-8. URL: https://www.sciencedirect.com/science/article/pii/0004370294901058 (visited on 04/02/2025).

[TN17] Théo Trouillon and Maximilian Nickel. *Complex and Holographic Embeddings of Knowledge Graphs: A Comparison*. en. arXiv:1707.01475 [cs, stat]. July 2017. URL: http://arxiv.org/abs/1707.01475 (visited on 11/06/2023).

[TAP24] Efthimis Tsilionis, Alexander Artikis, and Georgios Paliouras. "A Tensor-Based Formalization of the Event Calculus". en. *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*. Jeju, South Korea: International Joint Conferences on Artificial Intelligence Organization, Aug. 2024, 3584–3592. ISBN: 978-1-956792-04-1. DOI: 10.24963/ijcai.2024/397. URL: https://www.ijcai.org/proceedings/2024/397 (visited on 09/24/2024).

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. English. 1st edition. New York, NY: Cambridge University Press, Sept. 2018. ISBN: 978-1-108-41519-4.

[Wai19] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press, 2019. ISBN: 978-1-108-49802-9. DOI: 10.1017/9781108627771. (Visited on 11/23/2020).

[WJ08] Martin J. Wainwright and Michael Irwin Jordan. *Graphical Models, Exponential Families, and Variational Inference*. en. Now Publishers Inc, 2008. ISBN: 978-1-60198-184-4.

[Wol24] Sebastian Wolf. *libxerus/xerus*. original-date: 2016-03-10T15:27:47Z. Feb. 2024. URL: https://github.com/libxerus/xerus (visited on 06/23/2025).

[Wol83] Stephen Wolfram. "Statistical mechanics of cellular automata". *Reviews of Modern Physics* 55.3 (July 1983). Publisher: American Physical Society, 601–644. DOI: 10.1103/RevModPhys.55.601. URL: https://link.aps.org/doi/10.1103/RevModPhys.55.601 (visited on 02/11/2025).

[Wol02] Stephen Wolfram. *A New Kind of Science*. Englisch. Illustrated Edition. Champaign (Ill.): Wolfram Media, May 2002. ISBN: 978-1-57955-008-0.

[Yan+15] Bishan Yang et al. “Embedding Entities and Relations for Learning and Inference in Knowledge Bases”. en. arXiv:1412.6575 [cs]. arXiv, Aug. 2015. URL: http://arxiv.org/abs/1412.6575 (visited on 11/06/2023).

[Zie00] Günter M. Ziegler. “Lectures on 0/1-Polytopes”. en. *Polytopes — Combinatorics and Computation*. Ed. by Gil Kalai and Günter M. Ziegler. Basel: Birkhäuser, 2000, 1–41. ISBN: 978-3-0348-8438-9. DOI: 10.1007/978-3-0348-8438-9_1. URL: https://doi.org/10.1007/978-3-0348-8438-9_1 (visited on 03/04/2025).

[Zie13] Günter M. Ziegler. *Lectures on Polytopes*. English. 1995th edition. New York: Springer, Oct. 2013. ISBN: 978-0-387-94365-7.