THE TENSOR NETWORK APPROACH TO EFFICIENT AND EXPLAINABLE AI

RESEARCH NOTES IN THE ENEXA PROJECT

Alex Goessmann, DATEV eG

January 24, 2025

ABSTRACT

Tensor spaces appear naturally in factored representations of systems, when storing information about a systems state. Since the curse of dimensionality prevents feasible generic representations and reasoning, logical and probabilistic reasoning focuses on tradeoffs between the sparsity and the generality of reasoning. In this work we present these tradeoffs based on the tensor network formalism and formulate feasible reasoning algorithms based on tensor network contractions. We review the classical logical and probabilistic approaches to reasoning using and show applications in neuro-symbolic AI.

Contents

1 Introduction **9**

1.1 Application of the tensor network approach . . . 9

1.1.1 Graphical Models . . . 9

1.1.2 Neuro-Symbolic AI . . . 10

1.1.3 Network Science . . . 10

1.2 Represention schemes of systems . . . 10

1.2.1 Ontological Commitments . . . 10

1.2.2 Logical and Probabilistic Approaches . . . 10

1.3 Literature . . . 11

1.3.1 Broad topics . . . 11

1.3.2 Tensors Representations . . . 12

1.4 Outline . . . 12

I Decomposition and Inference of Factored Representations **12**

2 Notation **12**

2.1 Categorical Variables and Representations . . . 12

2.2 Tensors . . . 13

2.3 One-hot encodings . . . 13

2.4 Contractions . . . 14
2.4.1 Tensor Product . . . 14
2.5 Illustrations by Hypergraphs . . . 14
2.6 Generic Contractions . . . 16
2.6.1 Decompositions . . . 17
2.7 Properties of Tensors . . . 17
2.8 Encoding schemes for functions . . . 17
2.8.1 Real-valued functions . . . 18
2.8.2 Relational encodings . . . 18
2.8.3 Tensor-valued functions . . . 18

3 Probability Distributions **19**
3.1 Tensor Representation of Distributions . . . 19
3.2 Marginal Distribution . . . 19
3.3 Conditional Probabilities . . . 20
3.4 Bayes Theorem and the Chain Rule . . . 22
3.5 Independent Variables . . . 22
3.6 Graphical Models . . . 24
3.6.1 Bayesian Networks . . . 24
3.6.2 Example of a Bayesian Network: Hidden Markov Models . . . 26
3.6.3 Markov Networks . . . 26
3.6.4 Bayesian Networks as Markov Networks . . . 27
3.7 Exponential Families . . . 28
3.7.1 Tensor Network Representation . . . 28
3.7.2 Mean Parameters . . . 29
3.7.3 Examples . . . 30
3.8 Empirical Distributions . . . 30

4 Probabilistic Reasoning **32**
4.1 Queries . . . 32
4.1.1 Querying by functions . . . 32
4.1.2 MAP Queries . . . 33
4.1.3 Answering queries by energy contractions . . . 33
4.2 Sampling based on queries . . . 33
4.2.1 Exact Methods . . . 33
4.2.2 Approximate Methods . . . 34
4.2.3 Simulated Annealing . . . 34
4.3 Maximum Likelihood Estimation . . . 34
4.3.1 Likelihood and Cross Entropy . . . 35
4.3.2 Entropic Interpretation . . . 35

4.4 Forward Mapping in Exponential Families . . . 36
4.4.1 Variational Formulation . . . 36
4.4.2 Mean Field Method . . . 37
4.4.3 Mode Search by annealing . . . 38
4.5 Backward Mapping in Exponential Families . . . 38
4.5.1 Variational Formulation . . . 38
4.5.2 Connection with Maximum Likelihood Estimation . . . 38
4.5.3 Connection with Maximum Entropy . . . 39
4.5.4 Alternating Algorithms to Approximate the Backward Map . . . 40

5 Propositional Logics **40**
5.1 Encoding of Booleans . . . 41
5.1.1 Booleans as categorical variables . . . 41
5.1.2 One-hot Encoding . . . 41
5.2 Semantics of Propositional Formulas . . . 41
5.2.1 Formulas . . . 42
5.2.2 Relational encoding of formulas . . . 42
5.3 Syntax of Propositional Formulas . . . 43
5.3.1 Atomic Formulas . . . 43
5.3.2 Syntactical combination of formulas . . . 44
5.3.3 Syntactical decomposition of formulas . . . 46
5.4 Discussion and Outlook . . . 48

6 Logical Inference **48**
6.1 Entailment in Propositional Logics . . . 48
6.1.1 Deciding Entailment by contractions . . . 48
6.1.2 Contraction Knowledge Base . . . 49
6.1.3 Sparse Representation of a Knowledge Base . . . 49
6.2 Formulas as Random Variables . . . 49
6.2.1 Conditioning on the atoms . . . 50
6.2.2 Conditioning on the formula . . . 50
6.2.3 Probability of a function given a Knowledge Base . . . 51
6.2.4 Deciding entailment on Markov Networks . . . 52
6.3 Deciding Entailment by partial ordering . . . 53
6.3.1 Monotonicity of Entailment . . . 53
6.4 Deciding Entailment by local contractions . . . 53
6.4.1 Knowledge Propagation . . . 54

II Neuro-Symbolic Learning **54**

7 Formula Selecting Networks **55**
7.1 Construction schemes . . . 55
7.1.1 Connective Selecting Tensors . . . 55
7.1.2 Variable Selecting Tensor Network . . . 56
7.2 State Selecting Tensors . . . 57
7.3 Composition of formula selecting maps . . . 57
7.3.1 Formula Selecting Neuron . . . 57
7.3.2 Formula Selecting Neural Network . . . 58
7.4 Application of Formula Selecting Networks . . . 60
7.4.1 Representation of selection encodings . . . 60
7.4.2 Efficient Representation of Formulas . . . 60
7.4.3 Batch contraction of parametrized formulas . . . 61
7.4.4 Average contraction of parametrized formulas . . . 61
7.5 Examples of formula selecting neural networks . . . 61
7.5.1 Correlation . . . 61
7.5.2 Conjunctive and Disjunctive Normal Forms . . . 61
7.6 Extension to variables of larger dimension . . . 62

8 Markov Logic Networks **62**
8.1 Markov Logic Networks as Exponential Families . . . 63
8.2 Tensor Network Representation of MLNs . . . 63
8.2.1 Energy tensors of Markov Logic Networks . . . 64
8.3 Expressivity . . . 65
8.4 Examples . . . 66
8.4.1 Distribution of independent variables . . . 66
8.4.2 Boltzmann machines as MLNs . . . 67
8.5 Applications . . . 68

9 Hard and Hybrid Logic Networks **68**
9.1 The limit of hard logic . . . 68
9.2 Hard Logic Networks . . . 70
9.2.1 Conjunctive Normal Representation . . . 70
9.2.2 Polynomial Representation of Formulas . . . 70
9.3 Hybrid Logic Network . . . 71
9.3.1 Tensor Network Representation . . . 72
9.3.2 Reasoning Properties . . . 72
9.3.3 Expressivity . . . 73
9.4 Categorical Constraints . . . 73

10 Parameter Estimation **75**
10.1 Learning Markov Logic Networks . . . 75

10.2 Alternating Algorithms to Approximate the Backward Map . . . 76
10.3 Forward and backward mappings of MLNs in closed form . . . 77
10.3.1 Maxterms and Minterms . . . 78
10.3.2 Atomic formulas . . . 79
10.4 Parameter Estimation in Hybrid Networks . . . 79

11 Structure Learning **80**
11.1 Overparametrization . . . 80
11.2 Learning in the minterm family . . . 80
11.3 Greedy Learning . . . 81
11.3.1 Gradient heuristic and the proposal distribution . . . 81
11.3.2 Gain Heuristic . . . 83
11.3.3 Iterations . . . 84

12 Probabilistic Success Guarantees **84**
12.1 Fluctuations . . . 84
12.1.1 Fluctuation of the empirical distribution . . . 84
12.1.2 Fluctuation tensors and their widths . . . 85
12.1.3 Naive Exponential Family . . . 85
12.1.4 Mean parameter in Markov Logic Networks . . . 85
12.1.5 Energy tensor in proposal distributions . . . 86
12.1.6 Binary Features . . . 86
12.1.7 Widths of random tensors . . . 86
12.2 Expected and Empirical Risk Minimization . . . 86
12.2.1 Maximum Likelihood Estimation on random data . . . 86
12.2.2 Solution of the Expected Problem . . . 87
12.2.3 Recovery Guarantee based on Widths . . . 88
12.3 Guarantees for Mode of the Proposal Distribution . . . 88
12.4 Guarantees for Parameter Estimation . . . 89

13 Statistical Models of First Order Logic (FOL) **90**
13.1 World Tensors . . . 90
13.1.1 Special Case of Propositional Logics . . . 90
13.1.2 Enumeration of worlds . . . 90
13.1.3 Random World Tensors . . . 91
13.2 Formulas in first order logics . . . 91
13.2.1 Syntactical Decomposition of quantifier-free formulas . . . 91
13.2.2 Quantifiers . . . 92
13.2.3 Basis CP Decomposition . . . 93
13.2.4 Example: Queries . . . 93
13.3 Representation of Knowledge Graphs . . . 94

13.3.1 Representation as unary and binary predicates . . . 94
13.3.2 Representation as ternary predicate . . . 94
13.3.3 SPARQL Queries . . . 95
13.4 Probabilistic Relational Models . . . 96
13.4.1 Markov Logic Networks in FOL . . . 96
13.4.2 Importance Formula . . . 97
13.4.3 Decomposition of the log likelihood . . . 97
13.4.4 Interpretation as Likelihood of Propositional Dataset . . . 98
13.4.5 Approximation by Independent Samples . . . 100
13.4.6 Extraction of Samples from FOL Knowledge Bases . . . 100
13.4.7 Representation by Tensor Networks . . . 100
13.4.8 Basis CP Decomposition of extracted data . . . 101
13.4.9 Representation with auxiliary term variables . . . 101
13.4.10 Generic Tensor Network Decomposition of Extracted Data . . . 101
13.4.11 Design of the Formulas . . . 102
13.5 Generation of FOL worlds . . . 102
13.5.1 Samples by single objects . . . 102
13.6 Samples by pairs of objects . . . 102
13.7 Example: Generation of Knowledge Graphs . . . 103
13.7.1 Samples by single resources . . . 103
13.7.2 Samples by pairs of resources . . . 104
13.7.3 General orthogonal projection approach . . . 104
13.8 Discussion . . . 105

III Analysis of the Contraction Calculus 105

14 Coordinate Calculus 105
14.1 One-hot encodings as basis . . . 105
14.2 Selection Encodings . . . 106
14.2.1 Tensors as linear maps . . . 106
14.2.2 Selection encodings . . . 107
14.3 Discussion . . . 108

15 Directed Tensor Calculus 108
15.1 Directed Tensors . . . 108
15.1.1 Normation . . . 109
15.1.2 Contraction of Directed Tensors . . . 109

16 Contraction Equations 110
16.1 Contraction equations in logical and probabilistic reasoning . . . 110

16.2 Normation Equations . . . 111
16.3 Proof of Hammersley-Clifford Theorem . . . 111
16.4 Commutation of Contractions . . . 113
16.5 Support of Contractions . . . 114

17 Basis Calculus **115**
17.1 Encoding of Subsets and Relations . . . 115
17.1.1 Higher order relations . . . 116
17.2 Encoding of Functions . . . 117
17.2.1 Relational Encoding of Functions . . . 117
17.3 Calculus of relational encodings . . . 118
17.3.1 Function Evaluation . . . 118
17.3.2 Composition of function . . . 118
17.3.3 Compositions with real functions . . . 119
17.3.4 Decomposition in case of structured images . . . 121
17.4 Effective Calculus . . . 121
17.5 Applications in Machine Learning . . . 122

18 Contraction Message Passing **123**
18.1 Exact Contractions . . . 123
18.1.1 Construction of Cluster Graphs . . . 123
18.1.2 Message Passing to calculate contractions . . . 123
18.1.3 Variable Elimination Cluster Graphs . . . 124
18.1.4 Bethe Cluster Graphs . . . 124
18.1.5 Computational Complexity . . . 125
18.2 Approximate Contractions . . . 125
18.2.1 Exact Message Passing for Directed and Binary Contractions . . . 125
18.2.2 Case of Matrices . . . 126
18.2.3 Case of Tensors . . . 126
18.3 Basis Calculus . . . 126
18.4 Applications . . . 126

19 Sparse Tensor Representations **126**
19.1 CP Formats . . . 127
19.1.1 Directed Leg Cores . . . 127
19.1.2 Basis Leg Cores . . . 128
19.1.3 Basis+ Leg Cores . . . 129
19.2 Representation involving selection architectures . . . 130
19.2.1 Applications . . . 131
19.3 Constructive Bounds on CP Ranks . . . 131
19.3.1 Format Transformations . . . 132

19.3.2 Summation of CP Decompositions . . . 132
19.3.3 Contractions of CP Decompositions . . . 132
19.3.4 Normations of CP Decompositions . . . 133
19.3.5 Sparse Encoding of Functions . . . 133
19.3.6 Construction by averaging the incoming legs . . . 134
19.4 Subspaces of formulas . . . 134
19.4.1 Formula Subspaces . . . 134
19.4.2 Formula Decomposition as a Subspace Choice . . . 134

20 Optimization of Sparse Tensors **135**
20.1 Higher-Order Unconstrained Binary Optimization (HUBO) . . . 135
20.2 Quadratic Unconstrained Binary Optimization (QUBO) . . . 135
20.3 Integer Linear Programming . . . 136

21 Reasoning by Tensor Approximation **137**
21.1 Approximation of Energy tensors . . . 137
21.1.1 Direct Approximation . . . 137
21.1.2 Approximation involving Selection Architectures . . . 137
21.2 Transformation of Maximum Search to Risk Minimization . . . 137
21.2.1 Squares Risk Trick . . . 137
21.3 Parametercores . . . 138
21.4 Fitting a tensor by a formula tensor . . . 138
21.5 Alternating Solution of Least Squares Problems . . . 138

22 Uniform Concentration of Random Contractions **139**
22.1 Naive bounds given binomial coordinates . . . 139
22.1.1 Basis Vectors . . . 139
22.1.2 Sphere . . . 139
22.1.3 Bounds based on the sub-gaussian norm . . . 141
22.2 Chaining bounds given binomial coordinates . . . 141
22.2.1 Generic Width Bounds . . . 142

A Implementation in the `tnreason` package **142**
A.1 Script Language . . . 143
A.1.1 Propositional Sentences by Nested Lists . . . 143
A.1.2 Knowledge Bases . . . 143
A.1.3 Markov Logic Networks . . . 143
A.1.4 Neuro-Symbolic Architecture by Nested Lists . . . 144
A.2 Tensor Networks . . . 144
A.2.1 Core Nomenclature . . . 144
A.2.2 Color Nomenclature . . . 145

A.3 Architecture . . . 145
A.4 Subpackage engine . . . 145
A.5 Subpackage encoding . . . 147
A.5.1 Relational encoding of formulas . . . 147
A.5.2 Representation of MLNs . . . 147
A.5.3 Formula Selecting Maps . . . 147
A.6 Subpackage algorithms . . . 148
A.6.1 Alternating Least Squares . . . 148
A.6.2 Gibbs Sampling . . . 148
A.6.3 Knowledge Propagation . . . 148
A.7 Subpackage knowledge . . . 148
A.7.1 Distributions . . . 148
A.7.2 Inference . . . 149
A.7.3 Parameter Estimation . . . 149
A.7.4 Structure Learning . . . 150

1 Introduction

Reasoning based on tensors has the advantage to

- Provide a unifying approach to logical and probabilistic reasoning
- Define necessary workload of algorithms by contractions

The storage of numeric information about the states of a system with multiple variables is naturally a tensor. Tensors appear naturally in

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

The curse of dimensionality renders the representations of tensors by a list of basis elements infeasible. Therefore, sparse representation formats by tensor networks have been investigated.

Tensor network decompositions as representation schemes appear in

- Logics: Conjunctions of formulas are Hadamard products of the tensor representation of formulas (Coordinate Calculus/ Effective Calculus)
- Probability theory: Graphical models are tensor networks of the factors
- Data bases: Relations encoded by lists as storage of nonvanishing coordinates of a relation encoding

Tensor network contractions as reasoning schemes appear in

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

1.1 Application of the tensor network approach

1.1.1 Graphical Models

Further sparsity schemes for factors, by exploiting structure inside them.

1.1.2 Neuro-Symbolic AI

Tensor Approaches to Neuro-Symbolic AI

- TensorLog Cohen et al. (2020)
- Badreddine et al. (2022) representation of logic using tensor networks and automated differentiation to optimize.

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

Sparsity means restriction to functions, which are decomposability into a small number of neurons. Approximations of generic functions (see the universal approximation theorems) would require When restricting to functions based on a fixed architecture, the sparsity of functions

1.1.3 Network Science

Statistical Models of Knowledge Graphs going beyond the typical single edge type perspective of network science.

1.2 Represention schemes of systems

Following the book of Russel: Atomic, Factored and Structured Representations, which are framed as ontological commitments.

Tensors have frequent appearance in machine learning. We show in this chapter how they are natural choices to encode states the states of factored systems, which are systems described by a collection of categorical variables.

1.2.1 Ontological Commitments

Differing ontological commitments are made by atomic systems, where we just enumerate the state of a system and understand them as assignments to a single variable. We can always transform a factored representation of a system to an atomic one, just by enumerating the tuples of states of the factored system and interpreting them as a single variable. However, by doing so we would loose much of the structure of the representation, which we would like to exploit in reasoning processes.

A more generic representation of systems are structured representation. In structured systems the numbers of variables can differ depending on the state the system is in. Limitation to factored representations: Dealing with structured representations would mean differing order of the representing tensors.

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 continous variables.

1.2.2 Logical and Probabilistic Approaches

Besides ontological commitments in the choice of a representation scheme, one also needs to make epistomologic commitments, by defining what properties we can reason about. We can encode different properties of the system in the axes of the tensor representation. In logical approaches the properties of states are Boolean values representing, for example, whether a state is consistent with specific constraints. Probabilistic approaches represent on the coordinates of the tensors real numbers in $[0, 1]$ which can be interpreted as probability of a state. Compared with logical approached to reasoning, probabilistic approaches thus bear a more expressive modelling.

Tensor Networks can represent both probabilistic and logical reasoning on factored systems, since in the tensor space we can

- Represent probability distributions by storing probabilistic values in each coordinate
- Represent set of states by sums over different one-hot encodings (this enables logical calculus)

Both probability and logic provide a human-understandable interface to machine learning. As we will describe in the following sections, they can be combined in one formalism providing efficient reasoning. This formalism of tensors along their network decompositions and contractions bears the potential of parallel computations exploited in the AI-dedicated soft and hardware.

Probability represents the uncertainty of states. The categorical variables are called random variables and their joint distribution is represented by a probability tensor. Humans can interpret probabilities by Bayesian and frequentist

approaches. Reasoning based on Bayes Theorem has an intuitive interpretation in terms of evidence based update of prior distributions to posterior distributions. However it is based on interpreting (large amounts) of numbers, which makes it hard for humans to assess the probabilistic reasoning process.

Logics explains relations between sets of worlds in a human understandable way. Categorical variables have dimension 2, where the first is interpreted as indicating a False state and the second as a True state. We mainly restrict to propositional logics, where there are finite sets of such variables called atomic formulas. Using model-theoretic semantics it defines entailment of sets by other sets, which is understandable as a consequence relation.

Tensors unify both approaches since they are natural mathematical structures to represent properties of states in factored systems. The potential is then based in employing efficient multilinear algorithms based on tensor network alternations to solve reasoning problems. Further, algorithms formulates in tensor networks have a high parallelization potential, which is why they are of central interest in the development of AI-dedicated software and hardware.

The different areas have developed separated languages to describe similar objects. Here we want to provide a rough comparison of those in a dictionary.

	Probability Theory	**Propositional Logic**	**Tensors**
Atomic System	Random Variable	Atomic Formula	Vector
Factored System	Joint Distribution	Knowledge Base	Tensor
Categorical Variable	Random Variable	Atomic Formula	Axis of the Tensor

While the probability theory lacks to provide an intuition about sets of events, propositional syntax has limited functionality to represent uncertainties. Tensors on the other side can build a bridge by representing both functionalities and relying on probability theory and logics for respective interpretations.

1.3 Literature

1.3.1 Broad topics

Inductive Logic Programming:

- ILP is a classical task Muggleton and De Raedt (1994)
- Amie Galárraga et al. (2013) is a method of learning Horn clauses using a refinement operator.
- Class Expression Learning Lehmann et al. (2011) is a more recent approach which has recently seen further popularity in combination with reinforcement learning Demir and Ngonga Ngomo (2021) and neural networks Kouagou et al. (2022, 2023)

Statistical Relational AI: Getoor and Taskar (2019)

- Classical combination of logical and probabilistic approaches to reasoning

Neurosymbolic AI

- Required for more advanced AI Hochreiter (2022)
- Add the paradigm of neural computing to logical reasoning
- Potential benefits from Statistical Relational AI Marra et al. (2024)
- Tensor based approaches Cohen et al. (2020)
- Badreddine et al. (2022) representation of logic using tensor networks and automated differentiation to optimize.

Knowledge Graphs Hogan et al. (2021)

- The advent of large Knowledge Graphs enables reasoning methods.
- The bottleneck is the efficiency of methods to identify formulas from data.
- Knowledge Graphs are stored in a sparse format, i.e. only true atoms instead of all + truth label.

1.3.2 Tensors Representations

Tensor Representation is natural, if systems are described in factored representation.

Tensor Network formats

- HT Format **?**
- CP Format

Tensor Representation of Logics

- Tensor Networks have been applied in the automatization of logic reasoning Sakama et al. (2017); Sato (2017) apply Matrix multiplication in reasoning.
- Nickel et al. (2016) review over relational machine learning and latent features via matrix embeddings.

Tensor Representation of Knowledge Graphs

- Effective representation of queries
- Usage of tensor networks in embeddings Yang et al. (2015) and using complex extensions Trouillon and Nickel (2017); Trouillon (2017)

Tensor Representation of Graphical Models

- Duality of Graphical Models and Tensor Networks: Robeva and Seigal (2019)
- Expressivity studies Glasser et al. (2019)

Tensor Networks as Regressors

- Dynamical Systems learning Gelß et al. (2019); Goeßmann et al. (2020)

1.4 Outline

Part I: Review of classical approaches using tensor calculus

Part II: Neuro-symbolic learning

Part III: Numerical and Stochastic investigation of the calculus with tensors

Part I

Decomposition and Inference of Factored Representations

The computational automation of reasoning is rooted both in the probabilistic and the logical reasoning tradition. Both draw on the same ontological commitment that systems have a factored structure, that is their states are described by assignments to a set of variables. Based on this commitment both approaches bear a natural tensor representation of their states and a formalism of the respective reasoning algorithms based on multilinear methods.

2 Notation

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

2.1 Categorical Variables and Representations

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

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

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

of cardinality m.

We notate variables by large literals and indices by small literals.

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

2.2 Tensors

We define Tensors here in an non-canonical way based on categorical variables assigned to its axis. This allows us to define contractions without further specification of axes, based on comparisons of shared categorical variables.

In the following we will denote index sets as

$$[m_k] := \{0, \dots, m_{k-1}\}\,.$$

Tensors are multiway arrays and a generalization of vectors and matrices to higher orders. We will first define Tensors by their coordinates.

Definition 3 (Tensor)**.** *Let there be numbers $m_k \in \mathbb{N}$ for $k \in [d]$ and categorical variables X_k taking their values in $[m_k]$. We call maps*

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

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

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

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

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

which is, with the operations of coordinatewise summation and scalar multiplication, a linear space called a tensor space.

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

Example 1 (Trivial Tensor)**.** *The trivial tensor is defined as the map*

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

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

$$\mathbb{I}\left[X_0 = x_0, \dots, X_{d-1} = x_{d-1}\right] = 1\,.$$

2.3 One-hot encodings

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

Definition 4 (One-hot representations of Atomic Systems)**.** *Given an atomic system described by the categorical variable X, we define for each $x \in [m]$ the basis vector $e_x[X]$ by*

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

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

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

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

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

$$e_x[X] = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0] \,, \tag{2}$$

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

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

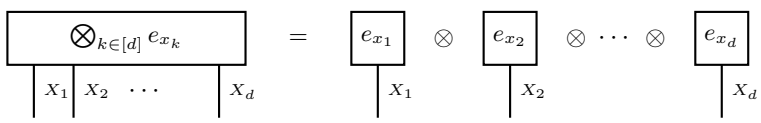
Definition 5 (One-hot representations of Factored Systems)**.** *Let there be a factored system defined by a tuple $(X_0, \ldots, X_{d-1})$ of variables taking values in $\bigtimes_{k\in[d]}[m_k]$. The one-hot encoding of its states is the tensor product of the one-hot encoding to each categorical variables,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

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

defined by mapping $x_0, \ldots, x_{d-1}$ to

$$e_{x_0,\ldots,x_{d-1}}[X_0, \ldots, X_{d-1}] =: \bigotimes_{k\in[d]} e_{x_k}[X_k] \,.$$

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



Remark 1 (Flattening of Tensors)**.** *The use the tensor product to represent states of factored systems can be motivated by the reduction to atomic systems by enumeration of the states. We have this property reflected in the state encoding of factored systems, since the tensor space $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ is isomorphic to the vector spaces $\mathbb{R}^{\prod_{k\in[d]} m_k}$. This operation is called flattening (or unfolding) of tensors with many axes to tensors of less axes.*

2.4 Contractions

2.4.1 Tensor Product

Definition 6 (Tensor Product)**.** *Let there be two tensor*

$$T[X_0, \ldots, X_{d-1}] : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R} \quad \textit{and} \quad \tilde{T}[Y_0, \ldots, Y_{p-1}] : \bigtimes_{l\in[r]} [m_l] \to \mathbb{R}$$

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

$$\left(T \otimes \tilde{T}\right)[X_0, \ldots, X_{d-1}, Y_0, \ldots, Y_{p-1}] : \left(\bigtimes_{k\in[d]} [m_k]\right) \times \left(\bigtimes_{l\in[r]} [m_l]\right) \to \mathbb{R}$$

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

$$\begin{aligned}\left(T \otimes \tilde{T}\right)&[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}, Y_0 = y_0 \ldots, Y_{r-1} = y_{r-1}] \\ &:= T[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] \cdot \tilde{T}[Y_0 = y_0 \ldots, Y_{r-1} = y_{r-1}] \,.\end{aligned}$$

2.5 Illustrations by Hypergraphs

Definition 7. *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 a diagrammatic illustration of tensors, where tensors are represented by block nodes and each axis as a leg decorated with a categorical variable X_k representing the choice of an element in the set $[m_k]$ as shown in Figure 1b).

Another useful depiction is by hypergraphs, which are generalizations of graphs allowing that edges being arbitrary nonempty subsets of the nodes, whereas canonical graphs assume pairs. Along this depiction (see Figure 1a) each axis of the tensor is represented by a node representing the variable X_k and the tensor T is associated with the hyperedge e connecting all variables.

The two representations are dual to each other in the sense that the nodes of the one are the hyperedges of the other graph.

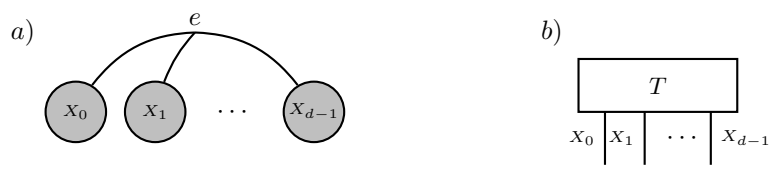
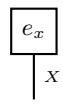


Figure 1: Depiction of Tensors a) Decoration of hyperedges connecting the variables X_k to each axis $k \in [d]$. b) Blockwise notation with axis denoted by open legs represented by the variables X_k.

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



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

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

Definition 8. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a hypergraph with nodes decorated by categorical variables X_v with dimensions*

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

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

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

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

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

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

Diagrammatic notation: Best to do version a) as used in the definition, highlighting that tensors have shared categorical variables with fixed dimensions.

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

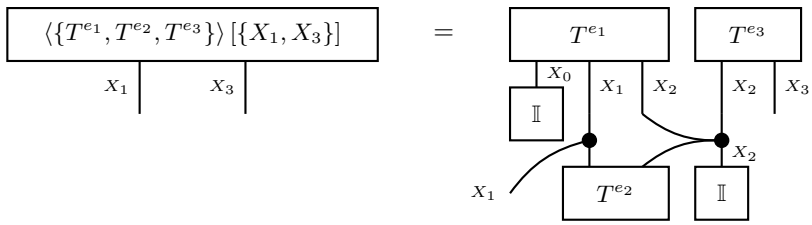


Figure 3: Example of a tensor network contraction of all but the variables X_1, X_3. Contraction of variables can always be depicted by closing the open legs with trivial tensors $\mathbb{I}$ performing index sums.

2.6 Generic Contractions

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

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

Definition 9. *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network on a decorated hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For any subset $\tilde{\mathcal{V}} \subset \mathcal{V}$ we define the contraction to be the tensor*

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

defined coordinatewise by the sum

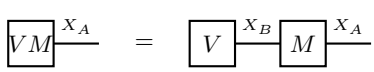
$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_v = x_v \,:\, v \in \tilde{\mathcal{V}} \right] = \sum_{\{x_v \in [m_v] \,:\, v \in \mathcal{V}/\tilde{\mathcal{V}}\}} \left(\prod_{e \in \mathcal{E}} T^e \left[X_v = x_v \,:\, v \in e \right] \right) . \tag{4}$$

Diagrammatic notation: Best to do version b), since this is easiest to see how tensors combine to new tensors by contractions.

Remark 2 (Alternative Notations)**.** *Contractions can also denoted by the Einstein summations of the indices along connected edges, understood as scalar product in each subspace. This is as in Definition 9, just omitting the sums. We found it useful in this work to do the diagrammatic representation instead, since it offers a better possibility to depict hierarchical arrangements of shared variables.*

Example 2. *Tensor products are examples of contractions, where the hypergraph consists of two disjoint edges, which union is left open.*

Example 3. *Matrix vector contractions are special cases of Tensor Contractions. We use the visualization as in Figure 1b where vectors are blocks with single categorical variables and matrices with two indices and matrix vector products by*



Here the index j is represented by a closed edge, which means that it is eliminated by a sum.

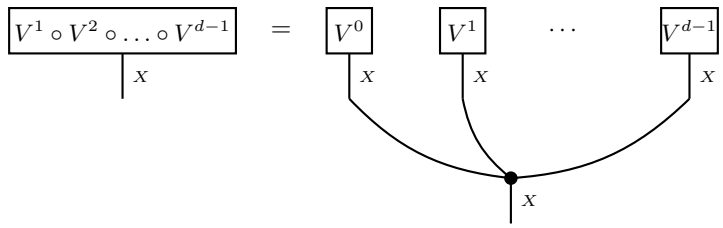
Example 4. *A node appearing in arbitrary many hyperedges denotes a Hadamard product of the axis of the respective decorating tensors. To give an example, let $V^k \in \mathbb{R}^p$ be vectors for $k \in [d]$. Their hadamard product is the vector*

$$V^1 \circ V^2 \circ \ldots \circ V^{d-1} \in \mathbb{R}^p$$

defined by

$$\left(V^1 \circ V^2 \circ \ldots \circ V^{d-1}\right)_x = \prod_{k \in [d]} V^k_x \,.$$

In a contraction diagram the Hadamard product is depicted by



2.6.1 Decompositions

Definition 10. *Let T be a tensor in $\bigotimes_{v\in\mathcal{V}} \mathbb{R}^{m_v}$. A Tensor Network Decomposition of a tensor $T\left[X_v \,:\, v \in \mathcal{V}\right]$ is a Tensor Network $\mathcal{T}^{\mathcal{G}}$ to a hypergraph containing such that*

$$T\left[X_v \,:\, v \in \mathcal{V}\right] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_v \,:\, v \in \mathcal{V}\right] .$$

2.7 Properties of Tensors

We will often encounter situations, where the coordinates of tensors are binary valued, i.e. the tensor is a map to the set $[2]$.

Definition 11. *We call a tensor binary, when $\operatorname{im}(T) \subset [2]$, i.e. all coordinates are either 0 or 1.*

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

Definition 12. *A Tensor*

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

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

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

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

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

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

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

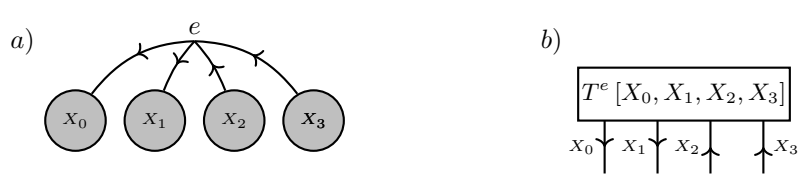
The normation of a on $\mathcal{V}^{\text{in}} \subset \mathcal{V}$ normable tensor is the tensor

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

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

We will investigate the contractions of directed tensors in Part III, where we show in Theorem 72 that normations are directed tensors.

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



2.8 Encoding schemes for functions

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

2.8.1 Real-valued functions

Example 5 (Uncertainty about States)**.** *The uncertainty about the state of a categorical variable X can be expressed in vectors. For example let there be real numbers $\mathbb{P}[X = x] \in [0, 1]$ for $x \in [m]$ with $\sum_{x\in[m]} \mathbb{P}[X = x] = 1$ with the interpretation that $\mathbb{P}[X = x]$ is the probability of a system being in state x. We can represent this uncertain state simply by a vector*

$$\mathbb{P}[X] \in \mathbb{R}^m$$

defined as the sum of one-hot representations weighted by $\mathbb{P}[X = x]$

$$\sum_{x\in[m]} \mathbb{P}[X = x] \cdot e_x[X] = [\mathbb{P}[X = 0] \quad \mathbb{P}[X = 1] \quad \cdots \quad \mathbb{P}[X = m - 1]] \ .$$

2.8.2 Relational encodings

We have already observed in Example **??**, that any function of a categorical variable has a representation as a linear function acting on the one-hot encoding of the variable. Let us now show how we can encode maps between factored systems. The scheme is described in more generality and detail (encoding of subsets and relations) in Chapter 17, see Definition 70.

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

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

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

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

defined by

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

When the categorical variables of the image factored system to a map f are not specified otherwise, we will denote them by X_f.

2.8.3 Tensor-valued functions

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

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

the selection encoding of f is a tensor

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

defined by the basis decomposition

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

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

We will provide more detail to the tensor representation of functions in Chapter 17.

3 Probability Distributions

After having discussed how to represent states of factored systems by tensors, let us now take advantage of these representation by associating properties with these states. We will associate with each state its probability and have represented a probability distribution about the states. This is an epistemological commitment about the system, since we assume the property of probability to each state and reason about this property.

3.1 Tensor Representation of Distributions

Let there be uncertainties of the assignments x_k to the categorical variables X_k of a factored system. We then understand X_k as random variables, which have a joint distribution defined by the uncertainties of the state assignments. To capture these uncertainties we now make use of the one-hot representation of factored systems in Chapter **??**.

Definition 16 (Probability Tensor)**.** *Let there be a factored system defined by a categorical variable X_k for each $k \in [d]$ taking values in $[m_k]$. A probability distribution over the states of $\mathcal{F}$ is a tensor*

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

such that

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

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

The probability tensor to the distribution is an object

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

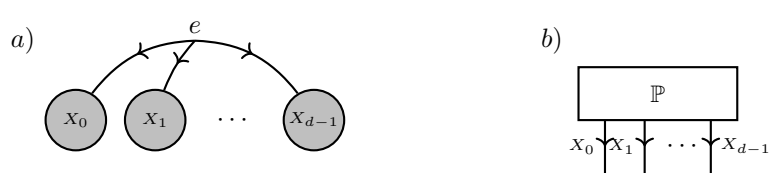
which is the sum over the one-hot encodings (see Lemma 23)

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

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

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

Probability tensors are depicted as



3.2 Marginal Distribution

The contraction formalism is especially suited in visualizing conditional distributions and marginalizations.

Definition 17 (Marginal Probability)**.** *Given a distribution $\mathbb{P}\left[X_0, X_1\right]$ of the categorical variables X_0 and X_1 the marginal distribution of the categorical variable X_0 is defined for each x_0 as the tensor*

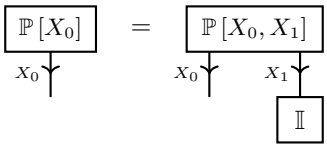
$$\mathbb{P}\left[X_0\right] : [m_0] \to \mathbb{R}$$

defined for $x_0 \in [m_0]$ by

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

Definition 17 generalizes to marginalizations of sets of variables, since we can always group a set of categorical variables and understand them as a single one.

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



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

Theorem 1. *Given a Tensor Network (see Definition 8) $\{\mathbb{P}\}$ consistent of the variables X_0, X_1 and hyperedge $\{X, X_1\}$ decorated with the tensor $\mathbb{P}$. Then the marginal distribution of the variable X is the contraction*

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

Further, any marginal distribution is a probability distribution.

Proof. We have $\mathbb{P}[X_0] = \langle\{\mathbb{P}\}\rangle[X_0]$ by definition. To show that $\mathbb{P}[X_0]$ is a probability distribution, we need to show that $\langle\{\mathbb{P}[X_0]\}\rangle[\varnothing] = 1$. But this follows from the normation of $\mathbb{P}$ and the commutativity of contractions (see Theorem 77 in Chapter 18) as

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

□

3.3 Conditional Probabilities

Conditional Probability Tensors are Probability Tensor with additional legs, where each slice to these additional legs defines a probability tensor.

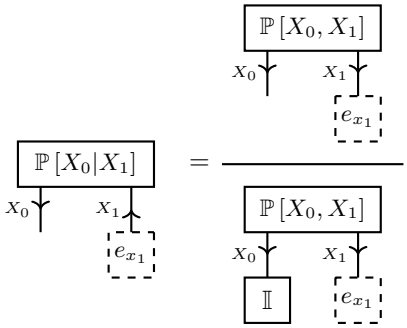
Definition 18 (Conditional Probability)**.** *Given a distribution $\mathbb{P}$ of the categorical variables X_0 and X_1, the conditioned distribution of X is defined by*

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

The conditional probability

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

is represented as a tensor with legs to X_0 and X_1. For each one-hot encoding e_{x_1} of the assignment x_1 to the variable X_1 we represent the conditional probability by the diagrams



Here we denote by the quotient a coordinatewise normation, as sketched by the dashed unit vector. We depict conditional variables by directed edges, where legs to conditions are incoming while the others outgoing.

We will discuss operations on tensors like conditioning more detail in Chapter 15 as normation operation of Definition 13. In Theorem 2 we will show that the resulting tensor is directed with incoming variables by the conditions.

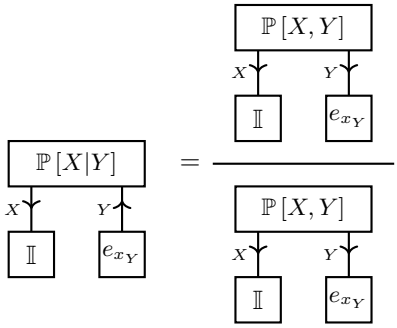
Theorem 2. *The tensor $\mathbb{P}[X_0|X_1]$ is the normation of $\mathbb{P}[X_0, X_1]$ on X_1 (see Definition 13), that is*

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

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

Proof. The first claim follows from a comparison of Definition 18 and 13. The second claim follows from the first and Theorem 72. Alternatively, the second claim can be showed using the diagrammatic notation as

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



$= \ 1 \, .$

□

Theorem 1 and 2 apply the formalism of contractions in the basic operations of probabilistic reasoning. Marginalization and Conditioning differs here only by the core e_{x_1} in the contraction.

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

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

Proof. By Theorem 2 a conditional probability tensor $\mathbb{P}[X_0|X_1]$ is the normation of a tensor and by Theorem 72 a directed tensor. Since probability tensors have only non-negative coordinates, their contractions with one-hot encodings also have only non-negative coordinates and also their normations. Conversely, let $T[X_\mathcal{V}]$ be a directed tensor with $\mathcal{V}^{\text{in}}$ incoming and $\mathcal{V}^{\text{out}}$ outgoing and non-negative coordinates. Then

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

is a probability tensor, since

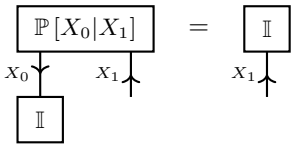
$$\sum_{x_{\mathcal{V}^{\text{in}}}} \sum_{x_{\mathcal{V}^{\text{out}}}} \mathbb{P}[X_\mathcal{V} = x_\mathcal{V}] = \sum_{x_{\mathcal{V}^{\text{in}}}} \sum_{x_{\mathcal{V}^{\text{out}}}} \frac{1}{\prod_{v \in \mathcal{V}^{\text{in}}} m_v} \cdot T[X_\mathcal{V} = x_\mathcal{V}] = \sum_{x_{\mathcal{V}^{\text{in}}}}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

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

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

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

Since conditional probabilities are directed tensors we therefore depict them by



.

Theorem 3 specifies a broad class of tensors to represent conditional probabilities. In combination with Theorem 81, which states that relational encodings are directed, we get that any relational encoding of a function is a conditional probability tensor.

3.4 Bayes Theorem and the Chain Rule

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

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

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

Proof. Directly from the more generic contraction equation Theorem 74, since by assumption of positivity of $\mathbb{P}[X_1]$, the tensor network $\{\mathbb{P}\}$ is normable with respect to $\{X_1\}$. □

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

Theorem 5 (Chain Rule)**.** *For any joint probability distribution $\mathbb{P}$ of the variables $\mathbb{P}[X_0, \ldots, X_{d-1}]$ we have*

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

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

Proof. This follows from Theorem 75. □

3.5 Independent Variables

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

Definition 19 (Independence)**.** *Given a joint distribution of variables X_0 and X_1, we say that X_0 is independent from X_1 if for any values x_0, x_1 we have*

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

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

Theorem 6. *Given a probability distribution $\mathbb{P}$, X_0 is independent from X_1, if and only if*

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

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

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

is equivalent to

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

We reorder the summations and arrive at

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

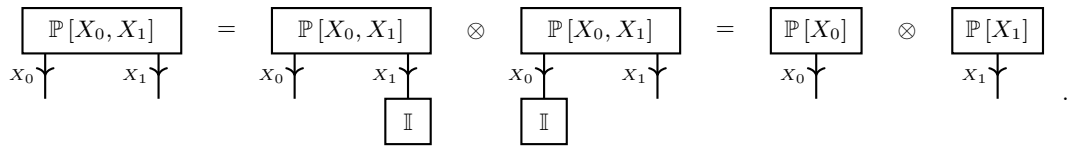
which is by Lemma 23 equal to the claim

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

□

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

In the tensor network decomposition we depict this by



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

Definition 20 (Conditional Independence)**.** *Given a joint distribution of variables X_0, X_1 and X_2, we say X_0 is independent from X_1 conditioned on X_2 if for any incides x_0, x_1 and x_2*

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

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

Theorem 7 (Conditional Independence as a Contraction Equation)**.** *Given a distribution $\mathbb{P}$ of variables X_0, X_1 and X_2, the variable X_0 is independent from X_1 if and only if the contraction equation*

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

holds.

Proof. Directly by Theorem 2 used on the conditional probabilities in Definition 20. □

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

Corollary 1. *If and only if X_0 is independent from X_1 conditioned on X_2 the probability distribution $\mathbb{P}$ satisfies*

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

Proof. Follows from Theorem 7 and Theorem 4. □

Corollary 2. *Whenever X_0 is independent of X_1 given X_2, we have*

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

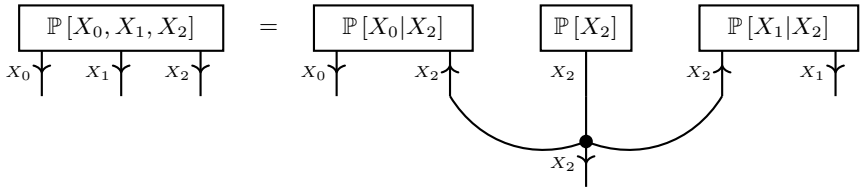


Figure 4: Diagrammatic visualization of the contraction equation in Corollary 1. Conditional independence of X_0 and X_1 given X_2 holds if the contraction on the right ride is equal to the probability tensor on the left side.

Theorem 8 (Markov Chain)**.** *Let there be a set of variables X_t where $t \in [T]$. When X_t is independent of $X_{0:t-2}$ conditioned on X_{t-1} (the Markov Property), then*

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

We depict this decomposition in Figure 5.

Proof. By the chain rule (Theorem 5) we have

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

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

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

Composing both equalities shows the claim. □

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

Remark 3. *Let us notice that the dimensionality dropped drastically through applying the independence assumption. The tensor space in the naive representation of any probability distribution has*

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

coordinates, while the Markov Chain is represented by

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

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

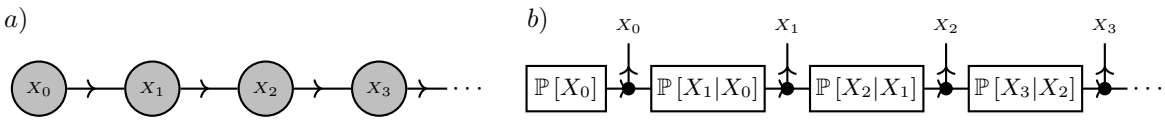


Figure 5: Depiction of a Markov Chain. a) Dependency Graph (of the corresponding chain Graphical Model). b) Dual Tensor Network representing the conditional probability factors.

3.6 Graphical Models

We have already depicted conditional dependency assumptions made for Markov Chains in Figure 5 and discussed the implied decomposition of the dual tensor networks. Graphical models provide a more general framework for conditional dependency assumptions and provide a generic approach to exploit independences in finding tensor network decompositions of $\mathbb{P}$.

Graphical Models are typically depictured by nodes to each variable and edges. Whether the graph is directed or not distinguished between Bayesian Networks and Markov Networks.

Remark 4 (Alternative definitions)**.** *In the literature, tensor networks are often called dual to the hypergraphs defining graphical models (see e.g. Robeva and Seigal (2019)). The duality becomes clear, when one interpretes the tensors as cores and their common variables as edges. We in this work avoid this ambiguity by directly defining tensor networks as decoration of hyperedges by tensors.*

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

Further, we directly use hypergraphs instead of the more canonical association of factors with cliques of a graph. This avoids the discussion of non-maximal cliques as decorated with trivial tensors. Such hypergraphs follow the same line of though compared with factor graphs, which are bipartite graphs with nodes either corresponding with single variables or with a collection of them affected by a factor.

3.6.1 Bayesian Networks

Bayesian networks are described by directed acyclic graphs (DAG). The probability distribution is a Hadamard product of conditional probabilities, where each variable has a conditional probability factor conditioned on the parents variables in the graph.

We introduce Bayesian Networks based on directed hypergraphs (see Definition 7) and define further properties.

Definition 21. *A directed path is a sequence $v_0, \dots v_r$ such that for any $l \in [r]$ there is an hyperedge $e = (e^{\text{in}}, e^{\text{out}}) \in \mathcal{E}$ such that $v_l \in e^{\text{in}}$ and $v_{l+1} \in e^{\text{out}}$. We call the hypergraph $\mathcal{G}$ acyclic, if there is no path with $r > 0$ such that $v_0 = v_r$. Given a directed hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we define for any node $v \in \mathcal{V}$ its parents by*

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

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

Definition 22. *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* $(\mathrm{Pa}(v), \{v\})$ *by a conditional probability distribution*

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

which represents the probability distribution

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

By definition each tensor decorating a hyperedge is directed with $X_{\mathrm{Pa}(v)}$ incoming and X_v outgoing. Thus, the directionality of the hypergraph is reflected in each tensor decorating a directed hyperedge. This allows us to verify with Theorem 73 that their contraction defines a probability distribution.

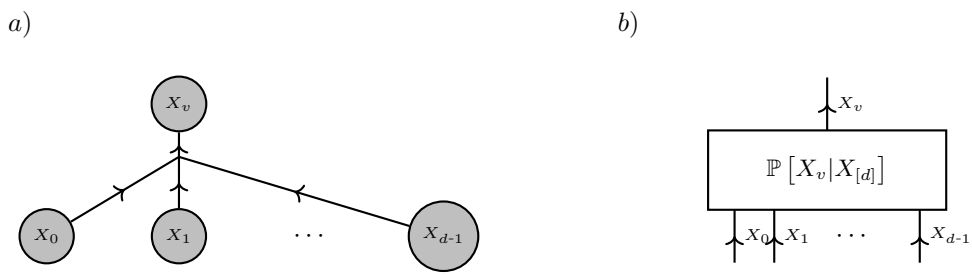


Figure 6: Example of a Factor of a Bayesian Network to the node X_v with parents $X_0, \ldots, X_{d-1}$, as subgraph $a)$ and dual tensor core $b)$.

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

Theorem 9 (Independence Characterization of Bayesian Networks)**.** *A probability distribution* $\mathbb{P}\left[X_{\mathcal{V}}\right]$ *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

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

Since $\prec$ is a topological ordering we have

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

We apply the assumed conditional independence with Corollary 2 and get

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

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

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

Here we have dropped in the third equation all tensors to the descendants, since their marginalization is trivial (which can be shown by a leaf-stripping argument). In the fourth equation we made use of the fact, that any directed path between the non-descendants and the node is through the parents of the node. By Theorem 7, it now follows that X_v is independent of $\mathrm{NonDes}(v)$ conditioned on $\mathrm{Pa}(v)$. □

3.6.2 Example of a Bayesian Network: Hidden Markov Models

We here extend the example of Markov Chains from Theorem 8 to a limited observation of the variables by observations. Let there be the variables X_t (states) and E_t (observations) with a discrete and finite time $t \in [T]$.

The conditional assumptions are

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

Then the probability tensor has the decomposition

$$\mathbb{P}\left[X_{0:T}, E_{0:T}\right] = \prod_{t\in[T]} \left(\mathbb{P}\left[X_t|X_{0:t-1}, E_{0:t-1}\right] \cdot \mathbb{P}\left[E_t|X_{0:t}, E_{0:t-1}\right]\right) \tag{6}$$

$$= \mathbb{P}\left[X_0\right] \cdot \mathbb{P}\left[E_0|X_0\right] \cdot \prod_{t\in[T],t>0} \left(\mathbb{P}\left[X_t|X_{t-1}\right] \cdot \mathbb{P}\left[E_t|X_t\right]\right) \tag{7}$$

Here we used the Chain Rule decomposition of Theorem 5 in the first equation and the conditional independence assumptions in the second.

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

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

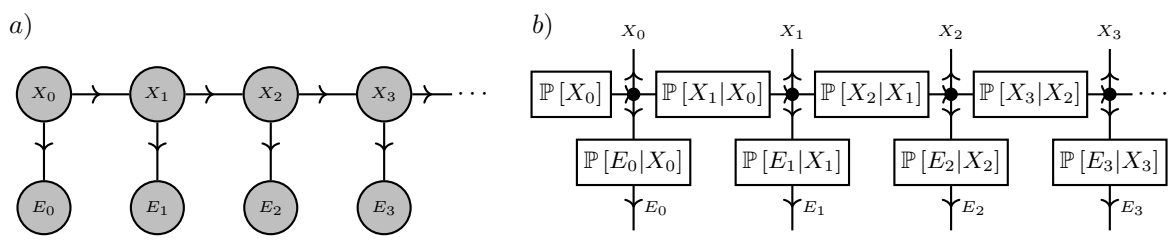


Figure 7: Depiction of a Hidden Markov Model. a) Dependency Graph (of the corresponding chain Graphical Model). b) Dual Tensor Network representing the conditional probability factors.

3.6.3 Markov Networks

While typically Markov Networks are defined on graphs, we define them here on hypergraphs to establish a direct connection to tensor networks defined on the same hypergraph. Along that line, Markov Networks are tensor networks with non-negative tensors (see Definition 8), which are interpreted as probability distributions after normation.

Definition 23. *Let $\mathcal{T}^{\mathcal{G}}$ be a Tensor Network on a hypergraph $\mathcal{G}$. Then the Markov Network to $\mathcal{T}^{\mathcal{G}}$ is the probability distribution of X_v defined by the tensor*

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

We call the denominator

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

the partition function of the Markov Network.

This established a generic relation between Markov Networks and Tensor Networks with coordinatewise non-negative contractions.

Let us now demonstrate how marginalizations and contractions can be done on Markov Networks.

Theorem 10 (Marginalization by Contractions)**.** *The marginalization of a Markov Network to $\mathcal{T}^{\mathcal{G}}$ on the variables of the subset $\tilde{\mathcal{V}}$ is*

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

and the distribution of $\tilde{\mathcal{V}}$ conditioned on $\bar{\mathcal{V}}$ is

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

Proof. From contraction by subcontraction Theorem 77. □

Definition 24 (Separation of Hypergraph)**.** *A path in a hypergraph is a sequence of nodes v_k for $k \in [d]$, such that for any $k \in [d-1]$ we find a hyperedge $e \in \mathcal{E}$ such that $(v_k, v_{k+1}) \subset e$. Given disjoint subsets A, B, C of nodes in a hypergraph $\mathcal{G}$ we say that C separates A and B with respect to $\mathcal{G}$, when any path starting at a node in A and ending in a node in B contains a node in C.*

To characterize Markov Networks in terms of coditional independencies we need to further define the property of clique-capturing.

Definition 25 (Clique-Capturing Hypergraph)**.** *We call a hypergraph $\mathcal{G}$ clique-capturing, when each subset $\tilde{\mathcal{V}} \subset \mathcal{V}$ is contained in a hyperedge, if for any $a, b \in \tilde{\mathcal{V}}$ there is a hyperedge $e \in \mathcal{E}$ with $a, b \in \tilde{\mathcal{V}}$.*

Let us now show a characterization of Markov Networks in terms of conditional independencies, which is analogous to Theorem 9.

Theorem 11. *Given a clique-capturing hypergraph $\mathcal{G}$, the set of positive Markov Networks on the hypergraph coincides with the set of positive probability distributions, such that each for each disjoint subsets of variables A, B, C we have X_A is independent of X_B conditioned on X_C, when C separates A and B in the hypergraph.*

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

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

By Theorem 7, it now follows that X_A is independent of X_B conditioned on X_C. The converse direction, i.e. that positive distributions respecting the conditional indpendence assumptions are representable as Markov Networks, is known as the Hammersley Clifford Theorem, which we will proof later in Section 16.3. □

From the proof of Theorem 11 Markov Networks with zero coordinates still satisfy the conditional independence assumption. However, the reverse is not true, that is there are distributions with vanishing coordinates, which satisfy the conditional independence assumptions, but cannot be represented as a Markov Network (see Example 4.4 in KOLLER).

3.6.4 Bayesian Networks as Markov Networks

Markov Networks are more flexible compared with Bayesian Networks, since any Bayesian Network is a Markov Network, as we will show in the next theorem.

Theorem 12. *Any Bayesian Network on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a Markov Network on a hypergraph $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}})$ with identical nodes and hyperedges consistent of a hyeredge to each node with v being the only outgoing node and*

$$\{\tilde{v} : (\tilde{v}, v) \in \mathcal{E}\}$$

being the incoming nodes. Each hyperedge of the Markov Network is decorated with the conditional probability distribution and the partition function is vanishing.

Proof. Each conditional probability distribution is associated with the hyperedge constructed to the representative node. The contraction of all conditional probability distributions is the Bayesian Network, which corresponds with the constructed Markov Network due to the trivial partition function. □

Theorem 12 states that Markov Networks are more generic compared with Bayesian Networks. While Markov Network allow for any Tensor Cores and norming the probability distribution globally, Bayesian Networks impose a local normation condition on each Tensor Core by demanding it to be a conditional probability tensor. In

our diagrammatic notation, the local normation of Bayesian Networks is highlighted by the directionality of the hypergraph. Generic Markov Networks are on undirected hypergraphs, where no local normation condition applies.

The representation of Bayesian Networks by Markov Networks is of special interest when representing conditional distributions. Bayesian Networks conditioned on evidence are no longer Bayesian Networks on the same graph, but Markov Networks on a hypergraph enriched by the evidence conditioned about.

3.7 Exponential Families

Overwork: Further integration of selection encodings!

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

$$\mathbb{P}\left[X_0, \ldots, X_{d-1} = x\right] \propto \nu(x) \cdot \sum_{l \in [p]} \phi_l(x) \cdot \theta(x)\,.$$

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

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

with

$$\gamma^{\phi}\left[X_{[d]} = x_{[d]}, L_{\phi} = l\right] = \phi_l(x)\,.$$

We then understand θ as a vector to the categorical variable L_{ϕ} and use Theorem 71 to get

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

Definition 26. *Given a sufficient statistics*

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

and a non-negative base measure

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

the set $\Gamma^{\phi,\nu} = \{\mathbb{P}^{(\phi,\theta,\nu)} : \theta[L_{\phi}] \in \mathbb{R}^p\}$ *of probability distributions*

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

is called the exponential family to ϕ*. If the base measure is positive, we define for each member with parameters* θ

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

the associated energy tensor.

We have

$$E^{(\phi,\theta,\nu)}[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] = \sum_{l \in [p]} \phi_l(x_0, \ldots, x_{d-1}) \cdot \theta(l)\,.$$

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

3.7.1 Tensor Network Representation

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

Theorem 13 (Generic Representation of Exponential Families)**.** *Given any base measure ν sufficient statistic ϕ we enumerate for each coordindate $l \in [p]$ the image $\mathrm{im}(\phi_l)$ by a variable X_{ϕ_l} taking values in $[|\mathrm{im}(\phi_l)|]$, given an interpretation map*

$$I^l : [|\mathrm{im}(\phi_l)|] \to \mathrm{im}(\phi_l) \ .$$

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

$$W^l[X_{\phi_l} = x_{\phi_l}] = \exp\left[\theta(l) \cdot I^l(X_{\phi_l})\right]$$

and have

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

Proof. We use an extended image of ϕ by

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

Theorem 84 implies

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

The claim follows from the equation

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

□

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

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

We will show this property in Theorem 85. One strategy to create ρ^ϕ is thus the creation of the encoding of all its coordinate maps. When the coordinate maps are sharing common components, a sparser representation can be derived through encodings of the components shared among the coordinate map encodings.

A tensor network representation of an exponential family is thus a Markov Network consistent of two types of cores. Computation cores are relational encodings of statistics ρ^{ϕ_l}. Our intuition is that they compute the hidden variable X_{ϕ_l}, based on Basis Calculus (see Chapter 17). Activation cores W^l exploit the computed variable and provide, when contracted with the relational encoding, a factor

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

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

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

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

Remark 5 (Comparison of relation and selection encodings)**.** *Relation encodings are in general of higher dimensions than selection encodings. In can thus be intractible to instantiate the probability distribution as a tensor networks, while the energy tensor can still be efficiently represented based on selection encodings. In this case, energy-based reasoning algorithms are tractible while more direct methods are intractible.*

3.7.2 Mean Parameters

Definition 27. *Let there be an exponential family defined by ϕ. We call the tensor*

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

the mean parameter tensor to a member $\mathbb{P}^{(\phi,\theta,\nu)}$ of an exponential family. The map

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

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

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

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

The image $\mathcal{M}$ of the forward map $\mathrm{im}\left(F^{(\phi,\nu)}\right)$ is the interior of the convex polytope $\mathcal{M}$. It contains any mean parameters $\langle\{\mathbb{P}, \phi\}\rangle [L_\phi]$ realizable by any probability distribution $\mathbb{P}$ Wainwright and Jordan (2008).

3.7.3 Examples

Example 6 (The naive exponential family)**.** *When taking as sufficient statistic the identity $\delta\left[X_{[d]}, L\right]$, we can represent any positive distribution $\mathbb{P}$ as a member of the exponential family, namely when choosing the canonical parameter*

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

The associated mean parameter is then

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

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

Theorem 14 (Exponential Representation of Markov Networks)**.** *For any hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we define a sufficient statistics*

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

where

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

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

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

where

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

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

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

$$\prod_{e \in \mathcal{E}} T^e(x_e) = \exp\left[\sum_{e \in \mathcal{E}} \theta_e(x_e)\right] = \exp\left[\sum_{e \in \mathcal{E}} \langle \theta_e, \phi_e(x_{\mathcal{V}})\rangle\right] . \tag{8}$$

Using that

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

we get

$$\langle \{T^e : e \in \mathcal{E}\}\rangle\left[X_{\mathcal{V}}\right] = \exp\left[\langle \theta, \phi\rangle\left[X_{\mathcal{V}}\right]\right] . \tag{9}$$

This implies

$$\langle \{T^e : e \in \mathcal{E}\}\rangle\left[X_{\mathcal{V}}|\varnothing\right] = \langle \exp\left[\langle \theta, \phi\rangle\left[X_{\mathcal{V}}\right]\right]\rangle\left[X_{\mathcal{V}}|\varnothing\right] . \tag{10}$$

□

The mean parameter of the Markov Network exponential family is the cartesian product of the marginals $\mu_e[X_e]$ are often refered to as beliefs in the literature.

3.8 Empirical Distributions

Definition 28. *Given a dataset $\{(x_0^j, \ldots, x_{d-1}^j) \,:\, j \in [m]\}$ of samples of the factored system we define the sample selector map*

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

elementwise by

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

Its relational encoding (see Definition 14) is the tensor

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

which we call a data tensor.

The Data Tensor is a conditional probability tensor and has a network decomposition depicted in Figure 8. The cores ρ^{D_k} are matrices storing the value of the categorical variable X_k in the sample world indexed by j. Whereas the one-hot encoding of single samples is a basis tensor (and therefore a basis elementary decomposition), the data tensor has a basis CP decomposition, see Section 19.1.2). This follows from Theorem 85, using the coordinate maps of D by

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

defined by

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

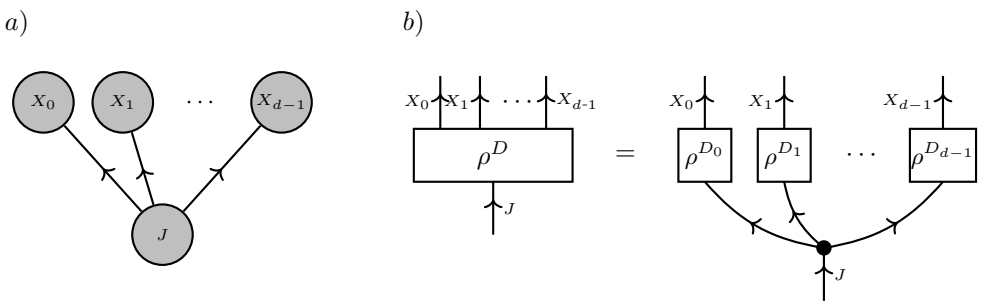


Figure 8: Representation of a dataset. a) As a Bayesian network. b) Decomposition of the data tensor into a tensor network in the CP Format. Without the contraction with the dashed $\frac{1}{m}\mathbb{I}$ core, the datacore encodes the distribution conditioned on a datapoint.

The Data Tensor a conditional probability tensor, which retrieves the respective sample distribution when selecting a sample. We can represent this probability distribution by a random variable X_j selecting a specific datapoint on which the atoms depend.

We define the empirical distribution by the normation of the relational encoding of the datamap as

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

Each coordinate of the empirical distribution is the frequency of the by the index specified word in the data.

Theorem 15. *Given a data map D we have*

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

and

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

Proof. The first claim is a special case of Theorem 85, to be shown in Chapter 17. To show the second claim we notice

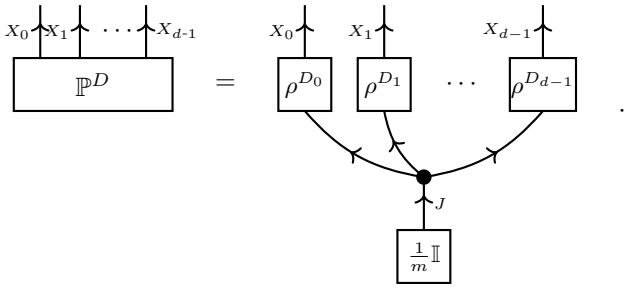
$$\langle \{ \rho^D \} \rangle \left[\varnothing \right] = \sum_{j \in [m]} \langle \{ \rho^D \left[J = j, X_{[d]} \right] \} \rangle \left[\varnothing \right] = m \,.$$

With the first claim it follows that

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

□

In a contraction diagram we represent the empirical distribution by



In another perspective, we can understand $\frac{1}{m}\mathbb{I}$ as the uniform probability distribution over the samples, which is by the map D forwarded to a distribution over $\bigtimes_{k\in[d]}[m_k]$.

4 Probabilistic Reasoning

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

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

One can increase the efficiency of inference algorithms by using approximative contractions. Here, message passing schemes can be applied as to be introduced in Chapter 18.

4.1 Queries

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

4.1.1 Querying by functions

We can formalize queries by retrieving expectations of functions given a distribution specified by probability tensors. We exploit basis calculus in defining categorical variables X_f to tensors f, which are enumerating the set $\mathrm{im}\,(f)$. More details on this scheme are provided in Chapter 17, see Definition 70 therein.

Definition 29. *The marginal query of a probability distribution $\mathbb{P}\left[X_{[d]}\right]$ by a tensor*

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

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

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

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

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

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

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

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

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

This will be shown in more detail in Chapter 17 in Corollary 10.

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

We further denote event queries by

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

where by e_z be denote the one hot encoding of the state z with respect to some enumeration. Let us note that they are further contraction of the queries in Definition 29 since by Theorem 77

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

4.1.2 MAP Queries

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

Definition 30. *Given a tensor T the MAP query is the problem*

$$\operatorname{argmax}_{x_0,\ldots,x_{d-1}} T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] . \tag{11}$$

By coordinate calculus, we notice that

$$T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] \left\langle \left\{T, e_{x_0,\ldots,x_{d-1}}\right\}\right\rangle [\varnothing] . \tag{12}$$

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

$$\max_{x_0,\ldots,x_{d-1}} T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] = \max_{\theta \in \Gamma^{\mathrm{EL}}} \langle \{T, \theta\} \rangle [\varnothing] . \tag{13}$$

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

Remark 6 (MAP queries on energy and probability tensors)**.** *Since the exponential function is monotonic, MAP queries on the energy tensor of an exponential family with uniform base measure are equivalent to MAP queries of their energies.*

4.1.3 Answering queries by energy contractions

Let us now interpret a probability tensor at hand as a member of an exponential family (see Section 3.7), which is always possible when taking the naive exponential family.

Lemma 1. *For any probability distribution $\mathbb{P}$ with $\mathbb{P} = \left\langle \exp\left[E\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$, disjoint subsets $A, B \subset [d]$ with $A \cup B = [d]$ and any x_B we have*

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

Proof. Since no summation is commuted. □

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

We notice, that Lemma 1 does not generalize to situations, where $A \cup B \neq [d]$, since summation over the indices of the variables $[d]/A \cup B$ and contraction do not commute.

Lemma 2. *For any probability distribution $\mathbb{P}$ with $\mathbb{P} = \left\langle \exp\left[E\left[X_{[d]}\right]\right]\right\rangle \left[X_{[d]}|\varnothing\right]$, disjoint subsets $A, B \subset [d]$ and any x_B we have*

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

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

4.2 Sampling based on queries

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

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

4.2.1 Exact Methods

Forward Sampling (see Algorithm 1) uses a chain decomposition (see Theorem 5) of a probability distribution to iteratively sample the variables.

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

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

which is only involving a single core of a Bayesian network. This can be shown using Corollary 11 to be derived in Chapter 17.

Algorithm 1 Forward Sampling

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

$$\mathbb{P}\left[X_k | X_l = x_l \,:\, l < k\right]$$

end for

4.2.2 Approximate Methods

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

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

Algorithm 2 Gibbs Sampling

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

$$\mathbb{P}\left[X_k | X_l = x_l \,:\, l \neq k\right]$$

  end for
end while

Gibbs can be implemented based on the energy tensor E of the probability tensor, as follows form the Lemma 1.

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

4.2.3 Simulated Annealing

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

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

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

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

For any exponential family the transformation

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

can be performed by rescaling the canonical parameters as

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

4.3 Maximum Likelihood Estimation

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

4.3.1 Likelihood and Cross Entropy

Given a datapoint $D(j)$ consisting of the images of the data selecting map D (see Definition 28), the likelihood given a Markov Logic Network is denoted as

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

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

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

It is convenient to apply a logarithm on the objective, which does not influence the optimum when optimizing this quantity. This is especially useful, when investigating the convergence of the objective for $m \to \infty$ (see Chapter 12).

Definition 31. *We define the loss of a distribution $\mathbb{P}$ as*

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

Lemma 3. *Given a data selecting map D and a positive distribution $\mathbb{P}$, we have*

$$\mathcal{L}_D\left(\mathbb{P}\right) = \left\langle\left\{\mathbb{P}^D, \ln\left[\mathbb{P}\right]\right\}\right\rangle\left[\varnothing\right] \ . \tag{14}$$

Proof. We have

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

□

We now state the Maximum Likelihood Problem in the form

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

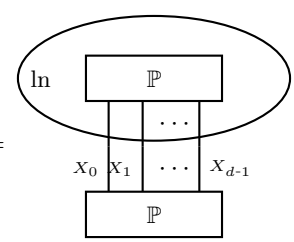
4.3.2 Entropic Interpretation

Definition 32 (Shannon entropy)**.** *The information content or the Shannon entropy of a distribution is defined as*

$$\mathbb{H}\left[\mathbb{P}\right] := \mathbb{E}_{X_{[d]}\sim\mathbb{P}}\left[-\ln\left[\mathbb{P}\left[X_{[d]}\right]\right]\right] = -\sum_{x_{[d]}} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] \cdot \ln\left[\mathbb{P}\left[X_{[d]} = x_{[d]}\right]\right] \ .$$

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

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

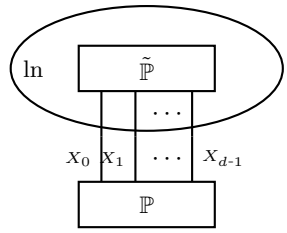


Definition 33 (Cross entropy)**.** *The cross entropy between two distributions is defined as*

$$\mathbb{H}\left[\mathbb{P}, \tilde{\mathbb{P}}\right] := \mathbb{E}_{X_{[d]}\sim\mathbb{P}}\left[-\ln\left[\tilde{\mathbb{P}}[X_{[d]}]\right]\right] = -\sum_{x_0,\ldots,x_{d-1}} \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] \cdot \ln\left[\tilde{\mathbb{P}}[X_{[d]} = x_{[d]}]\right] \ .$$

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

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



Often we have the situation that some entries of $\mathbb{P}$ will be zero and we can thus reduce the contraction of any coordinatewise tensor to these nonzero entries.

The Gibbs inequality states that

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

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

Definition 34 (Kullback Leibler Divergence)**.** *The KL divergence between two distributions is defined as*

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

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

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

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

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

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

$$\mathrm{argmax}_{\mathbb{P} \in \Gamma} \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}\right] \tag{15}$$

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

4.4 Forward Mapping in Exponential Families

Integrate: Selection encodings suffice for variational methods, relational encodings of statistics are required for markov network instantiations of exponential families.

Mean parameter coordinates are expectation queries to ϕ_l, by

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

Forward mappings have a closed form representation by

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

Following Wainwright and Jordan (2008), we can characterize the forward mapping to exponential families as a variational problem and provide an alternative characterization to this contraction.

4.4.1 Variational Formulation

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

We have

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

where

$$\mathcal{M} = \left\{ \left\langle \left\{ \mathbb{P}, \gamma^{\phi} \right\} \right\rangle \left[L\right] \,:\, \mathbb{P} \text{ a distribution} \right\}$$

and by $\mathbb{P}^{\mu}$ we denote a probability tensor reproducing the mean parameter μ.

In Wainwright and Jordan (2008): Forward mapping coincides with gradient, i.e. $\mu = \nabla A(\theta)$.

4.4.2 Mean Field Method

Restrict the maximum over the mean parameters of efficiently contractable distributions and get a lower bound.

We rewrite

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

where

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

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

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

Algorithm 3 is the alternation of legwise updates until a stopping criterion is met.

Algorithm 3 Naive Mean Field Approximation

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

$$V^k[X_k] = \mathbb{I}[X_k]$$

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

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

  end for
end while

More generically, one can use any Markov Network as the approximating family. Let $\mathcal{G}$ be any hypergraph, we define the problem

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

Theorem 16. *The Markov Network with hypercores $\{T[e] : e \in \mathcal{E}\}$ is a stationary point for Problem $\mathrm{P}_{\Gamma^{\mathcal{G}}, \mathbb{P}}$, if for all $e \in \mathcal{E}$*

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

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

Proof. By first order condition on the objective. □

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

Theorem 17. *For any hypergraph $\mathcal{G}$ and energy tensor E we have*

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

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

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

4.4.3 Mode Search by annealing

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

This is because

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

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

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

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

4.5 Backward Mapping in Exponential Families

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

4.5.1 Variational Formulation

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

Lemma 4. *Let there be a sufficient statistic ϕ. The map $B^{(\phi,\nu)} : \mathbb{R}^p \to \mathbb{R}^p$ defined as*

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

is a backward mapping.

Proof. From duality, see Theorem 3.4 in Wainwright and Jordan (2008). □

In Wainwright and Jordan (2008): The objective is the conjugate dual A^* of A, and backward mapping has an expression by the gradient, i.e. $\theta = \nabla A^*(\mu)$.

Theorem 18 (Moment Matching Criteria)**.** *We have that θ is a solution of the backward problem at μ^*, if and only if*

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

The equation in seeTheorem **??** are called moment matching, since the moment of the empirical distribution is matched by the moment of the fitting distribution.

4.5.2 Connection with Maximum Likelihood Estimation

Backward mapping coincides with the Maximum Likelihood Estimation as we show next.

We now take Γ to coincide with an exponential family $\Gamma^{\phi,\nu}$ for a sufficient statistic ϕ.

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

Lemma 5. *Let $\phi \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k} \otimes \mathbb{R}^p$ be a sufficient statistic and $\mathbb{P}^* \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ a probability distribution. For any member $\mathbb{P}^{(\phi,\theta,\nu)} \in \Gamma^{\phi,\nu}$ we have*

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

where

$$\mu^* = \langle \{\mathbb{P}^*, \phi\} \rangle [X_\phi]$$

and

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

The M-projection of $\mathbb{P}^*$ onto $\Gamma^{\phi,\nu}$ is $\mathbb{P}^{(\phi,\hat{\theta},\nu)}$ for

$$\hat{\theta} \in \operatorname{argmax}_{\theta} \langle \theta, \mu^* \rangle [\varnothing] - A(\theta).$$

Proof. By decomposing

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

we get

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

If follows that

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

□

Parameter Estimation is the M-Projection of a distribution onto the exponential family.

Theorem 19 (Wainwright and Jordan (2008))**. ??** *Given any probability distribution $\mathbb{P}\left[X_{[d]}\right]$ and a exponential family defined by the sufficient statistic ϕ, the M-Projection onto the family is the distribution $\mathbb{P}^{(\phi,\hat{\theta},\nu)}$ where*

$$\hat{\theta} = B^{(\phi,\nu)}(\langle\mathbb{P},\phi\rangle\left[X_{\phi}\right])\,.$$

Proof. $\langle\mathbb{P},\phi\rangle\left[X_{\phi}\right]$ is in $\operatorname{im}\left(F^{(\phi,\nu)}\right)$ and MLE has a variational characterization with maximum at the dual $\hat{\theta}$, see Wainwright and Jordan (2008). □

4.5.3 Connection with Maximum Entropy

The Maximum entropy problem with respect to matching expected statistics is

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

where the optimization is over all probability distributions $\mathbb{P}$.

Theorem 20. *Let ϕ be a map and $\mathbb{P}^{*}$ be any distribution of $\bigtimes_{k\in[d]}[2]$ and define*

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

Then the solution of $\mathrm{P}_{\mathbb{H}[],\mu^{*}}$ coincides with the member $\mathbb{P}^{(\phi,\hat{\theta},\nu)}$ of the exponential family $\Gamma^{\phi,\nu}$ where

$$\hat{\theta} = B^{(\phi,\nu)}(\mu^{*})$$

for a backward map $B^{(\phi,\nu)}$ of $\Gamma^{\phi,\nu}$.

Proof. Classical result based on duality of maximum entropy and maximum likelihood, shown e.g. in Koller Book. □

Theorem 20 states, that when the maximum entropy problem has a solution (i.e. $\mu^{*}\in\mathcal{M}$), then the solution is in the exponential family to the statistic ϕ. This is a main motivation of the usage of exponential families as models.

4.5.4 Alternating Algorithms to Approximate the Backward Map

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

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

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

This condition is equal to the collection of moment matching equations (see Theorem **??**)

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

Lemma 6. *For any sufficient statistic ϕ a parameter vector θ and a $l \in [p]$ we define*

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

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

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

Proof. We have

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

and

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

Here we used

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

and redundancies of copies of relational encodings. It follows that

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

is equal to

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

□

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 $\mathrm{im}\left(\phi_l\right)$ 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 Section 10.2.

The computation of T_l in Algorithm 4 can be intractable and be replaced by an approximative procedure based on message passing schemes.

5 Propositional Logics

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

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

Algorithm 4 Alternating Moment Matching

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

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

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

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

  end for
end while

5.1 Encoding of Booleans

Propositional logic amounts to reason about Boolean variables, which are categorical variables with 2 possible values. Before applying this insight in the representation of propositional formulas, we first investigate how Boolean calculus can be represented by multilinear operations.

5.1.1 Booleans as categorical variables

To represent Booleans by categorical variables X with two states we use the following standard group homomorphism

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

by the map

$$[\cdot] : \{\mathrm{True}, \mathrm{False}\} \to \{1, 0\}$$

defined as

$$[\mathrm{True}] = 1 \quad , \quad [\mathrm{False}] = 0 \, .$$

The multiplication is performed in the binary tensor contractions and can thus be interpreted as the $\wedge$ connective performed on Boolean coordinates.

While the conjunction of is in this embedding performed by multiplications, operations like the negation

$$[\neg X] = 1 - [X]$$

are affine linear. Direct applications of these affine linear operations to perform logical calculus will be discussed later in Section **??**.

However, in this chapter we will circumvent the problems arising with affine linearity by using the one-hot encoding of Booleans.

5.1.2 One-hot Encoding

Booleans are categorical variables with $m = 2$ states, where we interpret the states $[2] = \{0, 1\}$ by $\{\mathrm{True}, \mathrm{False}\}$. The one-hot encoding of Booleans

$$e : [2] \to \{e_0[X], e_1[X]\} \subset \mathbb{R}^2$$

is thus understood as an encoding of truth values.

As discussed before, the one-hot encoding is rich enough to represent any function of the state by a linear function on the encoding. The truth of formulas is a function of the truth of atomic formulas, and thus representable by linear functions of the one-hot encodings.

5.2 Semantics of Propositional Formulas

The epistemological commitments 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.

5.2.1 Formulas

Logics is especially strong in interpreting binary 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 35. *A propositional formula $f[X_0, \dots, X_{d-1}]$ depending on d atoms X_k is a tensor*

$$f[X_0, \dots, X_{d-1}] : \bigtimes_{k \in [d]} [2] \to [2] \subset \mathbb{R} .$$

We call $x_0, \dots, x_{d-1} \in \bigtimes_{k \in [d]} [2]$ a model of a propositional formula f, if

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

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

The propositional formulas coincide therefore with the binary tensors (see Definition 11).

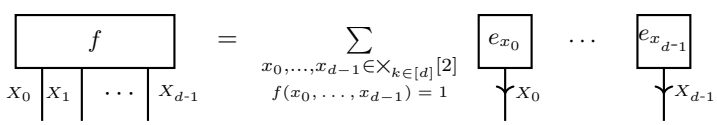
Since propositional formulas are binary valued tensors, the generic decomposition of Lemma 23 simplifies to

$$f[X_0, \dots, X_{d-1}] = \sum_{x_0, \dots, x_{d-1} \in \bigtimes_{k \in [d]} [2]} f[X_0 = x_0, \dots, X_{d-1} = x_{d-1}] \cdot e_{x_0, \dots, x_{d-1}}[X_0, \dots, X_{d-1}] \tag{16}$$

$$= \sum_{x_0, \dots, x_{d-1} \in \bigtimes_{k \in [d]} [2] \,:\, f[X_0 = x_0, \dots, X_{d-1} = x_{d-1}] = 1} e_{x_0, \dots, x_{d-1}}[X_0, \dots, X_{d-1}] . \tag{17}$$

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 17 (see Definition 67).

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.

5.2.2 Relational encoding of formulas

There are two ways to represent formulas by tensors. One way is to understand $[2]$ as subset of $\mathbb{R}$ and interpreting the formula directly as a tensor (as in Definition 35). Another way is to understand $[2]$ as the possible values of a categorical variable. Following this second perspective, formulas are maps between factored systems, where the image system is the factored systems of atoms and the target system the atomic system defined by a variable X_f representing the formula satisfaction. We can then build the relational encoding (Definition 14) of that map to represent the formula (see Figure 9).

Definition 36 (Relation Encoding of Formulas)**.** *Given a factored system with d atoms $X_0, \dots, X_{d-1}$ and a propositional formula f, we define the relational encoding of f (see Definition 14)*

$$\rho^f[X_0, \dots, X_{d-1}, X_f] \in \left(\bigotimes_{k \in [d]} \mathbb{R}^2 \right) \otimes \mathbb{R}^2$$

by

$$\rho^f[X_0, \dots, X_{d-1}, X_f] = \sum_{x_0, \dots, x_{d-1} \in \bigtimes_{k \in [d]} [2]} e_{x_0, \dots, x_{d-1}}[X_0, \dots, X_{d-1}] \otimes e_{f(x_0, \dots, x_{d-1})}[X_f] . \tag{18}$$

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

Lemma 7. *For any formula f we have*

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

In particular

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

Proof. We can decompose relational encodings of formulas into the sum (see Figure 9)

$$\rho^f = e_0 \otimes \left(\sum_{x_0,\ldots,x_{d-1}\,:\,f(x_0,\ldots,x_{d-1})=0} e_{x_0,\ldots,x_{d-1}} \right) \tag{19}$$

$$+ e_1 \otimes \left(\sum_{x_0,\ldots,x_{d-1}\,:\,f(x_0,\ldots,x_{d-1})=1} e_{x_0,\ldots,x_{d-1}} \right) \tag{20}$$

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

Compared with the direct interpretation of a formula as a tensor and the decomposition into models in Equation 16, we notice that the relational encoding also represents encoding of worlds where the formula is not satisfied. This representation is required to represent arbitrary propositional formulas by contracted tensor networks of its components, as will be investigated in the following sections.

The relational decomposition ρ^f has coordinates

$$\left\langle \left\{\rho^f, e_{x_0,\ldots,x_{d-1}}\right\}\right\rangle \left[X_f\right] = \begin{cases} e_1 & \text{if the world where } X_k = x_k \text{ is a model of } f \\ e_0 & \text{else}\,. \end{cases} \tag{21}$$

The contractions of the relational encoding therefore calculate whether an assignment of atoms is a model of the formula, using basis calculus (see Theorem 82).

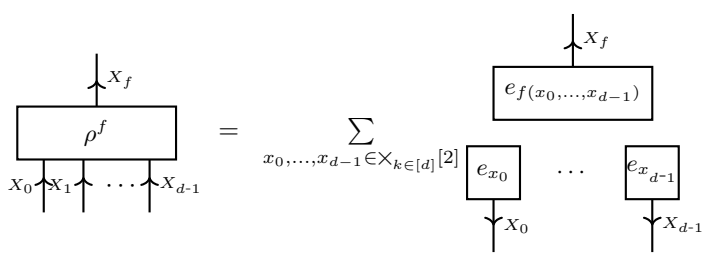


Figure 9: Relational encoding of a propositional formula. The encoding is a sum of the one hot encodings of all states of the factored system in a tensor product with basis vectors, which encode whether the state is a model of the formula. The tensor is directed, since any contraction with an encoded state results in the basis vector evaluating the formula, which we called basis calculus.

5.3 Syntax of Propositional Formulas

Relational encodings of propositional formulas are especially useful when representing function compositions by the representation of their components (see Theorem 83). In propositional logics, the syntax of defining propositional formulas is oriented on compositions of formulas by connectives. We in this section investigate the decomposition schemes of relational encodings into tensor networks of component encodings for binary tensors following propositional logic syntax.

5.3.1 Atomic Formulas

We call atomic formulas the most granular formulas, which are not splitted into compositions of other formulas. Our syntactic decomposition of propositional formulas will then investigate, how any propositional formula can be represented by these.

Definition 37. *The tensors $f_k\left[X_0,\dots,X_{d-1}\right]$ defined for $x_0,\dots,x_{d-1}$ as*

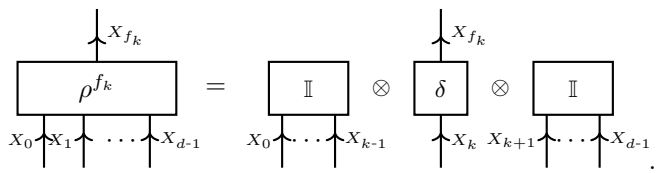
$$f_k\left[X_0=x_0,\dots,X_{d-1}=x_{d-1}\right]=x_k$$

are called atomic formulas.

Theorem 21. *The relational encoding of any atomic formula $f_k\left[X_0,\dots,X_{d-1}\right]$ has a tensor decomposition by*

$$\rho^{X_k}\left[X_0,\dots,X_{d-1},X_{f_k}\right]=\langle\{\delta\left[X_k,X_{f_k}\right]\}\rangle\left[X_0,\dots,X_{d-1},X_{f_k}\right]\,.$$

The decomposition is depicted in a network diagram as



Proof. We have by definition

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

□

Remark 7 (Representation of atomic formula tensors for connective action)**.** *Need to represent this as $\langle\delta\otimes\mathbb{I},e_1^k\rangle$, where the bracket indicates contraction along the kth axis. The core e_1^k can be replaced by further operations based on logical connectives. The axis k is changed from an axis associated with an atom truth to an axis associated with an formula truth.*

5.3.2 Syntactical combination of formulas

Formula tensors are elements of tensor spaces with $d+1$ axis. The number of coordinates thus grows exponentially with the number of atoms, which is

$$\dim\left[\mathbb{R}^2\otimes\bigotimes_{k\in[d]}\mathbb{R}^2\right]=2^{d+1}\,.$$

When the number of atoms in a theory is large, the naive representation of formula tensors will be thus intractable. In contrast, most logical formulas appearing in practical knowledge bases are sparse in the sense that they have short representations in a logical syntax. Motivated by this consideration we now discuss propositional syntax and investigate the sparse decomposition of formula tensors along their formula structure to avoid the curse of dimensionality.

In logical syntax formulas are described by atomic formulas recursively connected via connectives. We show, that representations of logical connectives $\circ\in\{\neg,\wedge,\vee,\oplus,\Rightarrow,\Leftrightarrow\}$ can be represented by feasible tensor cores ρ° contracted along a tensor network.

Example 7. *We use the following connectives:*

- *negation* $\neg:[2]\to[2]$ *by the vector*

$$\neg[X_f]=\begin{bmatrix}0\\1\end{bmatrix}\tag{22}$$

- *conjunctions* $\wedge:[2]\times[2]\to[2]$

$$\wedge[X_f,X_h]=\begin{bmatrix}0&0\\0&1\end{bmatrix}\tag{23}$$

- *disjunctions* $\vee : [2] \times [2] \to [2]$

$$\vee[X_f, X_h] = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \tag{24}$$

- *exact disjunction* $\oplus : [2] \times [2] \to [2]$

$$\oplus[X_f, X_h] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{25}$$

- *implications* $\Rightarrow$: $[2] \times [2] \to [2]$

$$\Rightarrow [X_f, X_h] = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{26}$$

- *biimplication* $\Leftrightarrow$: $[2] \times [2] \to [2]$

$$\Leftrightarrow [X_f, X_h] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{27}$$

Lemma 8. *Let there be formulas f and h depending on categorical variables $X_{[d]} = X_0, \dots, X_{d-1}$ and a map*

$$\circ : [2] \times [2] \to [2] \,.$$

Then we have

$$\rho^{f \circ h} \left[X_{[d]}, X_{f \circ h}\right] = \left\langle \{\rho^{\circ} \left[X_f, X_h, X_{f \circ h}\right], \rho^{f} \left[X_{[d]}, X_f\right], \rho^{h} \left[X_{[d]}, X_h\right]\}\right\rangle \left[X_{[d]}, X_{f \circ h}\right]$$

and for any map $\circ : [2] \to [2]$

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

Proof. This follows from Theorem 83 to be shown in Chapter 17. □

Theorem 22 (Composition of Formulas)**.** *Let there be a set of binary variables $X_{\mathcal{V}}$ including atoms $X_0, \dots, X_{d-1}$ and image variables to some formulas. For any formula $f\,[X_0, \dots, X_{d-1}]$, which has a syntactical composition into connectives $\{\circ_l [X_{\{\mathcal{V}_l\}}] : l \in [p]\}$ taking their inputs by variables $X_{\{\mathcal{V}_l\}} \subset X_{\mathcal{V}}$ and output by a variable $X_{\circ_l}$ we have that*

$$\rho^{f} \left[X_0, \dots, X_{d-1}, X_f\right] = \left\langle \left\{\rho^{\circ_l} \left[X_{\{\mathcal{V}_l\}}, X_{\circ_l} : l \in [p]\right]\right\}\right\rangle \left[X_0, \dots, X_{d-1}, X_f\right] \,.$$

Proof. When a variable in $X_{\mathcal{V}}$ appears multiple times as input to connectives, we replace it by a set of copies (which wont change the contraction, since all tensors are binary and Theorem 79 can be applied). The claim follows then from iterative application of Lemma 8. □

Remark 8 (d-ary connectives such as $\wedge$ and $\vee$)**.** *Since the decomposition of relational encoding can be applied to generic function compositions (see Theorem 83), we can also allow for d-ary connectives*

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

in Theorem 83 The connectives $\wedge$ and $\vee$ satisfy associativity and have thus straightforward generalizations to the d-ary case. This is because associativity can be exploited to represent the relational encoding by any tree-structured composition of binary $\wedge$ and $\vee$ connectives.

In general, any $d$ary logical connective is a map

Propositional syntax consists in the application of connectives on atomic formulas, and recursively on the results of such constructions. When passed towards connective cores, atomic formula tensors act trivial on the legs and just identify the corresponding atomic formula index x_{X_k} with x_k. This is due to the fact, that the Hadamard product with the trivial tensor $\mathbb{I}$ leaves any tensor invariant, and the contraction with the elementary matrix δ identifies indices with each other. We can thus savely ignore the atomic formula tensors appearing in the decomposition of formula tensors to non-atomic formulas. An example of such a decomposition is depicted in Figure 11.

Remark 9 (HT Interpretation of Formula Tensor Networks)**.** *The sketched decomposition of the formula tensor into a network is a hierarchical tree decomposition of the formula tensor, which we will describe in more detail in Section 19.4. At each decomposition of a formula into subformulas, two subspaces spanned by the respective atomic spaces are selected.*

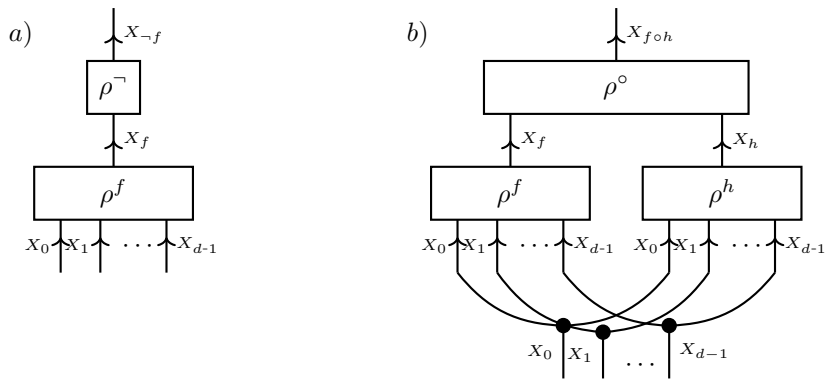


Figure 10: a) Relational encoding of a negated formula f as a tensor network of the encoded formula and the encoded connective $\neg$. b) Relational encoding of a composition of formulas f, h by a connective $\circ \in \{\wedge, \vee, \oplus, \Rightarrow, \Leftrightarrow\}$. The encoding is a contraction of encodings to f, h and $\circ$.

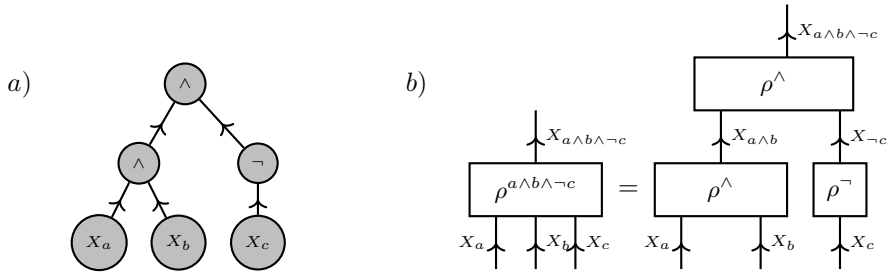


Figure 11: Decomposition of the formula tensor to $f = a \wedge b \wedge \neg c$ into unary (matrix) and binary (third order tensor) cores. a) Visualization of f as a graph. b) Dual Tensor Network decomposition of f. We can make use of the invariance of a Hadamard product with a constant tensor $\mathbb{I}$ and thus not draw axis to atoms not affected by a formula.

5.3.3 Syntactical decomposition of formulas

We have seen how the decomposition of complex formulas into connectives acting on the component formulas can be exploited to find effective representations of the semantics by tensor networks. Here the question arises here, how to perform such decompositions in case of a missing syntactical representation of a formula. By Definition 35 any binary tensor is a formula. We show in the following, how we can find a syntactic specification of a formula given its tensor.

Definition 38 (Terms and Clauses)**.** *Given two disjoint subsets $\mathcal{V}_0$ and $\mathcal{V}_1$ of the $[d]$, the corresponding term is the formula defined on the indices $x_0, \dots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ by*

$$Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1} = \left(\bigwedge_{k\in\mathcal{V}_0} \neg f_k\right) \wedge \left(\bigwedge_{k\in\mathcal{V}_1} f_k\right)$$

and the corresponding clause is the formula defined on the indices $x_0, \dots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ by

$$Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1} = \left(\bigvee_{k\in\mathcal{V}_0} f_k\right) \vee \left(\bigvee_{k\in\mathcal{V}_1} \neg f_k\right),$$

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 9. *Terms are contractions of one-hot encodings, that is for any disjoint subsets $\mathcal{V}_0, \mathcal{V}_1 \subset [d]$ we have*

$$Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1}[X_{[d]}] = \left\langle e_{\{x_k=0:k\in\mathcal{V}_0\}\cup\{x_k=1:k\in\mathcal{V}_1\}}\right\rangle \left[X_{[d]}\right] .$$

Clauses are substractions of one-hot encodings from the trivial tensor, that is for any disjoint subsets $\mathcal{V}_0, \mathcal{V}_1 \subset [d]$ we have

$$Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1}[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - \left\langle e_{\{x_k=0:k\in\mathcal{V}_0\}\cup\{x_k=1:k\in\mathcal{V}_1\}}\right\rangle \left[X_{[d]}\right] .$$

The reference of the formulas in the case $\mathcal{V}_0\dot\cup\mathcal{V}_1 = [d]$ as minterms and maxterms is due to the fact, that minterms are formulas with unique models and maxterms are formulas with a unique world not satisfying the formula. We use this insight and enumerate maxterms and minterms by the index $x \in \bigtimes_{k\in[d]}[2]$ of the unique world where the minterm is satisfied, respectively the maxterm is not satisfied. For any $\mathcal{V}_0\dot\cup\mathcal{V}_1 = [d]$ we take the index tuple $x_0, \ldots, x_{d-1}$ where $x_k = 0$ if $k \in \mathcal{V}_0$ and $x_k = 1$ if $k \in \mathcal{V}_1$ and define

$$Z^{\vee}_{x_0,\ldots,x_{d-1}} = Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1} \quad \text{and} \quad Z^{\wedge}_{x_0,\ldots,x_{d-1}} = Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1} .$$

Corollary 3. *Minterms are basis elements of the tensor space, that is for any $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ we have*

$$Z^{\wedge}_{x_0,\ldots,x_{d-1}} = e_{x_0,\ldots,x_{d-1}}$$

Maxterms are substraction of basis elements from the trivial tensor, that is for any $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ we have

$$Z^{\vee}_{x_0,\ldots,x_{d-1}} = \mathbb{I}\left[X_{[d]}\right] - e_{x_0,\ldots,x_{d-1}} .$$

Proof. Follows from Lemma 9, since when $\mathcal{V}_0 \cup \mathcal{V}_1 = [d]$ the contractions 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 23. *For any binary tensor $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]}\mathbb{R}^2$ with two-dimensional axes we have*

$$T\left[X_{[d]}\right] = \left(\bigvee_{x_0,\ldots,x_{d-1}:T\left[X_{[d]}=x_{[d]}\right]=1} Z^{\wedge}_{\{k:x_k=0\},\{k:x_k=1\}}\right)\left[X_{[d]}\right]$$

and

$$T\left[X_{[d]}\right] = \left(\bigwedge_{x_0,\ldots,x_{d-1}:T\left[X_{[d]}=x_{[d]}\right]=0} Z^{\vee}_{\{k:x_k=0\},\{k:x_k=1\}}\right)\left[X_{[d]}\right] .$$

Proof. To show the representation by minterms we use the decomposition

$$T\left[X_{[d]}\right] = \sum_{x_{[d]}:T\left[X_{[d]}=x_{[d]}\right]=1} e_{x_{[d]}}\left[X_{[d]}\right]$$

and notice that each term in the disjunction modifies the formula by adding respective world $x_{[d]}$ to the models of the formula. To show the representation by maxterms we use the decomposition

$$T\left[X_{[d]}\right] = \mathbb{I}\left[X_{[d]}\right] \quad - \sum_{x_{[d]}:T\left[X_{[d]}=x_{[d]}\right]=0} e_{x_{[d]}}\left[X_{[d]}\right]$$

and notice that each term in the conjunction modifies the formula by removing the respective world $x_{[d]}$ from the models of the formula. Thus, both decompositions are propositional formulas with the same set of models as the formula T and are thus identical to T. □

The decompositions found in Theorem 23 are also called canonical normal forms to propositional formulas $T\left[X_{[d]}\right]$.

Remark 10 (Efficient Representation in Propositional Syntax)**.** *The decomposition in Theorem 23 is a basis CP decomposition of the binary tensor and will further be investigated in Chapter 19. The formulas constructed in the proof of Theorem 23 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 Theorem 23. We will provide such constructions in Chapter 19, where we show that dropping the demand of directionality and investigating binary CP Decompositions will improve the sparsity of the polynomial formula representation.*

5.4 Discussion and Outlook

Further study of representing Knowledge Bases based on Tensor Networks of its formulas in Chapter 9 (see Theorem 40).

6 Logical Inference

We approach logical inference by defining probability distributions based on propositional formulas and then apply the methodology introduced in the more generic situation of probabilistic inference. Logical approaches pay here special attention to situations of certainty, where a state of a variable has probability 1. In this situation, we say that the corresponding formula is entailed.

We start the discussion by showing how formulas can be interpreted by distributions and define logical entailment based on corresponding probabilistic queries. This enables us to define logical entailment based on the resulting conditional distributions.

Remark 11 (Interpretation of Contractions in Logical Reasoning)**.** *The coordinates of contracted binary tensor networks describe whether the by the coordinate indexed world is a model of the Knowledge Base at hand. Contractions, which only leave a part variables open, store the counts of the world respecting conditions given by the choice of slices. When contracting without open variables, we thus get the total worldcount.*

This is consistent with the probabilistic interpretation of contractions, when applying the frequentist interpretation of probability and defining normed worldcounts as probabilities.

6.1 Entailment in Propositional Logics

Definition 39 (Entailment of propositional formulas)**.** *Given two propositional formulas $\mathcal{KB}$ and f we say that $\mathcal{KB}$ entails f, denoted by $\mathcal{KB} \models f$, if any model of $\mathcal{KB}$ is also a model of f, that is*

$$\forall_{x_{[d]} \in \times_{k \in [d]}[2]} \left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

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

Entailment can be understood by subset relations of the models of formulas. This perspective can be applied with subset encodings in Chapter 17.

6.1.1 Deciding Entailment by contractions

Theorem 24 (Contraction Criterion of Entailment)**.** *We habe $\mathcal{KB} \models f$ if and only if*

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

Proof. If for a $x_0, \ldots, x_{d-1}$ we have $\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1$ but not $\left(f(x_0, \ldots, x_{d-1}) = 1\right)$, the contraction would be at least 1. Conversely, if the contraction is at least one, we would find $x_0, \ldots, x_{d-1}$ with $\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1$ and $\neg f\left[X_{[d]} = x_{[d]}\right] = 1$, therefore $f\left[X_{[d]} = x_{[d]}\right] = 0$. It follows that $\mathcal{KB} \models f$ does not hold. □

To decide whether a formula is entailed, or its negation is entailed (in which case one says that the formula is contradicted) by a single contraction, one can perform the contraction

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

and use that

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

and

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

6.1.2 Contraction Knowledge Base

We now show how to implement a propositional Knowledge Base with the TELL and ASK operations based on Theorem 4.

Algorithm 5 Contraction Knowledge Base

```
ASK(formula f)
  T[X_f] ← ⟨{KB, ρ^f}⟩ [X_f]
  if T[X_f = 0] = 0 then
      return Entailed
  end if
  if T[X_f = 1] = 0 then
      return Contradicted
  end if
  return Contingent
TELL(formula f)
  if ASK(f) returns Contingent: then
      KB ← KB ∧ f
  end if
```

Comment: TELL checks whether the formula to be added is entailed, in which case it is redundant to add, and whether the formula to be added is contradicted, in which case the knowledge base would become unsatisfiable.

6.1.3 Sparse Representation of a Knowledge Base

Let us now investigate how to sparsely represent a Knowledge Base. Towards getting insights on this we first show that entailed formulas can be dropped from the Knowledge Base.

Theorem 25. *If and only if $\mathcal{KB} \models f$ we have*

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

Proof. For any world indexed by a coordinate $x_0, \ldots, x_{d-1}$, $\mathcal{KB}\left[X_{[d]} = x_{[d]}\right]$ indicates whether the world is a model of $\mathcal{KB}$. We have entailment, when the models of $\mathcal{KB} \cup f$ coincide with those of $\mathcal{KB}$. □

Remark 12 (Sparsest Description of a Knowledge Base)**.** *Given a set of worlds indexed by T, find the sparsest set of formulas $\mathcal{KB}$ such that*

$$T = \mathcal{KB}$$

would be benefitial for small computational complexity. Since the formula tensors are invariant under entailment, we can drop entailed formulas.

6.2 Formulas as Random Variables

Aim here: Relate with the probabilistic reasoning concepts of marginal and conditional distributions.

Given a probability distribution $\mathbb{P}$ of atoms we add a variable by building the Markov Network of $\mathbb{P}$ and ρ^f to get a joint distribution of the atoms and a query formula f

There are two ways of interpreting formula tensors as conditional probabilities. The standard one, which we also used above, understands the atomic legs as conditions and calculates the truth of the formula. Another understands a formula as a condition.

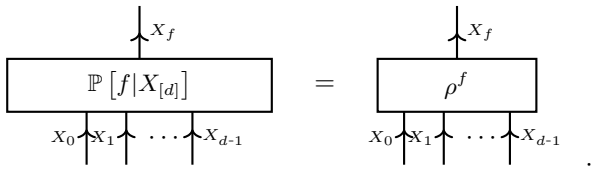
6.2.1 Conditioning on the atoms

Our main interpretation understands each tuple of indices $x_0, \ldots, x_{d-1}$ as conditions of a probability tensor. Given a truth assignment to the atomic variables X_k, that is a choice of indices x_k, determines the truth of the formula. We thus interpret the formula tensors as defining a conditional probability of f given the atoms X_k indexed by x_k.

Theorem 26. *The relational encoding of any propositional formula f coincides with the conditional probability of that formula conditioned on the identity on the atoms, that is*

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

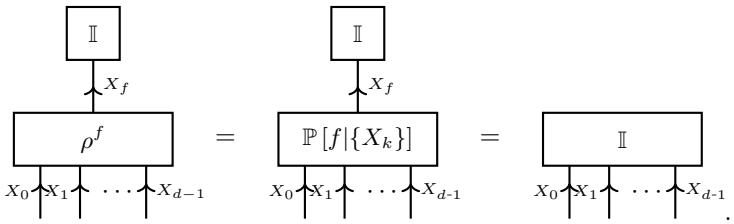
We depict this by



Proof. The distribution $\mathbb{P}$ does not influence the conditional query, since the normation acts on any state. □

The conditional query $\mathbb{P}\left[X_f | X_0, \ldots, X_{d-1}\right]$ provides an interpretation of ρ^f as a conditional probability. This is also reflected in the fact that both $\mathbb{P}\left[f|X_{[d]}\right]$ and ρ^f are directed, since the first is a normation by Defintion 29 and the second a encoding of a function.

This directly implies using Theorem **??** the trivialization of the formula tensor when contracting its head axis indexed by x_f with the trivial vector $\mathbb{I}$, depicted as

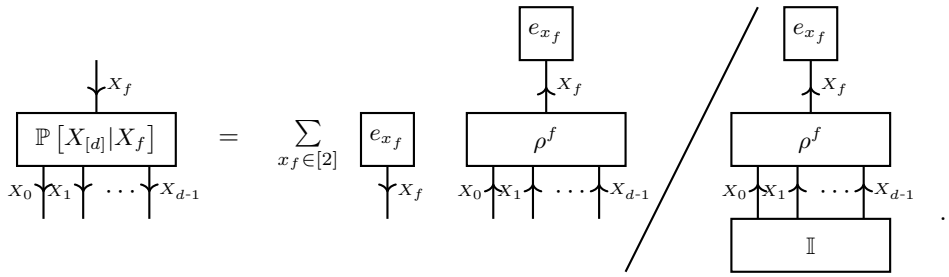


6.2.2 Conditioning on the formula

Let us now converse the order of conditioning from $\mathbb{P}\left[f|X_{[d]}\right]$ to $\mathbb{P}\left[X_{[d]}|f\right]$. In this way, we have propositonal formulas defining probability distributions on the factored system of atoms.

Given a Markov Network $\mathbb{P}$ with a single core ρ^f for a propositional formula f. By definition we have

$$\mathbb{P}\left[X_0, \ldots, X_{d-1} | X_f\right] = \left\langle \left\{\rho^f\right\} \right\rangle \left[X_0, \ldots, X_{d-1} | X_f\right] .$$



Let us further investigate the slices of $\mathbb{P}\left[X_{[d]}|f\right]$ with respect to f, which define distributions of the states of the factored system. To this end, let us condition on the event of $f = 1$, for which we have the distribution

$$\mathbb{P}\left[X_{[d]}|f = 1\right] = \frac{1}{\langle\{f\}\rangle\left[\varnothing\right]} \sum_{x_0,\ldots,x_{d-1} \in \times_{k\in[d]}[2] \,:\, f\left[X_{[d]}=x_{[d]}\right]=1} e_{x_0,\ldots,x_{d-1}} . \tag{28}$$

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 $f = 0$ we get a uniform distribution of the models of $\neg f$.

The probability distribution in Equation (28) is well defined except for the case that $\langle\{f\}\rangle[\varnothing] = 0$. In this case we have $f = 0$ and call f unsatisfiable, since it has no models.

From an epistemological point of view, probability theory is a generalization of logics, since we allow for probability values in the interval $[0, 1]$. The set of distributions being constructed by conditioning on propositional formulas as in Equation (28) correspond within the set of probability distributions with those having constant coordinates on their support. While the probability tensors with nonvanishing coordinates build a $2^d - 1$-dimensional manifold, where the formulas parametrize 2^{2^d} probability tensors, most of which having vanishing coordinates.

6.2.3 Probability of a function given a Knowledge Base

We can now combine the ideas of the previous two subsections and define probabilities of formulas f given the satisfaction of another formula $\mathcal{KB}$, which we call a Knowledge Base. We have by Theorem 26

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

Of special interest is the marginal probability of X_f given that $X_{\mathcal{KB}}$ is satisfied, that is

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

Remark 13 (Case of Unsatisfiable Knowledge Bases)**.** *When the Knowledge Base is not satisfiable, one cannot normate it and the probability distribution is not dedfined.*

Theorem 27. *Given a satisfiable formula $\mathcal{KB}$, we have $\mathcal{KB} \models f$, if and only if*

$$\mathbb{P}\left[X_f = 0|X_{\mathcal{KB}} = 1\right] = 0\,.$$

Proof. Since $\mathcal{KB}$ is satisfiable, we have $\langle\{\mathcal{KB}\}\rangle[\varnothing] > 0$ and

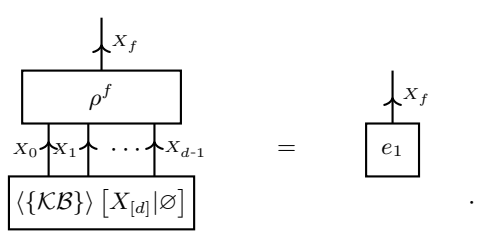
$$\mathbb{P}\left[X_f = 0|X_{\mathcal{KB}} = 1\right] = \frac{\langle\{\neg f, \mathcal{KB}\}\rangle[\varnothing]}{\langle\{\mathcal{KB}\}\rangle[\varnothing]}\,.$$

This term vanishes if and only if $\langle\{\neg f, \mathcal{KB}\}\rangle[\varnothing]$ vanish. Thus, the condition is equivalent to the condition in Theorem 24. □

Given that $\mathcal{KB}$ is satisfiable, we therefore have $\mathcal{KB} \models f$ if and only if

$$\mathbb{P}\left[X_f|X_{\mathcal{KB}} = 1\right] = e_1\,. \tag{29}$$

We depict this condition by the contraction diagram



We can omit the normation by $\langle\{\mathcal{KB}\}\rangle[\varnothing]$ when deciding entailment, as we state next.

Corollary 4. *Given a satisfiable formula $\mathcal{KB}$, we have $\mathcal{KB} \models f$ (respectively $\mathcal{KB} \models \neg f$), if and only if*

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

Relating entailment to probability distributions motivates an extension of Definition39 of entailment to arbitrary probability distributions.

Definition 40. *For any propositional formula f and a probability distribution $\mathbb{P}$ we say that $\mathbb{P}$ probabilistically entails f, denoted as $\mathbb{P} \models f$, if*

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

If $\mathbb{P} \models \neg f$ we say that $\mathbb{P}$ probabilistically contradicts f.

By Theorem 27 the definition of entailment reduces to propositional formulas by choosing $\mathbb{P} = \langle \{\mathcal{KB}\} \rangle \left[X_{[d]} | \varnothing \right]$

6.2.4 Deciding entailment on Markov Networks

Let us now relate the probabilistic entailment definition 40 with the logical entailment. Given a generic probability distribution $\mathbb{P}$ we can build a Knowledge Base by the indicator function of the support as

$$\mathcal{KB}^{\mathbb{P}} = \chi \circ \mathbb{P}$$

where $\chi : \mathbb{R} \to \mathbb{R}$ is defined as $\chi(x) = 1$ if $x \neq 0$ and $\chi(x) = 0$ else.

Theorem 28. *Any probability distribution $\mathbb{P}$ probabilistically entails a formula f, if and only if the Knowledge Base $\mathcal{KB}^{\mathbb{P}}$ logically entails f.*

Proof. Whenever $\mathbb{P}$ does not entail f probabilistically we find a state $x_{[d]} \in \bigtimes_{k \in [d]} [2]$ such that

$$\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0 \quad \text{and} \quad f\left[X_{[d]} = x_{[d]}\right] = 0\,.$$

We further have $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0$ if and only if $\mathcal{KB}^{\mathbb{P}}[X_{[d]} = x_{[d]}] = 1$ and

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

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

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

is not satisfied. Therefore, probabilistic entailment of f by $\mathbb{P}$ is equivalent to logical entailment of f by $\mathcal{KB}^{\mathbb{P}}$. □

Theorem 29. *Let $\mathcal{T}^{\mathcal{G}} = \{T^e \,:\, e \in \mathcal{E}\}$ be a non-negative Tensor Network on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\tilde{\mathcal{V}} \subset \mathcal{V}$ be a subset and*

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

and

$$\tilde{\mathbb{P}}[X_{\tilde{\mathcal{V}}}] = \langle \{\chi \circ T[e] \,:\, e \in \mathcal{E}\} \rangle [X_{\tilde{\mathcal{V}}} | \varnothing]$$

Then we have for any f that $\mathbb{P} \models f$ if and only if $\tilde{\mathbb{P}} \models f$.

Proof. We first show

$$\chi \circ \mathbb{P} = \chi \circ \tilde{\mathbb{P}}\,. \tag{30}$$

The claim follows then from Theorem 28. To show (30) let there be $X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}$ such that $\mathbb{P}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$. Then for any $X_{\mathcal{V}} = x_{\mathcal{V}}$ extending $X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}$ we have $\langle \{T[e] \,:\, e \in \mathcal{E}\} \rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0$ and thus also $\langle \{\chi \circ T[e] \,:\, e \in \mathcal{E}\} \rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0$ and $\tilde{\mathbb{P}}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$. One can similarly show, that when $\tilde{\mathbb{P}}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$ then also $\mathbb{P}[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = 0$. The support of the distributions $\mathbb{P}$ and $\tilde{\mathbb{P}}$ is thus identical and (30) holds. □

For any positive tensor T we have

$$\chi \circ T[X_e] = \mathbb{I}[X_e]\ ,$$

which does not influence the distribution and can be omitted from the Markov Network. By Theorem 29, when deciding eintailment, we can reduce all tensors of a Markov Network to their support and omit those with full support. Since the support indicating tensors $\chi \circ T[X_e]$ are Boolean, each is a propositional formula and the Markov Network is turned into a Knowledge Base of their conjunctions. Deciding probabilstic entailment is thus traced back to logical entailment.

6.3 Deciding Entailment by partial ordering

Classically entailment in propositional logics is defined by a model-theoretic approach. According to that approach, the entailment statement $\mathcal{KB} \models f$ holds, whenever any model of $\mathcal{KB}$ is also a model of f. We will in the following show, that this is equal to our definition based on probabilistic queries.

Definition 41 (Partial ordering of tensors)**.** *We say that two tensors f and h in a tensor space $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ are partially ordered, denoted by*

$$f \prec h\,,$$

if for all $x_0, \ldots, x_{d-1} \in \bigtimes_{k\in[d]}[m_k]$

$$f(x_0, \ldots, x_{d-1}) \leq h(x_0, \ldots, x_{d-1})\,.$$

We notice, that whenever $f \prec h$ holds, for any model $x_0, \ldots, x_{d-1}$ of f we have

$$1 = f(x_0, \ldots, x_{d-1}) \leq h(x_0, \ldots, x_{d-1})$$

and thus $h(x_0, \ldots, x_{d-1}) = 1$. Therefore any model of f is also a model of h. We show in the next theorem, that this is equivalent to the entailment statement $f \models h$.

Theorem 30 (Partial Ordering Criterion)**.** *We have $\mathcal{KB} \models f$ if and only if $\mathcal{KB} \prec f$.*

Proof. Directly by definition, since both $\mathcal{KB}$ and f are Boolean and therefore for any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have that

$$\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] \leq f\left[X_{[d]} = x_{[d]}\right]$$

is equivalent to

$$\left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right).$$

Therefore, $\mathcal{KB} \prec f$ is equivalent to

$$\forall_{x_{[d]}\in\bigtimes_{k\in[d]}[2]}\left(\mathcal{KB}\left[X_{[d]} = x_{[d]}\right] = 1\right) \to \left(f\left[X_{[d]} = x_{[d]}\right] = 1\right),$$

which is equal to $\mathcal{KB} \models f$. □

6.3.1 Monotonicity of Entailment

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

Theorem 31 (Monotonicity of Entailment)**.** *For any Markov Network on the decorated hypergraph $\mathcal{G}$ and any subgraph $\tilde{\mathcal{G}}$, we have for any formula that $\mathbb{P}^{\mathcal{G}} \models f$ if $\mathbb{P}^{\tilde{\mathcal{G}}} \models f$.*

Proof. Based on the reduction to Knowledge Bases by Theorem 28 and the monotonocity of binary contractions as shown in Theorem 78. □

Remark 14. *To make use of Theorem 31 we can exploit any entailment criterion. However, there is no claim about entailment being false, when the entailment Theorem 31 therefore just provides a sufficient but not necessary criterion of entailment with respect to $\mathbb{P}^{\mathcal{G}}$.*

6.4 Deciding Entailment by local contractions

Global entailment can become inefficient, when

- we are interested in batches of entailment checks. Here we can make use of dynamic programming (store partial contraction results in the Knowledge Cores).
- the network is large. Although efficient tensor network contraction often work, they might get infeasible when the tensor network has a large connectivity. For many

An alternative to deciding entailment by global operations is the use of local operations. Here we interpret a part of the network (for example a single core) as an own knowledge base (with atomic formulas being the roots of the directed subgraph, that is potentially differing with the atoms in the global perspective) and perform entailment with respect to that.

Remark 15. *Tradeoff between generality and efficiency While generic entailment decision algorithms (those by the full network) can decide any entailment, local algorithms as presented here can only perform some, but therefore more effectively as operating batchwise (dynamically deciding entailment for many leg variables). This is a typical phenomenon in logical reasoning and related to decidability.*

6.4.1 Knowledge Propagation

Let us now draw on these insights and store partial entailment results in Knowledge Cores, which is a use of the dynamic programming paradigm. We then iterate over local entailment checks, where we recursively add further entailment checks to be redone due to additional knowledge. We then call the local checks until convergence Entailment Propagation, since different stadia of knowledge are propagated through the network. We describe local Knowledge Propagation in a generic way in Algorithm 6.

Algorithm 6 Knowledge Propagation (KP)

Tensor Network $\mathcal{T}^{\mathcal{G}}$, $K^e = \mathbb{I}\left[X_e\right]$
while Stopping Criterion is not met **do**
  Choose e, subset M of $\mathcal{T}^{\mathcal{G}}$ and of $\{K^e : e \in \mathcal{E}\}$ containing K^e
  Update
$$K^e \leftarrow \chi \circ \langle M \rangle \left[X_e\right]$$
end while

Each chosen subset M is understood as a local knowledge base, which is then applied for local entailment.

There are different ways of implementing Algorithm 6, by choosing an order of local knowledge bases M and a stopping criterion.

Theorem 32. *In Entailment Propagation Algorithm 6, K^e is monotonically decreasing with respect to the partial ordering and greater than $\tilde{K^e}$ defined as*
$$\tilde{K^e} = \chi\left(\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [e]\right) .$$

Proof. We deduce the theorem from generic properties of the support of contractions, see Section 16.5. Monotonic decreasing follows from montonocity of tensor contractions, see Theorem 78. By Theorem 79 we have during any state of the algorithm
$$\chi \circ \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[X_{\mathcal{V}}\right] = \chi \circ \left\langle \mathcal{T}^{\mathcal{G}} \cup \{K^e : e \in \mathcal{E}\} \right\rangle \left[X_{\mathcal{V}}\right] .$$
If follows that
$$\tilde{K^e} = \chi \circ \left\langle \mathcal{T}^{\mathcal{G}} \cup \{K^e : e \in \mathcal{E}\} \right\rangle \left[X_e\right]$$
and by Theorem 78
$$\tilde{K^e} \prec K^e .$$
□

Corollary 5. *Whenever for a formula $f\left[X_{\tilde{\mathcal{V}}}\right]$ and a K^e we have*
$$\left\langle K^e, \rho^f \right\rangle \left[X_f = 0\right] = 0$$
then the Markov Network $\mathcal{T}^{\mathcal{G}}$ probabilistically entails f.

Another way to use Algorithm 6 to decide entailement of formulas f is adding each ρ^f to $\mathcal{T}^{\mathcal{G}}$ and defining Knowledge Cores $K^f[X_f]$. Since then the Knowledge Core has only two dimensions, there are only four possible cores with the interpretation

- e_1: the formula is known to be true
- e_0: the formula is known to be false
- $\mathbb{I}$: the formula is not known
- 0: the knowledge base is inconsistent

Part II
Neuro-Symbolic Learning

We now employ tensor networks to define architectures and algorithms for neuro-symbolic reasoning based on the logical and probabilistic foundations. Markov Logic Networks will be taken as generative models to be learned from data, using formula selecting tensor networks and likelihood optimization algorithms.

7 Formula Selecting Networks

In this chapter we will investigate efficient schemes to represent collections of formulas with similar structure in one tensor network.

Definition 42. *Given a set of p formulas $\{f_l : l \in [p]\}$, we define the formula selecting map as*

$$\mathcal{H}\left[X_{[d]}, L\right] : \bigtimes_{k\in[d]} [2] \times [p] \to [2]$$

defined for $l \in [p]$ by

$$\mathcal{H}\left[X_{[d]} = x_0, \ldots, x_{d-1}, L = l\right] = f_l\left[X_{[d]} = x_0, \ldots, x_{d-1}\right] .$$

We introduce a selection variable L and depict the formula selection in Figure 12.

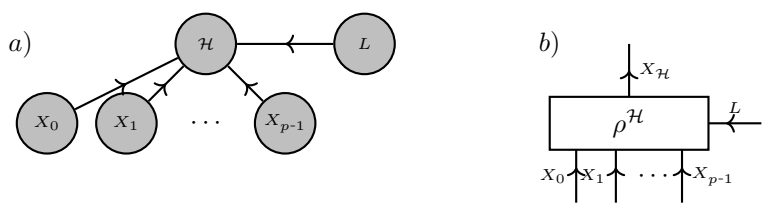


Figure 12: Representation of the Formula Selecting map as a a) Graphical Model with a selection variable $\mathcal{H}$. b) Dual Tensor Core with selection variable corresponding with an additional axis.

A naive representation of the formula selecting map is as a sum

$$\mathcal{H} = \sum_{l\in[p]} f_l\left[X_{[d]}\right] \otimes e_l\left[L\right] .$$

Such a representation scheme requires linear resources in the number of formulas. We will show in the following, that we can exploit common structure in formulas to drastically reduce this resource consumption.

7.1 Construction schemes

Let us now investigate efficient schemes to define sets of formulas to be used in the definition of $\mathcal{H}$. We will motivate the folding of the selection variable into multiple selection variables by compositions of selection maps.

7.1.1 Connective Selecting Tensors

We represent choices over connectives with a fixed number of arguments by adding a selection variable to the cores and defining each slice by a candidate connective.

Definition 43. *Let $\{\circ_0, \ldots, \circ_{p_C-1}\}$ be a set of connectives with d arguments. The associated connective selection map is*

$$\mathcal{H}_C\left[X_{[d]}, L_C\right] : \bigtimes_{k\in[d]} [2] \times [p_C] \to [2]$$

defined for each $l_C \in [p_C]$ and $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ by

$$\mathcal{H}_C\left[X_{[d]} = x_{[d]}, L_C = l_C\right] = \circ_{l_C}\left[X_{[d]} = x_{[d]}\right] .$$

We depict the relational encoding of connective selection maps in Figure 13.

Remark 16 (HT Interpretation of Superposed Formula Tensor Networks)**.** *Continuing Remark 9: Superposed Formula Tensors have a decomposition into a* HT *as sketched here, where we distinguish between formula selection subspaces (indices l_s) and atomic subspaces (indices x_k). At each formula selection we thus have a decomposition into three subspaces, two of atomic formulas and one for the formula selection.*

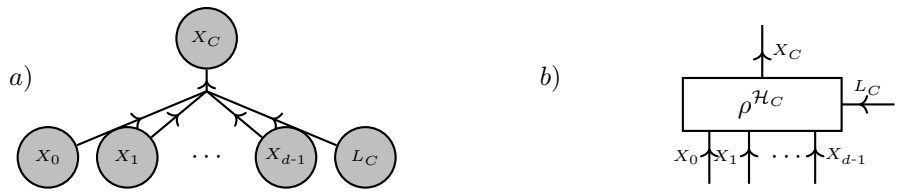


Figure 13: Connective Selector.

7.1.2 Variable Selecting Tensor Network

Definition 44. *The selection of one out of p variables in a list $X_{[p]}$ is done by variable selecting maps*

$$\mathcal{H}_V\left[X_{[p]}, L_V\right] : \left(\bigtimes_{l\in[p]} [2]\right) \times [p] \to [2] \tag{31}$$

are defined coordinatewise by

$$\mathcal{H}_V\left[X_0 = x_0, \ldots, X_{p-1} = x_{p-1}, L_V = l_V\right] = x_l\,. \tag{32}$$

Variable selecting maps appear in the literature as multiplex gates (see Definition 5.3 in Koller and Friedman (2009)). The relational encoding of the variable selection map has a decomposition

$$\rho^{\mathcal{H}_V}\left[X_{\mathcal{H}_V}, X_{[p_V]}\right] = \sum_{l_V\in[p_V]} \rho^{X_{l_V}}\left[X_{\mathcal{H}_V}, X_{l_V}\right] \otimes e_{l_V}\left[L_V\right]\,.$$

This structure is exploited in the next theorem to derive a tensor network decomposition of $\rho^{\mathcal{H}_V}$.

Theorem 33 (Decomposition of Variable Selecting Maps)**.** *Given a list $X_{[p_V]}$ of variables, we define for each $l_V \in [p_V]$ the tensors*

$$T^{V_{l_V}}\left[X_{l_V}, L_V\right] = \delta\left[X_{\mathcal{H}_V}, X_{l_V}\right] \otimes e_{l_V}\left[L_V\right] + \mathbb{I}\left[X_{\mathcal{H}_V}, X_{l_V}\right] \otimes \left(\mathbb{I}\left[L_V\right] - e_{l_V}\left[L_V\right]\right)\,.$$

Then we have (see Figure 14)

$$\rho^{\mathcal{H}_V}\left[X_{\mathcal{H}_V}, X_{[p]}, L_V\right] = \left\langle\left\{T^{V_{l_V}}\left[X_{\mathcal{H}_V}, X_{l_V}, L_V\right] : l_V \in [p_V]\right\}\right\rangle\left[X_{\mathcal{H}_V}, X_{[p]}, L_V\right]\,.$$

Proof. We show the equivalence of the tensors on an arbitrary coordinates. For $\tilde{l}_V \in [p_V]$, $X_{\mathcal{H}_V} \in [2]$ and $x_{[p_V]} \in \bigtimes_{k\in[p_V]}[2]$ we have

$$\begin{aligned}
&\left\langle\left\{T^{V_{l_V}}\left[X_{\mathcal{H}_V}, X_{l_V}, L_V\right] : l_V \in [p_V]\right\}\right\rangle\left[X_{\mathcal{H}_V} = x_{\mathcal{H}_V}, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V\right] \\
&\quad = \prod_{l_V\in[p_V]} T^{V_{l_V}}\left[X_{\mathcal{H}_V} = x_{\mathcal{H}_V}, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V\right] \\
&\quad = T^{V_{\tilde{l}_V}}\left[X_{\mathcal{H}_V} = x_{\mathcal{H}_V}, X_{l_V} = x_{l_V}, L_V = \tilde{l}_V\right] \\
&\quad = \begin{cases} 1 & \text{if} \quad x_{\mathcal{H}_V} = x_{l_V} \\ 0 & \text{else} \end{cases} \\
&= \rho^{\mathcal{H}_V}\left[X_{\mathcal{H}_V} = x_{\mathcal{H}_V}, X_{[p]} = x_{[p]}, L_V = \tilde{l}_V\right]
\end{aligned}$$

In the second equality, we used that the tensor $T^{V_{l_V}}$ have coordinates 1 whenever $\tilde{l}_V \neq l_V$. □

The decomposition provided by Theorem 33 is in a CP format, as will be further discussed in Chapter 19. The introduced tensors $T^{V_{l_V}}$ are Boolean, but not directed and therefore encodings of relations but not functions (see Chapter 17).

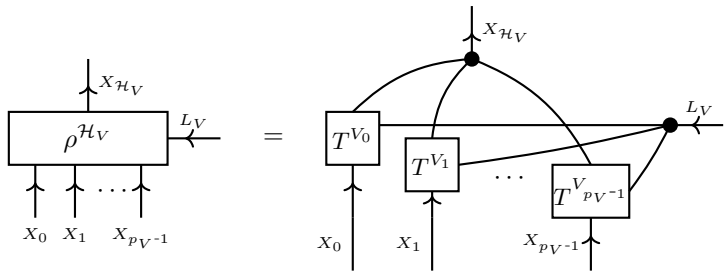


Figure 14: Decomposition of the relational encoding of a variable selecting tensor into a network of tensors defined in Theorem 33. The decomposition is in a CP-Format (see Chapter 19.

7.2 State Selecting Tensors

As an alternative, one can select a state of a categorical variable X.

Definition 45. *Given a categorical variable X_S with dimension m_S and a selection variable L_S with dimension $p_S = m_S$ the state selecting tensor*

$$\mathcal{H}_S\left[X_S, L_S\right] : [m_S] \times [p_S] \to [2]$$

is defined on $x_S \in [m_S]$ and $l_S \in [p_S]$ by

$$\mathcal{H}_S\left[X = x, L_S = l_S\right] = \begin{cases} 1 & \textit{if} \quad x = l_S \\ 0 & \textit{else} \end{cases} .$$

State selecting tensors can also be realized by variable selecting tensors. In Section 9.4 we will describe methods to build atomic variables indicating the states of a categorical variable. This would, however, increase the number of variables in a tensor network and can thus lead to an exponential overhead of dimensions. State selecting tensors can therefore be seen as a mean to avoid such dimension increases.

Comment: State Selectors can be integrated in Variable Selection framework. In this perspective, Variable selection networks are the specific case to $X = 1$.

7.3 Composition of formula selecting maps

We will now parametrize the sets $\mathcal{F}$ with additional indices and define formula selector maps subsuming all formulas. To handle large sets of formulas, we further fold the selection variable into tuples of selection variables.

Definition 46. *Let there be a formula $f_{l_0,\ldots,l_{n-1}}$ for each index tuple in $l_0, \ldots, l_{n-1} \in \bigtimes_{s\in[n]} p_s$, where $n, p_0, \ldots, p_{n-1} \in \mathbb{N}$. The folded formula selector map (see Figure 15) is the map*

$$\mathcal{H}\left[X_{[d]}, L_{[n]}\right] : \left(\bigtimes_{k\in[d]} [2]\right) \times \left(\bigtimes_{s\in[n]} p_s\right) \to [2]$$

with the coordinates at the indices $x_{[d]} \in \bigtimes_{k\in[d]} [2]$, $l_{[n]} \in \bigtimes_{s\in[n]} p_s$

$$\mathcal{H}\left[X_{[d]} = x_{[d]}, L_{[n]} = l_{[n]}\right] = f_{l_{[n]}}\left[X_{[d]} = x_{[d]}\right] .$$

We will find formula selector maps by composition variables selector maps (Definition 44) and connective selector maps (Definition 43). This is especially useful to provide efficient decompositions of relational encodings.

7.3.1 Formula Selecting Neuron

The folding of the selection variable is motivated by the composition of selection maps. We call the composition of a connective selection with variable selection maps for each argument a formula selecting neuron.

Definition 47. *Given an order $n \in \mathbb{N}$ let there be a connective selector $L_\circ$ selecting connectives of order n and let $\mathcal{H}_{V,0}, \ldots, \mathcal{H}_{V,n-1}$ be a collection of variable selectors. The corresponding logical neuron is the map*

$$\sigma\left[X_{[d]}, L_{[n]}\right] : \left(\bigtimes_{k\in[d]} [2]\right) \times [p_C] \times \left(\bigtimes_{s\in[n]} [p_s]\right) \to [2]$$

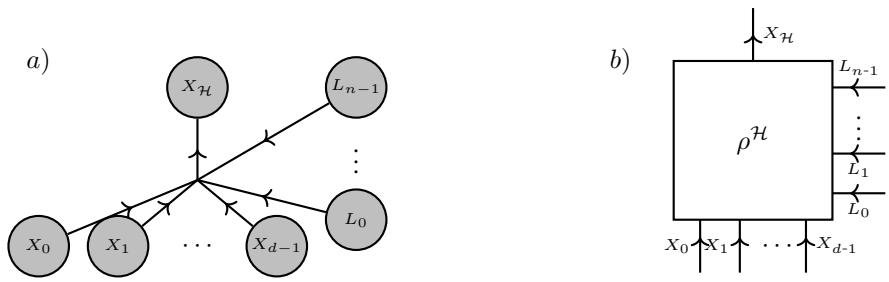


Figure 15: Relational encoding of the folded map $\mathcal{H}$.

defined for $x_{[d]} \in \bigtimes_{k\in[d]}[2]$, $l_C \in [p_C]$ *and* $l_0 \dots l_{n-1} \in \bigtimes_{s\in[n]}[p_s]$ *by*

$$\sigma(x_0, \dots, x_{d-1}, l_C, l_0 \dots l_{n-1}) = \mathcal{H}_C(\mathcal{H}_{V,0}(x_0, \dots, x_{d-1}, l_0), \dots, \mathcal{H}_{V,n-1}(x_0, \dots, x_{d-1}, l_{n-1}), l_C)\,.$$

Each neuron has a tensor network decomposition by a connective selector tensor and a variable selector tensor network for each argument, as we state in the next theorem.

Theorem 34. *Decomposition of formula selecting neurons Let σ a logical neuron, defined for a connective selector $L_\circ$ and variable selectors $\mathcal{H}_{V,0}, \dots, \mathcal{H}_{V,n-1}$. Then we have (see Figure 16 for the example of $n = 2$):*

$$\begin{aligned}
&\rho^{\sigma}\left[X_\sigma, X_{[d]}, L_C, L_{V,0}, \dots, L_{V,n-1}\right] \\
&\quad = \langle\{\rho^{\mathcal{H}_C}\left[X_\sigma, X_{V,0}, \dots, X_{V,n-1}\right], \\
&\qquad \rho^{\mathcal{H}_{V,0}}\left[X_{V,0}, X_{[d]}, L_{V,0}\right], \dots, \rho^{\mathcal{H}_{V,n-1}}\left[X_{V,n-1}, X_{[d]}, L_{V,n-1}\right]\}\rangle\left[X_\sigma, X_{[d]}, L_C, L_{V,0}, \dots, L_{V,n-1}\right]\,.
\end{aligned}$$

Proof. By composition Theorem 83. □

Example of a formula selecting neuron: Given a skeleton expression and a set of candidates at each placeholder, we parameterize a set of formulas by the assignment of candidate atoms to each placeholder position. Let us denote the set of formulas, which are generated through choosing atoms from $\mathcal{M}^s$ for the skeleton formula S by

$$\mathcal{F}_S := \left\{S\left(Z^1, \dots, Z^d\right) : Z^k \in \mathcal{M}^k\right\}$$

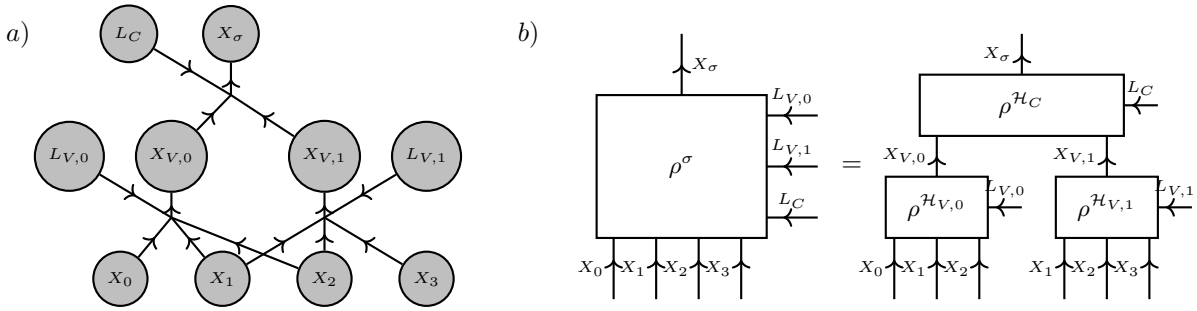


Figure 16: Example of a logical neuron σ of order $n = 2$. a) Selection and categorical variables and their interdependencies visualized in a hypergraph. b) Relational encoding of the logical neuron and tensor network decomposition into variable selecting and connective selecting tensors.

7.3.2 Formula Selecting Neural Network

Single neurons have a limited expressivity, since for each choice of the selection variables they can just express single connectives acting on atomic variables. The expressivity is extended to all propositional formulas, when allowing for networks of neurons, which can select each others as input arguments.

Definition 48. *An architecture graph $\mathcal{G}^{\mathcal{A}} = (\mathcal{V}^{\mathcal{A}}, \mathcal{E}^{\mathcal{A}})$ is an acyclic directed hypergraph with nodes appearing at most once as outgoing nodes. Nodes appearing only as outgoing nodes are input neurons and are labeled by $\mathcal{A}^{\text{in}}$ and nodes not appearing as outgoing nodes are the output neurons in the set $\mathcal{A}^{\text{out}}$ (see Figure 17 for an example).*

Given an architecture graph $\mathcal{G}^{\mathcal{A}} = (\mathcal{V}^{\mathcal{A}}, \mathcal{E}^{\mathcal{A}})$, a formula selecting neural network *$\mathcal{H}_{\mathcal{A}}$ is a tensor network of logical neurons at each $\sigma \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}$, such that each neuron depends on variables $X_{\text{Pa}(\sigma)}$ and on selection variables L_σ. The collection of all selection variable is notated by $L_{\mathcal{A}}$.*

The activation tensor of each neuron $\sigma \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}$ is

$$\sigma^{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \{\rho^{\tilde{\sigma}} \,:\, \tilde{\sigma} \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}\} \cup \{e_1\left[X_\sigma\right]\}\right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] \,.$$

The activation tensor of the formula selecting neural network is the contraction

$$\mathcal{H}_{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \{\rho^{\sigma^{\mathcal{A}}}\left[X_\sigma, X_{\text{Pa}(\sigma)}, L_{\mathcal{A}}\right] \,:\, \sigma \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}\} \cup \{e_1\left[X_\sigma\right] \,:\, \sigma \in \mathcal{A}^{\text{out}}\}\right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] \,.$$

The expressivity of a formula selecting neural network $\mathcal{H}_{\mathcal{A}}$ is the formula set

$$\mathcal{F}_{\mathcal{A}} = \left\{ \mathcal{A}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}} = l_{\mathcal{A}}\right] : l_{\mathcal{A}} \in \bigtimes_{s \in [n]} p_s \right\} \,.$$

The activation tensor of each neuron depends in general on the activation tensor of its ancestor neurons with respect to the directed graph $\mathcal{G}^{\mathcal{A}}$, and thus inherits the selection variables.

We notice that the architecture graph is a scheme to construct the variable dependency graph of the tensor network $\mathcal{F}_{\mathcal{A}}$. To this end, we replace each neuron $\sigma \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}$ by an output variable X_σ and further add selection variables L_σ to the directed edges, that is to each directed hyperedge $(\{\sigma\}, \text{Pa}(\sigma)) \in \mathcal{E}^{\mathcal{A}}$ we construct a directed hyperedge $(\{X_\sigma\}, X_{\text{Pa}(\sigma)} \cup L_\sigma)$.

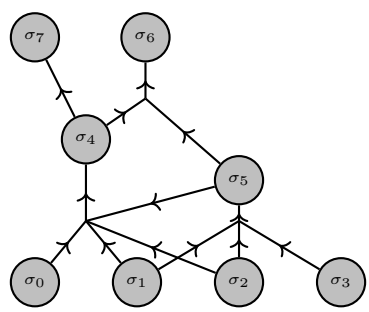


Figure 17: Example of an architecture graph $\mathcal{G}^{\mathcal{A}}$ with input neurons $\mathcal{A}^{\text{in}} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$ and output neurons $\mathcal{A}^{\text{out}} = \{\sigma_6, \sigma_7\}$

Theorem 35. *Given fixed selection variables $L_{\mathcal{A}}$, the formula selecting neural network is the conjunction of output neurons, that is*

$$\mathcal{H}_{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \bigwedge_{\sigma \in \mathcal{A}^{\text{out}}} \sigma\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] \,.$$

Proof. By effective calculus (see Theorem 86), we have

$$\left\langle \{\rho^{\wedge}\left[X_{\wedge}, X_{[d]}\right], e_1\left[X_{\wedge}\right]\}\right\rangle \left[X_{[d]}\right] = \bigotimes_{k \in [d]} e_1\left[X_k\right]$$

and thus

$$\mathcal{H}_{\mathcal{A}}\left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] = \left\langle \{\rho^{\sigma} \,:\, \sigma \in \mathcal{V}^{\mathcal{A}}/\mathcal{A}^{\text{in}}\} \cup \{\rho^{\wedge}\left[X_{\wedge}, X_\sigma \,:\, \sigma \in \mathcal{A}^{\text{out}}\right], e_1\left[X_{\wedge}\right]\}\right\rangle \left[X_{\mathcal{A}^{\text{in}}}, L_{\mathcal{A}}\right] \,.$$

□

By the commutation of contractions, we can further use Theorem 34 to decompose each tensor ρ^σ into connective and variable selecting components to get a sparse representation of a formula selecting neural network $\mathcal{H}_{\mathcal{A}}$.

7.4 Application of Formula Selecting Networks

There are two main applications of formula selecting networks. First, when contracting the selection variables with a weight tensor we get a weighted sum of the parametrized formulas. Second, when contracting the categorical variables with a distribution or a knowledge base, we get a tensor storing the satisfaction rates respectively the world counts of the parametrized formulas.

7.4.1 Representation of selection encodings

In technical perspective: FSN provide efficient representation of $\gamma^{\mathcal{F}}$ -> Use for exponential families, structure learning.

Lemma 10. *Given a set* $\{f_{l_0,\dots,l_{n-1}} : l_0,\dots,l_{n-1} \in \bigtimes_{s\in[n]} p_s\}$ *of propositional formulas we define the statistic*

$$\mathcal{F} : x_0,\dots,x_{d-1} \to (f_{l_0,\dots,l_{n-1}}(x_0,\dots,x_{d-1}))_{l_0,\dots,l_{n-1}} .$$

and the formula selecting map

$$\mathcal{H} : x_0,\dots,x_{d-1},l_0,\dots,l_{n-1} \to f_{l_0,\dots,l_{n-1}}(x_0,\dots,x_{d-1}) .$$

Then

$$\gamma^{\mathcal{F}}\left[X_{[d]},L_{[n]}\right] = \mathcal{H}\left[X_{[d]},L_{[n]}\right] .$$

Proof. For any indices $l_{[n]} \in \bigtimes_{s\in[n]} p_s$ and $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have

$$\gamma^{\mathcal{F}}\left[X_{[d]}=x_{[d]},L_{[n]}=l_{[n]}\right] = f_{l_0,\dots,l_{n-1}}(x_0,\dots,x_{d-1}) = \mathcal{H}\left[X_{[d]}=x_{[d]},L_{[n]}=l_{[n]}\right] .$$

□

Technically, relational encodings have been exploited to derive decompositions based on basis calculus. Selection encodings on the other hand enable the application of formula selecting networks as superpositions of formulas.

7.4.2 Efficient Representation of Formulas

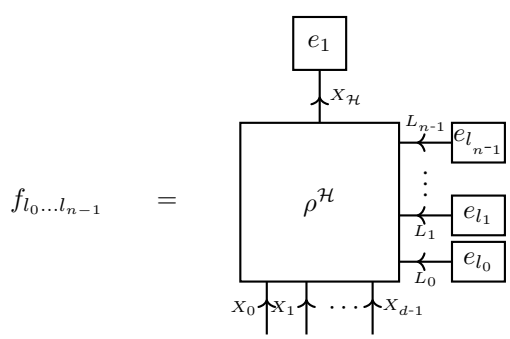
Weight contracted at the selection variables is elementary, then single formula retrieved.

Formula Selecting Neural Networks are means to represent exponentially many formulas with linear (in sum of candidates list lengths) storage. Their contraction with probability tensor networks, is thus a batchwise evaluation of exponentially many formulas. This is possible due to redundancies in logical calculus due to modular combinations of subformulas.

We can retrieve specific formulas by slicing the selection variables, i.e. for $l_0 \dots l_{n-1}$ we have

$$f_{l_0\dots l_{n-1}}[X_{[d]}] = \mathcal{H}\left[X_{[d]}, L = l_0\dots l_{n-1}\right] .$$

In a tensor network diagram we depict this by



Another perspective on the efficient formula evaluation by selection tensor networks is dynamic computing. Evaluating a formula requires evaluations of its subformulas, which are done by subcontractions and saved for different subformulas due to the additional selection legs.

However, we need to avoid contracting the tensor with leaving all selection legs open, since this would require exponential storage demand.

We can avoid this storage bottleneck by extending the contractions by additional cores leaving less variable legs open. This is the case when contracting gradients of the parameter tensor networks in alternating least squares approaches. Other methods avoiding the bottleneck can be constructed by MCMC sampling, for example Gibbs Sampling. Here we only need to vary local components of the formula reflected in keeping only single variable legs open.

7.4.3 Batch contraction of parametrized formulas

Given a set $\mathcal{F}$ of formulas, we build a formula selecting network parametrizing the formulas. The contraction

$$\left\langle \mathcal{T}^{\mathcal{G}}, \mathcal{H} \right\rangle \left[L_{[n]} \right]$$

is a tensor containing the contractions of the formulas $f_{l_{[n]}}$ with an arbitrary tensor network $\mathcal{T}^{\mathcal{G}}$ as

$$\left\langle \left\{ \mathcal{T}^{\mathcal{G}}, f_{l_{[n]}} \right\} \right\rangle [\varnothing] = \left\langle \left\{ \mathcal{T}^{\mathcal{G}}, \mathcal{H} \right\} \right\rangle \left[L_{[n]} = l_{[n]} \right] .$$

7.4.4 Average contraction of parametrized formulas

We show in the next two examples, how a full contraction of the formula selecting map with a probability distribution or a knowledge base can be interpreted.

Example 8 (Average satisfaction of formulas)**.** *The average of the formula satisfactions in $\mathcal{F}$ giben a probability tensor $\mathbb{P}$ is*

$$\frac{1}{\prod_{s \in [n]} p_s} \cdot \left\langle \left\{ \mathbb{P}, \gamma^{\mathcal{F}} \right\} \right\rangle [\varnothing] .$$

Example 9 (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*

$$\left\langle \left\{ \mathcal{KB}, \gamma^{\mathcal{F}} \right\} \right\rangle [\varnothing] = 0 .$$

We use Lemma 10 to get that $\gamma^{\mathcal{F}} = \mathcal{H}$. When the formulas are representable in a folded scheme, we find tensor network decompositions of $\mathcal{H}$ and exploit them along efficient representations of $\mathcal{KB}$ in an efficient calculation of $\left\langle \left\{ \mathcal{KB}, \gamma^{\mathcal{F}} \right\} \right\rangle [\varnothing]$. This is further equal to

$$\mathcal{KB} \models \neg \left(\bigvee_{f \in \mathcal{F}} f \right) .$$

7.5 Examples of formula selecting neural networks

7.5.1 Correlation

For example (see Figure 18) consider the logical neuron with single activation candidate $\{\wedge\}$ and two variable selectors selecting d atomic variables $X_{[d]}$. The expressivity of this network is the set of all conjunctions of the atoms

$$\{X_k \wedge X_l \,:\, k, l \in [d]\}$$

Contracting with a probability distribution, we use the tensor

$$T \left[L_{V,0}, L_{V,1} \right] = \left\langle \{\mathcal{H}_{\mathcal{A}}\} \right\rangle \left[L_{V,0}, L_{V,1} \right]$$

to read of covariances as

$$\mathrm{Cov}(X_k, X_l) = T \left[L_{V,0} = k, L_{V,1} = l \right] - T \left[L_{V,0} = k, L_{V,1} = k \right] \cdot T \left[L_{V,0} = l, L_{V,1} = l \right] .$$

7.5.2 Conjunctive and Disjunctive Normal Forms

We can represent any propositional knowledge base by the following scheme: Literal selecting neurons by connective identity/negation (selecting positive/negative literal) and selecting one of the atoms. Single output neuron representing the disjunction combining the literal selecting neurons. Number of neurons defined by the maximal clause size plus one. Smaller clauses can be covered when adding False as a possible choice (The respective neuron has to choose the identity, otherwise the full clause will be trivial).

The parameter core is in the basis CP format and each slice selects a clause of the knowledge base. When taking the slice values to infinity (e.g. by an annealing procedure), the represented member of the exponential family converges to the uniform distribution of the models of the knowledge base.

Useful to derive basis+ CP format based on CNF!

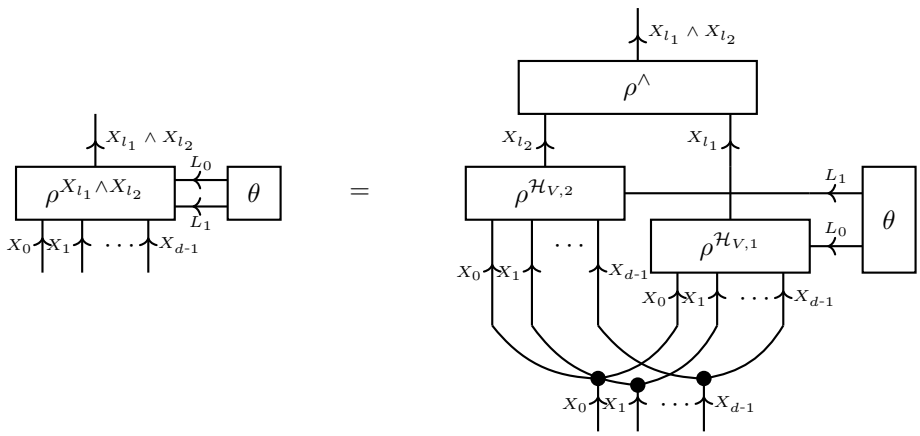


Figure 18: Superposition of the encoded formulas $\rho^{X_{l_1} \wedge X_{l_2}}$ with weight $\theta_{l_1 l_2}$

Remark 17 (Minterms and Maxterms)**.** *All minterms and maxterms can be represented by a two layer selection tensor networks without variable selection in two layers. The bottom layer has an* $\neg/\mathrm{Id}$ *connective selection neuron to each atom and the upper layer consists of a single dary conjunction.*

7.6 Extension to variables of larger dimension

Connective selecting tensors: Can encode arbitrary functions h_l of discrete variables, but need $X_{\mathcal{H}_C}$ to be an enumeration of the, in particular to be of dimension

$$m_{\mathcal{H}_C} = \left|\cup_{l \in [p]} \mathrm{im}\,(h_l)\right| \,.$$

Variable selecting tensors can be understood as specific cases of connective selecting tensors and can thus also be generalized in a straight forward manner by

$$m_{\mathcal{H}_C} = \left|\cup_{l \in [p]} \mathrm{im}\,(h_l)\right| \,.$$

State selecting tensors are directly

Example 10 (Discretization of a continuous neuron)**.** *Let there be a neuron*

$$\sigma(w, y) : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}\,.$$

When $w \in \mathcal{M}^{weight} \subset \mathbb{R}^d$ *and* $x \in \mathcal{M}^x \subset \mathbb{R}^d$ *have*

$$|\sigma(\langle w, x\rangle\,[])| \leq \left|\mathcal{M}^{weight}\right| \cdot |\mathcal{M}^x| \,.$$

To represent the discretization of the neuron, we use the subset encoding scheme of Definition 67. The variables O_{weight} *indexing* $\mathcal{M}^{weight}$ *will be understood as selection incoming variables and the variables* O_{weight} *indexing* $\mathcal{M}^{weight}$ *as categorical incoming variables. We further define a variable* O_σ *indexing* $\mathrm{im}\,(\sigma|_{\mathcal{M}^{weight} \times \mathcal{M}^x})$ *and have a tensor*

$$\rho^\sigma\,[O_\sigma, O_{\mathcal{M}^x}, O_{weight}] \,.$$

If the neuron is of the form

$$\sigma(w, x) = \psi(\sum_i w_i \cdot x_i)$$

a decomposition into multiplication at each coordinate and summation of the results, with relational encodings for each, can be done.

8 Markov Logic Networks

Markov Logic Networks exploit the efficiency and interpretability of logical calculus as well as the expressivity of graphical models.

8.1 Markov Logic Networks as Exponential Families

We introduce Markov Logic Networks in the formalism of exponential families (see Section 3.7).

Definition 49 (Markov Logic Networks)**.** *Markov Logic Networks are exponential families $\Gamma^{\mathcal{F}}$ with sufficient statistics by functions*

$$\mathcal{F} : \bigtimes_{k\in[d]} [2] \to \bigtimes_{f\in\mathcal{F}} [2] \subset \mathbb{R}^{|\mathcal{F}|}$$

defined coordinatewise by propositional formulas $f \in \mathcal{F}$.

Since the image of each feature is contained in $[2]$, they are propositional formulas (see Def. 35).

Conversely, any binary feature ϕ_l of an exponential family defines a propositional formula (see Definition 35). Thus, any exponential family of distributions of $\bigtimes_{k\in[d]}[2]$, such that $\operatorname{im}(\phi_l) \subset \{0,1\}$ for all $l \in [p]$ is a set of Markov Logic Networks with fixed formulas.

The sufficient statistics consistent in a map $\mathcal{F}$ of formulas brings the following advantages:

- Numerical Advantage: The sufficient statistics is decomposable into logical connectives. If the formulas are sparse (in the sense of limited number of connectives necessary in their representation), this gives rise to efficient tensor network decompositions of the relational encoding.
- Statistical Advantage: Since each formula is Boolean valued, the coordinates of the sufficient statistic are Bernoulli variables. Due to their boundedness, they and their averages (by Hoeffdings inequality) are sub-Gaussian variables with favorable concentration properties (absence of heavy tails).

Remark 18 (Alternative Definitions)**.** *We here defined MLNs on propositional logic, while originally they are defined in FOL. The relation of both frameworks will be discussed further in Chapter 13. Besides that we allow for degeneracy and improper MLNs.*

8.2 Tensor Network Representation of MLNs

Based on the previous discussion on the representation of exponential families by tensor networks in Section 3.7 we now derive a representation for Markov Logic Networks.

Theorem 36 (Relational Encodings for Probability Representation)**.** *A Markov Logic Network to a set of formulas $\mathcal{F} = \{f_l : l \in [p]\}$ is represented as*

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

where we denote for each $l \in [p]$ a head core

$$W^{f_l,\theta[L=l]}[X_{f_l}] = [1 \quad \exp\left[\theta\left[L = l\right]\right]]\left[X_{f_l}\right].$$

Proof. The claim follows from Theorem 13 and the following contraction equations. We have with the grouped variable $X_{\mathcal{F}} = \{X_{f_l} : l \in [p]\}$

$$\rho^{\mathcal{F}}\left[X_{[d]}, X_{\mathcal{F}}\right] = \left\langle \{\rho^{f_l}\left[X_{f_l}, X_{[d]}\right] : l \in [p]\} \right\rangle \left[X_{[d]}, X_{\mathcal{F}}\right] .$$

Since we have a Markov Logic Network we have $\operatorname{im}(f_l) \subset [2]$ and thus

$$W^{f_l,\theta[L=l]}[X_{f_l} = x_{f_l}] = \begin{cases} 1 & \text{for} \quad X_{f_l} = 0 \\ \exp\left[\theta\left[L = l\right]\right] & \text{for} \quad X_{f_l} = 1 \end{cases}$$

Using these equations, the claim follows from Theorem 13. □

Since any member of an exponential family is a Markov Network with tensors to each coordinate of the statistic, also Markov Logic Networks are Markov Networks.

Corollary 6. *Given a set $\mathcal{F}$ of formulas on atomic variables $X_{\mathcal{V}}$, we construct a $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ are decorated by the atoms and*

$$\mathcal{E} = \{\mathcal{V}^f : f \in \mathcal{F}\},$$

where by $\mathcal{V}^f$ we denote the minimal set such that there exists a tensor $T\left[X_{\mathcal{V}^f}\right]$ with

$$f\left[X_{\mathcal{V}}\right] = T\left[X_{\mathcal{V}^f}\right] \otimes \mathbb{I}\left[X_{\mathcal{V}/\mathcal{V}^f}\right] .$$

Any Markov Logic Network $(\mathcal{F}, \theta)$ is then a Markov Network given the graph $\mathcal{G}^{\mathcal{F}}$ $\{\exp\left[\theta\left[L = l\right] \cdot f_l\right] : l \in [p]\}$.

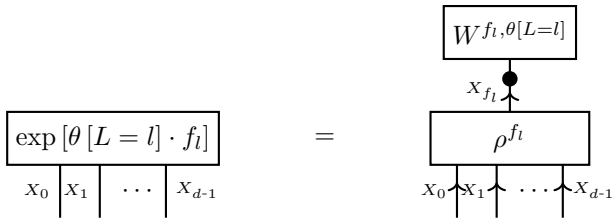


Figure 19: Factor of a Markov Logic Network to a formula f_l.

Markov Logic Networks are Markov Networks with the factors given in a restricted form from the weighted truth of a formula. Each formula is seen as a factor of the graphical model.

There are two sparsity mechanisms drastically reducing the number of parameters (and loosing generality):

- Factors/Formulas contain only subsets of atoms (already in Corollary 6 exploited): The underlying assumptions of conditional independence loss generality.
- Structure in the factors: In MLN each factor corresponds with a formula evaluated on possible worlds. Again, any possible factor can be represented by a formula, but we concentrate on small formulas (see Theorem **??**).

We can extend the set of variables, by including the hidden formulas, and get a Markov Network of the relational encodings of connectives and headcores. Here hidden variables are additional variables facilitating the decomposition, but not appearing in open variables of contractions when doing reasoning. One can then exploit redundancies and make sure that every subresult is computed just once, by dropping relational encodings with identical head functions.

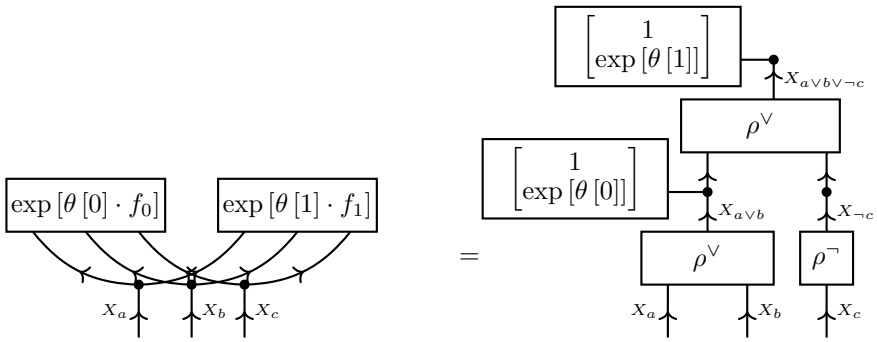


Figure 20: Example of a decomposed Markov Network representation of a Markov Logic Network with formulas $\{f_0 = a \vee b, f_1 = a \vee b \vee \neg c\}$. Since both formulas share the subformula $a \vee b$, their contracted factors have a representation by a connected tensor network.

8.2.1 Energy tensors of Markov Logic Networks

With the energy tensor

$$E^{(\mathcal{F},\theta)} = \sum_{l \in [p]} \theta\left[L = l\right] \cdot f_l\left[X_{[d]}\right] = \left\langle \left\{ \gamma^{\mathcal{F}}\left[X_{[d]}, L\right], \theta\left[L\right] \right\} \right\rangle \left[X_{[d]}\right] \tag{33}$$

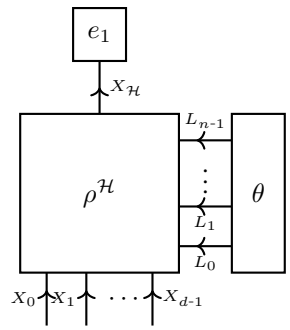
the MLN is the distribution

$$\mathbb{P}^{(\mathcal{F},\theta)}[X_{[d]}] = \left\langle \exp\left[E^{(\mathcal{F},\theta)}\right] \right\rangle \left[X_{[d]} | \varnothing\right] . \tag{34}$$

In case of a common structure of the formulas in a Markov Logic Network, Formula selecting networks can be applied to represent their energies.

We represent the superposition of formulas as a contraction with s parameter tensor. Given a factored parametrization of formulas $f_{l_0 \ldots l_{n-1}}$ with indices l_s we have the superposition by the network representation:

$$\sum_{l_{[n]} \in \times_{s \in [n]} [p_s]} \theta \left[L_{[n]} = l_{[n]} \right] f_{l_{[n]}} \quad =$$



If the number of atoms and parameters gets large, it is important to represent the tensor $f_{l_0 \dots l_{n-1}}$ efficiently in tensor network format and avoid contractions. To avoid inefficiency issues, we also have to represent the parameter tensor θ in a tensor network format to improve the variance of estimations (see Chapter 12) and provide efficient numerical algorithms.

However, when required to instantiate the probability distribution of a Markov Logic Network as a tensor network, we need to exponentiate and normate the energy tensor, a task for which relational encodings are required. For such tasks, contractions of formula selecting networks are not sufficient and each formula with a nonvanishing weight needs to be instantiated as a factor tensor of a Markov Network.

8.3 Expressivity

Based on Markov Logic Networks containing only maxterms and minterms (see Definition 38), we here provide an expressivity study. There are 2^d maxterms and 2^d minterms which are enough to represent any probability distribution as we show next.

Theorem 37. *Let there be a positive probability tensor*

$$\mathbb{P}\left[X_{[d]}\right] \in \bigotimes_{k \in [d]} \mathbb{R}^2 .$$

Then the Markov Logic Network of minterms (see Definition 38)

$$\mathcal{F}_\wedge = \{ Z^\wedge_{x_0,\dots,x_{d-1}} \,:\, x_0, \dots, x_{d-1} \in \bigtimes_{k \in [d]} [2] \}$$

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 wparameters

$$\theta \left[L_0 = x_0, \dots, L_{d-1} = x_{d-1} \right] = - \ln \mathbb{P} \left[X_0 = x_0, \dots, X_{d-1} = x_{d-1} \right]$$

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

$$E^{(\mathcal{F},\theta)} = \ln \mathbb{P}\left[X_{[d]}\right]$$

and therefore

$$\mathbb{P}^{(\mathcal{F},\theta)}[X_{[d]}] = \left\langle \left\{ \exp\left[E^{(\mathcal{F},\theta)} \right] \right\} \right\rangle \left[X_{[d]} | \varnothing \right] = \left\langle \left\{ \exp\left[E^{(\mathcal{F},\theta)} \right] \right\} \right\rangle \left[X_{[d]} \right] = \mathbb{P}\left[X_{[d]}\right] .$$

In the case of minterms, we notice that for any $x_0, \dots, x_{d-1} \in \bigtimes_{k \in [d]} [2]$

$$Z^\wedge_{x_0,\dots,x_{d-1}}[X_{[d]}] = e_{x_0,\dots,x_{d-1}} \left[X_{[d]} \right]$$

and thus with the weights in the claim

$$\sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \left(\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]\right) \cdot Z^{\wedge}_{x_0,\ldots,x_{d-1}}[X_{[d]}] = \ln \mathbb{P}\left[X_{[d]}\right] .$$

For the maxterms we have analogously

$$Z^{\vee}_{x_0,\ldots,x_{d-1}}[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]$$

and thus that the maximal clauses coincide with the one-hot encodings of respective states. We thus have

$$\begin{aligned}
&\sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \left(-\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]\right) \cdot Z^{\vee}_{x_0,\ldots,x_{d-1}}[X_{[d]}] \\
&= \left(\sum_{\mathcal{V}_0\subset[d]} \left(-\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]\right) \cdot \mathbb{I}\left[X_{[d]}\right]\right) \\
&\quad + \left(\sum_{\mathcal{V}_0\subset[d]} \left(\ln \mathbb{P}\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right]\right) \cdot e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right]\right) \\
&= \ln \mathbb{P}\left[X_{[d]}\right] + \lambda \cdot \mathbb{I}\left[X_{[d]}\right] ,
\end{aligned}$$

where λ is a constant. □

In general, this representation is redundant, since any offset of the weight by $\lambda \cdot \mathbb{I}$ results in the same distribution. However, the only $\bar{\theta}$ are multiples of $\mathbb{I}\left[X_{[d]}\right]$.

Theorem 37 is the analogue in Markov Logic to Theorem 23, which shows that any binary tensor has a representation by a logical formula, to probability tensors. Here we require positive distributions for well-defined energy tensors.

Remark 19 (Representation of Markov Networks)**.** *If a probability distribution is representable as a Markov Network, we only need to activate clauses and terms, which variables are contained in factors. Make a theorem out of that?*

8.4 Examples

8.4.1 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 38. *Let $\mathbb{P}\left[X_{[d]}\right]$ be a positive probability distribution, such that disjoint subsets of atoms are independent from each other. Then $\mathbb{P}\left[X_{[d]}\right]$ is the Markov Logic Network of atomic formulas*

$$\mathcal{F}_{[d]} = \{X_k \,:\, k \in [d]\}$$

and parameters

$$\theta\left[L = k\right] = \ln\left[\frac{\langle\mathbb{P}\rangle\left[X_k = 1\right]}{\langle\mathbb{P}\rangle\left[X_k = 0\right]}\right]$$

Proof. By Theorem 6 we get a decomposition

$$\mathbb{P}\left[X_{[d]}\right] = \bigotimes_{k\in[d]} \mathbb{P}^k\left[X_k\right]$$

where

$$\mathbb{P}^k\left[X_k\right] = \langle\{\mathbb{P}\}\rangle\left[X_k\right] .$$

By assumption of positivity, the vector $\mathbb{P}^k\left[X_k\right]$ is positive for each $k \in [d]$ and the parameter

$$\theta^k = \ln\left[\frac{\mathbb{P}^k\left[X_k = 1\right]}{\mathbb{P}^k\left[X_k = 0\right]}\right]$$

well-defined.

We then notice, that

$$\mathbb{P}^{(\{X_k\},\theta^k)}[X_k] = \mathbb{P}^k\left[X_k\right]$$

and therefore with the parameter vector of dimension $p = d$ defined as

$$\theta\left[L\right] = \sum_{k\in[d]} \theta^k \cdot e_k\left[L\right]$$

we have

$$\begin{aligned}\mathbb{P}^{(\{X_k\,:\,k\in[d]\},\theta)}[X_{[d]}] &= \bigotimes_{k\in[d]} \mathbb{P}^{(\{X_k\},\theta^k)}[X_k] \\ &= \bigotimes_{k\in[d]} \mathbb{P}^k\left[X_k\right] \\ &= \mathbb{P}\left[X_{[d]}\right] .\end{aligned}$$

□

In Theorem 38 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 Chapter 9.

All atomic formulas can be selected by a single variable selecting tensor, that is

$$E^{(\{X_k\,:\,k\in[d]\},\theta)}[X_{[d]}] = \left\langle\left\{\mathcal{H}_V\left[X_{[d]}, L\right], \theta\left[L\right]\right\}\right\rangle\left[X_{[d]}\right] .$$

In case of negative $\theta\left[k\right]$ it is convenient to replace X_k by $\neg X_k$, in order to facilitate the interpretation. The probability distribution is left invariant, when also replacing $\theta\left[k\right]$ by $-\theta\left[k\right]$.

8.4.2 Boltzmann machines as MLNs

A Boltzmann machine is a member of an exponential family with the energy tensor

$$E^{W,b}[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] = \sum_{k,l\in[d]} W[L_{V,0} = k, L_{V,1} = l]x_k x_l + \sum_{k,l\in[d]} b[L_{V,0} = k] .$$

We notice, that this coincides with the energy tensor of a Markov Logic Network with formula set

$$\mathcal{F} = \{X_k \Leftrightarrow X_l \,:\, k, l \in [d]\} \cup \{X_k \,:\, k \in [d]\}$$

with cardinality $d^2 + d$.

Each formula is in the expressivity of an architecture consisting of a single binary logical neuron selecting any variable of $X_{[d]}$ in each argument and selecting connectives $\{\Leftrightarrow, \triangleleft\}$, where by $\triangleleft$ we refer to a connective passing the first argument, defined for $x_0 \in [m_0], x_1 \in [m_1]$ as

$$\triangleleft[X_0 = x_0, X_1 = x_1] = \mathcal{H}_V\left[X_0 = x_0, X_1, L_V = 0\right] .$$

The weight is

$$\theta = e_0\left[L_C\right] \otimes W + e_1\left[L_C\right] \otimes b[L_{V,0}] \otimes e_0\left[L_{V,0}\right]$$

And we have

$$E^{W,b}[X_{[d]}] = \left\langle\left\{\mathcal{H}_{\mathcal{A}}\left[X_{[d]}, L_C, L_{V,0}, L_{V,1}\right], \theta\left[L_C, L_{V,0}, L_{V,1}\right]\right\}\right\rangle\left[X_{[d]}\right] .$$

Often Boltzmann machines are formulated with hidden variables. To average those out, one needs to instantiate the probability distribution instead of the energy tensor and leave only visible variables open in a contraction.

Markov Logic Networks go beyond the Boltzmann machines already for binary formulas, by the flexibility to capture further dependencies beyond the correlation. We can use any binary logical connective and have an associated formula where we can put a weight on.

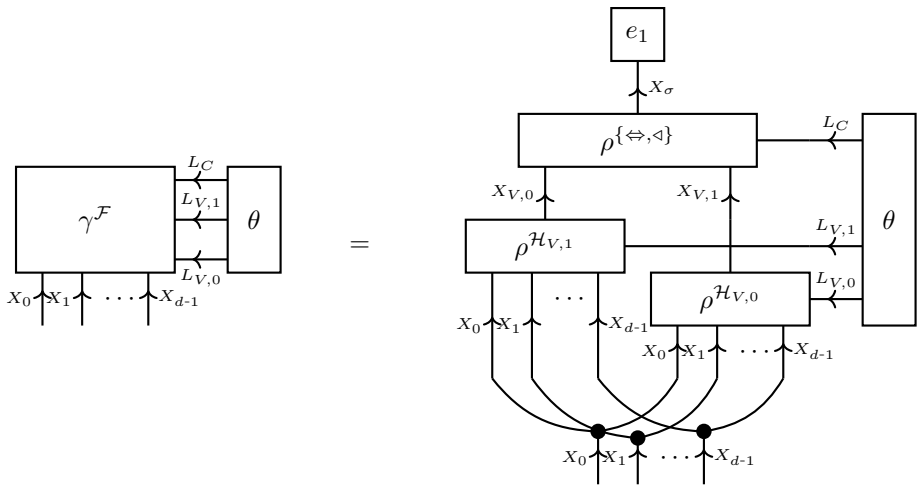


Figure 21: Tensor network representation of the energy of a Boltzmann machine

8.5 Applications

Markov Logic Networks as neuro-symbolic architectures:

- Neural Paradigm here by decompositions of logical formulas into their connectives. In more generality by decompositions of sufficient statistics into composed functions, using Basis Calculus. Deeper nodes as carrying correlations of lower nodes.
- Symbolic Paradigm by interpretability of propositional logics.

Markov Logic Networks as trainable Machine Learning models:

- Expressivity: Can represent any distributions, as shown by Theorem 37, with 2^d formulas.
- Efficiency: Can only handle small subsets of possible formulas, since their possible number is huge. Tensor networks provide means to efficiently represent formulas depending on many variables and reason based on contractions.
- Differentiability: Distributions are differentiable functions of their weights, see Parameter Estimation Chapter. The log-likelihood of data is therefore also differentiable function of their weights and we can exploit first-order methods in their optimization.
- Structure Learning: We need to find differentiable parametrizations of logical formulas respecting a chosen architecture. In Chapter 7 such representations are described based on Selector Tensor Networks.

9 Hard and Hybrid Logic Networks

Hard Logic Networks are Knowledge Bases, Hybrid Logic Networks are exponential families on Knowledge Bases. This makes it impossible to build energy tensors without basemeasures capturing vanishing coordinates.

Work in Theorem 29 to reduce entailment!

While exponential families are positive distributions, in logics probability distributions can assign states zero probability. As a consequence, Markov Logic Networks have a soft logic interpretation in the sense that violation of activated formulas have nonzero probability. We here discuss their hard logic counterparts, where worlds not satisfying activated formulas have zero probability.

Further we investigate, how both hard and soft logic factors can be combined to hybrid networks.

9.1 The limit of hard logic

The probability function of Markov Logic Networks with positive weights mimiks the tensor network representation of the knowledge base, which is the conjunction of the formulas. The maxima of the probability function coincide

with the models of the corresponding knowledge base, if the latter is satisfiable. However, since the Markov Logic Network is defined as a normed exponentiation of the weighted formula sum, it is a positive distribution whereas uniform distributions among the models of a knowledge base assign zero probability to world failing to be a model. Since both distributions are tensors in the same space to a factored system, we can take the limits of large weights and observe, that Markov Logic Networks indeed converge to normed knowledge bases.

Lemma 11. *When taking the limit of large weights $\theta_f \to \infty$ we observe a coordinatewise (in the sense of a convergence of each coordinate of the tensor) convergence*

$$\langle \exp[\theta_f \cdot f] \rangle \left[X_{[d]} | \varnothing\right] \to \langle f \rangle \left[X_{[d]} | \varnothing\right] \tag{35}$$

Proof. We have

$$\mathcal{Z}((\mathcal{F}, \theta)) = (\prod_k m_k - \langle f \rangle [\varnothing]) + \langle f \rangle [\varnothing] \cdot \exp[\theta_f]$$

and therefore for any $x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [2]$ with $f(x_0, \ldots, x_{d-1}) = 1$

$$\begin{aligned}\langle \exp[\theta_f \cdot f] \rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1} | \varnothing] &= \frac{\exp[\theta_f]}{(\prod_k m_k - \langle f \rangle [\varnothing]) + \langle f \rangle [\varnothing] \cdot \exp[\theta_f]} \\ &\to \frac{1}{\langle f \rangle [\varnothing]} = \langle f \rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1} | \varnothing] \ .\end{aligned}$$

For any $x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [2]$ with $f(x_0, \ldots, x_{d-1}) = 0$ we have on the other side

$$\begin{aligned}\langle \exp[\theta_f \cdot f] \rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1} | \varnothing] &= \frac{1}{(\prod_k m_k - \langle f \rangle [\varnothing]) + \langle f \rangle [\varnothing] \cdot \exp[\theta_f]} \\ &\to 0 = \langle f \rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1} | \varnothing] \ .\end{aligned}$$

□

Theorem 39. *Let $\mathcal{F}$ be a formulaset and θ a positive parameter vector. If the formula*

$$\mathcal{KB} = \bigwedge_{f \in \mathcal{F}} f$$

is satisfiable we have in the limit $\beta \to \infty$ the coordinatewise convergence

$$\mathbb{P}^{(\mathcal{F}, \beta \cdot \theta)}[X_{[d]}] \to \langle \mathcal{KB} \rangle \left[X_{[d]} | \right] .$$

Proof. Since $\mathcal{KB}$ is satisfiable we find $x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [2]$ with

$$\left\langle \exp\left[\sum_{f \in \mathcal{F}} \beta \cdot \theta_f \cdot f\right] \right\rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] = \exp\left[\beta \cdot \sum_{f \in \mathcal{F}} \theta_f\right]$$

and the partition function obeys

$$\left\langle \exp\left[\sum_{f \in \mathcal{F}} \beta \cdot \theta_f \cdot f\right] \right\rangle [\varnothing] \geq \exp\left[\beta \cdot \sum_{f \in \mathcal{F}} \theta_f\right] .$$

For any state $x_0, \ldots, x_{d-1} \in \bigtimes_{k \in [d]} [2]$ with $\mathcal{KB}(x_0, \ldots, x_{d-1}) = 0$ we find $h \in \mathcal{F}$ with $h(x_0, \ldots, x_{d-1}) = 0$ and have

$$\frac{\left\langle \exp\left[\sum_{f \in \mathcal{F}} \beta \cdot \theta_f \cdot f\right] \right\rangle [X_0 = x_0, \ldots, X_{d-1} = x_{d-1}]}{\left\langle \exp\left[\sum_{f \in \mathcal{F}} \beta \cdot \theta_f \cdot f\right] \right\rangle [\varnothing]} \leq \frac{\exp\left[\beta \cdot \sum_{f \in \mathcal{F}: f \neq h} \theta_f\right]}{\exp\left[\beta \cdot \sum_{f \in \mathcal{F}} \theta_f\right]} = \exp[\beta \cdot \theta_h] \to 0 \, .$$

The limit of the distribution has thus support only on the models of $\mathcal{KB}$. Since any model of $\mathcal{KB}$ has same energy at any β the limit is a uniform distribution and coincides therefor with

$$\langle \mathcal{KB} \rangle \left[X_{[d]} | \right] .$$

□

Remark 20 (More generic situation of simulated annealing)**.** *The process of taking $\beta \to \infty$ is known as simulated annealing, see Chapter 4. From the discussion there we have the more general statement, that the limiting distribution is the uniform distribution among the maxima of $\mathbb{P}^{(\mathcal{F}, \theta)}[X_{[d]}]$. If the formula $\mathcal{KB}$ is not satisfiable the normation $\langle \mathcal{KB} \rangle \left[X_{[d]} | \varnothing\right]$ does not exist and the limit distribution has another syntactical representation, to be gained e.g. by minterm or maxterm representation (see Theorem 23).*

9.2 Hard Logic Networks

Hard Logic Network coincide with Knowledge Bases. We use $\wedge$ symmetry to represent them as a contraction of the formulas building the Knowledge Base as conjunction.

Theorem 40 (Conjunction Decomposition of Knowledge Bases)**.** *For a Knowledge Base*

$$\mathcal{KB} = \bigwedge_{f \in \mathcal{F}} f$$

we have

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

and

$$\mathcal{KB}\left[X_{[d]}\right] = \left\langle \left\{\rho^f\left[X_f, X_{[d]}\right] \,:\, f \in \mathcal{F}\right\} \cup \left\{e_1\left[X_f\right] \,:\, f \in \mathcal{F}\right\}\right\rangle \left[X_{[d]}\right] .$$

Proof. By the $\wedge$-symmetry, see effective calculus and

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

□

Remark 21. *$\wedge$ symmetry does not generalize to Markov Logic Networks In Markov Logic, similar decompositions are not possible. For example, consider a MLN with a single formula $X_0 \wedge X_1$ and nonvanishing weight θ. This does not coincide with the distribution of a MLN of two formulas X_0 and X_1. To see this, we notice that with respect to the distribution of the first MLN, both variables are not independent, while for any MLN constructed by the two atomic formulas they are.*

9.2.1 Conjunctive Normal Representation

Poly reps here?

We can now apply the representation symmetries to represent a propositional Knowledge Base in conjunctive normal form. A Knowledge Base in Conjunctive Normal Form is a conjunction of clauses, where clauses are disjunctions of literals being atoms (positive literals) or negated atoms (negative literals).

9.2.2 Polynomial Representation of Formulas

We can further derive representation schemes for Knowledge Bases, which are in conjunctive normal forms.

Formulas can be represented as sparse polynomials, which will be discussed in more detail in Chapter 19 (see Definition 75).

Lemma 12. *Any term is representable by a single monomial and any clause is representable by at most two monomials.*

Proof. Let $\mathcal{V}_0$ and $\mathcal{V}_1$ be disjoint subsets of $\mathcal{V}$, then we have

$$Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1} = e_{\{x_k=0:k\in\mathcal{V}_0\}\cup\{x_k=1:k\in\mathcal{V}_1\}}\left[X_{\mathcal{V}_0\cup\mathcal{V}_1}\right] \otimes \mathbb{I}\left[X_{\mathcal{V}/(\mathcal{V}_0\cup\mathcal{V}_1)}\right]$$

and

$$Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1} = \mathbb{I}\left[X_{\mathcal{V}}\right] - e_{\{x_k=0:k\in\mathcal{V}_0\}\cup\{x_k=1:k\in\mathcal{V}_1\}}\left[X_{\mathcal{V}_0\cup\mathcal{V}_1}\right] \otimes \mathbb{I}\left[X_{\mathcal{V}/(\mathcal{V}_0\cup\mathcal{V}_1)}\right] .$$

We notice, that any tensors $\mathbb{I}$ and $e_x \otimes \mathbb{I}$ habe basis+-rank of 1 and therefore $Z^{\wedge}_{\mathcal{V}_0,\mathcal{V}_1}$ of 1 and $Z^{\vee}_{\mathcal{V}_0,\mathcal{V}_1}$ of at most 2. □

We apply Lemma 12 to show the following sparsity bound on the energy tensor of Markov Logic Networks.

Theorem 41. *Any formula f with a conjunctive normal form of n clauses satisfies*

$$\tilde{\ell}\left(f\right) \leq 2^n .$$

For any set $\mathcal{F}$ of formulas each with a conjunctive normal form of n_f clauses satisfies for any θ

$$\tilde{\ell}\left(\sum_{f\in\mathcal{F}} \theta_f \cdot f\right) \leq \sum_{f\in\mathcal{F}} 2^{n_f} .$$

Proof. Let f have a CNF with clauses indexed by $l \in [n]$ and each clause represented by subsets $\mathcal{V}_0^l, \mathcal{V}_1^l$, that is

$$f = \bigwedge_{l \in [n]} Z^{\vee}_{\mathcal{V}_0^l, \mathcal{V}_1^l} \ .$$

We now use the rank bound of Theorem 99 and Lemma 12 to get

$$\tilde{\ell}(f) \leq \prod_{l \in [n]} \tilde{\ell}\left(Z^{\vee}_{\mathcal{V}_0^l, \mathcal{V}_1^l}\right) \leq 2^n \, .$$

Given a collection of formulas $\mathcal{F}$, each with a CNF of n_f clauses we apply Theorem 98 and get

$$\tilde{\ell}\left(\sum_{f \in \mathcal{F}} \theta_f \cdot f\right) \leq \sum_{f \in \mathcal{F}} \tilde{\ell}(f) \leq \sum_{f \in \mathcal{F}} 2^{n_f} \, .$$

□

9.3 Hybrid Logic Network

Markov Logic Networks are by definition positive distributions. In contrary, Hard Logic Networks model uniform distributions over model sets of the respective Knowledge Base and therefore have vanishing coordinates. We now show how to combine both approaches by defining Hybrid Logic Networks. We orient on Example 3.6 in Wainwright and Jordan (2008) and choose hard constraints as a base measure $\nu^{\mathcal{KB}}$, probabilistic soft formulas as sufficient statistics as before.

Definition 50. *Given a set of formulas $\mathcal{F}$ with weights θ and set $\mathcal{KB}$ of formulas, which conjunction is satisfiable, the hybrid logic network is the probability distribution*

$$\mathbb{P}^{(\mathcal{F},\theta,\nu^{\mathcal{KB}})}[X_{[d]}] = \langle \{f : f \in \mathcal{KB}\} \cup \{\exp[\theta_f \cdot f] : f \in \mathcal{F}\} \rangle \left[X_{[d]} | \varnothing\right] \ ,$$

which is the member of the exponential family with statistic by $\mathcal{F}$ and the base measure

$$\nu^{\mathcal{KB}}[X_{[d]}] = \langle \{f : f \in \mathcal{KB}\} \rangle \left[X_{[d]}\right] \ .$$

The assumption of a satisfiable set $\mathcal{KB}$ is necessary, as we show next.

Theorem 42. *If any only if $\bigwedge_{f \in \mathcal{KB}} f$ is satisfiable, the tensor*

$$\langle \{f : f \in \mathcal{KB}\} \cup \{\exp[\theta_f \cdot f] : f \in \mathcal{F}\} \rangle \left[X_{[d]}\right]$$

is normable.

Proof. We need to show that

$$\langle \{f : f \in \mathcal{KB}\} \cup \{\exp[\theta_f \cdot f] : f \in \mathcal{F}\} \rangle [\varnothing] > 0 \, . \tag{36}$$

Since the conjunction of $\mathcal{KB}$ is satisfiable we find a $x_{[d]}$ with $f\left[X_{[d]} = x_{[d]}\right] = 1$ for all $f \in \mathcal{KB}$. Then

$$\begin{aligned}
\langle \{f : f \in \mathcal{KB}\} \cup \{\exp[\theta_f \cdot f] : f \in \mathcal{F}\} \rangle \left[X_{[d]} = x_{[d]}\right] &= \left(\prod_{f \in \mathcal{KB}} f\left[X_{[d]} = x_{[d]}\right]\right) \cdot \left(\prod_{f \in \mathcal{F}} \exp[\theta_f \cdot f]\left[X_{[d]} = x_{[d]}\right]\right) \\
&= \left(\prod_{f \in \mathcal{F}} \exp[\theta_f \cdot f]\left[X_{[d]} = x_{[d]}\right]\right) \\
&> 0 \, .
\end{aligned}$$

Condition (36) follows from this and the Hybrid Logic Network is well-defined. □

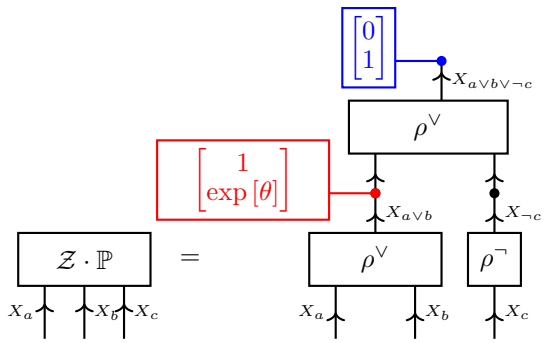


Figure 22: Diagram of a formula tensor with activated heads, containing hard constraint cores and probabilistic weight cores .

9.3.1 Tensor Network Representation

We can employ the formula decompositions to represent both probabilistic facts of the MLN and hard facts (seen as the limit of large weights).

Theorem 43. *For any hybrid logic network we have*

$$\mathbb{P}^{(\mathcal{F},\theta,\mathcal{KB})}[X_{[d]}] = \left\langle \{\rho^f\left[X_f, X_{[d]}\right] : f \in \mathcal{F}\cup\mathcal{KB}\} \cup \{e_1\left[X_f\right] : f \in \mathcal{KB}\} \cup \{W^f[X_f] : f \in \mathcal{F}\}\right\rangle \left[X_{[d]}|\varnothing\right] .$$

Proof. By Lemma 7. □

For an example see Figure 22.

Remark 22. *Probability interpretation using the Partition function The tensor networks here represent unnormalized probability distributions. The probability distribution can be normed by the quotient with the naive contraction of the network, the partition function.*

9.3.2 Reasoning Properties

Theorem 44. *Let $(\mathcal{F},\theta,\mathcal{KB})$ define a Hybrid Logic Network. Given a query formula f we have that*

$$\mathbb{P}^{(\mathcal{F},\theta,\mathcal{KB})} \models f$$

if and only if

$$\mathcal{KB} \models f .$$

Proof. Application of Theorem 29 on the representation of Hybrid Logic Networks as Markov Networks in Theorem 43. □

Formulas in $\mathcal{F}$, which are entailed or contradicted by $\mathcal{KB}$ are redundant

Theorem 45. *If for a formula f and $\mathcal{KB}$ we have*

$$\mathcal{KB} \models f \quad \text{or} \quad \mathcal{KB} \models \neg f$$

then for any $(\mathcal{F},\theta,\mathcal{KB})$

$$\mathbb{P}^{(\mathcal{F}/\{f\},\tilde{\theta},\mathcal{KB})}\left[X_{[d]}\right] = \mathbb{P}^{(\mathcal{F},\theta,\mathcal{KB})}\left[X_{[d]}\right] ,$$

where $\tilde{\theta}$ denotes the tensor θ, where the coordinate to f is dropped, if $f \in \mathcal{F}$.

Proof. Isolate the factor to the hard formula, which is constant for all situations. □

A similar statement holds for the hard formulas itself, as shown in Theorem 25. However, notice that if $\mathcal{KB}/\{f\} \models \neg f$, then $\mathcal{KB}\cup\{f\}$ is not satisfiable and a hybrid logic network cannot be defined for $\mathcal{KB}\cup\{f\}$ as hard logic formulas.

9.3.3 Expressivity

Hybrid Logic Networks extend the expressivity result of Theorem 37 to arbitrary probability tensors, dropping the positivity constraints for Markov Logic Networks.

Theorem 46. *Let $\mathbb{P}\left[X_{[d]}\right]$ a possibly not positive probability tensor we build a base measure*

$$\nu^{\mathcal{KB}} = \chi\left(\mathbb{P}\left[X_{[d]}\right]\right)$$

and a parameter tensor

$$\theta\left[L_{[d]} = x_{[d]}\right] = \begin{cases} 0 & \textit{if} \quad \mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0 \\ \ln\left[\mathbb{P}\left[X_{[d]} = x_{[d]}\right]\right] & \textit{else} \end{cases} .$$

Then the probability tensor is the member of the minterm exponential family with base measure $\mathcal{KB}$ and parameter θ, that is

$$\mathbb{P}\left[(\mathcal{F}_\wedge, \theta, \nu^{\mathcal{KB}})\right]$$

Proof. It suffices to show that

$$\left\langle\left\{\nu^{\mathcal{KB}}, \exp\left[\left\langle\gamma^{\mathcal{F}_\wedge}\theta\right\rangle\left[X_{[d]}\right]\right]\right\}\right\rangle\left[X_{[d]}\right] = \mathbb{P}\left[X_{[d]}\right] .$$

For indices $x_{[d]}$ with $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] = 0$ we have $\nu^{\mathcal{KB}}\left[X_{[d]} = x_{[d]}\right] = 0$ and thus also

$$\left\langle\left\{\nu^{\mathcal{KB}}, \exp\left[\left\langle\gamma^{\mathcal{F}_\wedge}\theta\right\rangle\left[X_{[d]}\right]\right]\right\}\right\rangle\left[X_{[d]} = x_{[d]}\right] = 0 .$$

For indices $x_{[d]}$ with $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0$ we have $\nu^{\mathcal{KB}}\left[X_{[d]} = x_{[d]}\right] = 1$ and

$$\begin{aligned}\left\langle\left\{\nu^{\mathcal{KB}}, \exp\left[\left\langle\gamma^{\mathcal{F}_\wedge}\theta\right\rangle\left[X_{[d]}\right]\right]\right\}\right\rangle\left[X_{[d]} = x_{[d]}\right] &= \prod_{l_{[d]}} \exp\left[\theta\left[L_{[d]} = l_{[d]}\right] \cdot Z^{\wedge}_{l_{[d]}}\left[X_{[d]} = x_{[d]}\right]\right] \\ &= \exp\left[\theta\left[L_{[d]} = x_{[d]}\right]\right] \\ &= \mathbb{P}\left[X_{[d]} = x_{[d]}\right] .\end{aligned}$$

□

9.4 Categorical Constraints

Also called atomization of categorical variables.

We made the assumption that all categorical variables in factored systems to be represented by propositional logics take binary values (i.e. $m = 2$). In cases where a categorical variable X takes multiple values we define for each x an atomic formula X_x representing whether X is assigned by x in a specific state. Following this construction we have the constraint that exactly one of the atoms X_x is 1 at each state.

To capture the constraints resulting from this construction we introduce auxiliary parts. Such constraints can also be expressed by a formula but would result in an unnecessary large tensor network.

Definition 51 (Categorical Constraint)**.** *Given a list $X_0, \ldots, X_{m-1}$ of binary variables and a categorical variable X with dimension m a categorical constraint is a tensor $Z[X, X_{[m]}]$ defined as*

$$Z(x, x_A) = \begin{cases} 1 & \textit{if} \quad x_{[m]} = e_x \\ 0 & \textit{else} . \end{cases}$$

With Theorem 85 the tensor representation of ρ^Z decomposes in a basis CP format (see Figure 23b) of if its coordinate maps Z_x, where $x \in [m]$. For the cores

$$\rho^{Z_x} = e_x\left[X\right] \otimes e_1\left[X_x\right] + \left(\mathbb{I}\left[X\right] - e_x\left[X\right]\right) \otimes e_0\left[X_x\right] \tag{37}$$

we have by Theorem 85

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

In the next theorem we show how a categorical constraint can be enforced in a tensor network by adding the tensor Z to a contraction.

Theorem 47. *For any tensor $T\left[X_{[d]}\right]$ and a categorical constraint defined by an ordered subset $X_A \subset X_{[d]}$, a variable $X \in X_{[d]}$ we have*

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

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

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

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

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

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

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

□

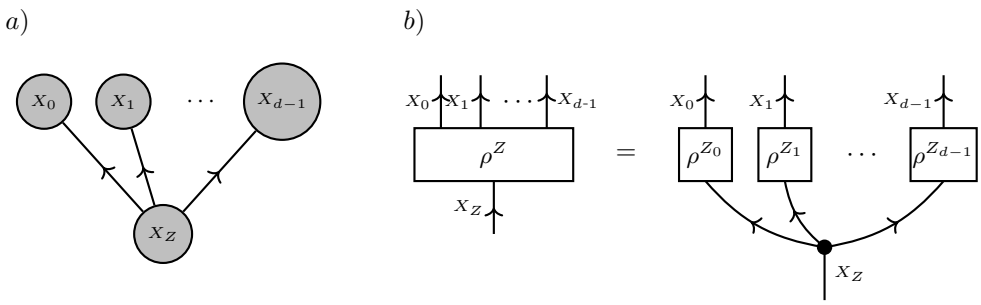


Figure 23: Representation of a categorical constraint in a CP Format tensor network. a) Representation of the dependency of the graphical model. b) Tensor Representation with further network decomposition. We average by contraction with the dashed tensor $\mathbb{I}$, if we do not specify the active atom.

Remark 23 (Combination of Constraints)**.** *We can combine constraint cores by Hadamard products in the dual tensor network representation, as long as they can be satisfied together. An example, where this is not the case, are the categorical constraints to the three sets*

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

Besides the categorical cores also the datacores have a similar bayesian network affecting the atoms by another hidden variable. Combining both is welldefined, only when all datapoints satisfy the categorical constraints (that is only one of the atoms in each constraint is active).

Example 11 (Sudoku)**.** *An interesting example, where categorical constraints are combined is Sudoku, the game of assigning numbers to a grid. For a $n \in \mathbb{N}$ we define variables $X_{i,j,k}$ where $i, j, k \in [n^2]$ and $m_{i,j,k} = 2$. By understanding i as a line index and j as a column index, they are ordered in a grid as sketched in Figure 11 in the case $n = 3$.*

At each position there is a single number, that is for each $i, j \in [n^2]$ have a constraint

$$\{X_{i,j,k} \,:\, k \in [n^2]\}$$

Here the n^6 variables are the random variables whether a specific position has a specific number assigned. The $3 \cdot n^2$ constraints are

- *Each number k appears exactly once in each row $i \in [n^2]$:*

$$\{X_{i,j,k} \,:\, j \in [n^2]\}$$

- *Each number k appears exactly once in each column $j \in [n^2]$:*

$$\{X_{i,j,k} \,:\, i \in [n^2]\}$$

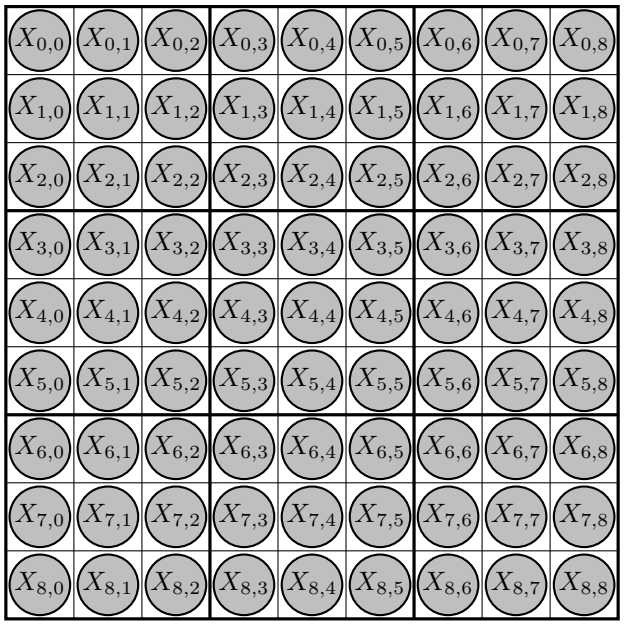


Figure 24: Sudoku grid of categorical variables, here drawn in the standard case of $n = 3$, each with dimension $m = n^2 = 9$.

- *Each number appears exactly once in each square* $s, r \in [n]$

$$\{X_{i+n\cdot s, j+n\cdot r, k} \,:\, i, j \in [n]\}$$

In total we have $3 \cdot n^2 + n^4$ *constraints for* n^6 *variables.*

Reasoning by Entailment propagation! Also, probabilistic choices possible when exact (!) contraction at a position not a basis vector, then can choose one possibility.

10 Parameter Estimation

We estimate the canonical parameters θ in an exponential family. We first discuss parameter estimation in the more generic situation of exponential families and then discuss the more specific situation of Markov Logic Networks.

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

10.1 Learning Markov Logic Networks

Markov Logic Networks are exponential families with statistics by a set $\mathcal{F}$ of propositional formulas. We furthermore allow for base measures, to also include the discussion of Hybrid Logic Networks. Based on this fact, we can apply the theory of probabilistic inference, developed in Chapter 4.

The Maximum Likelihood Problem on MLNs is the estimation

$$\operatorname{argmax}_\theta \langle \{\theta, \mu_D\} \rangle [\varnothing] - A(\theta) \,.$$

where

$$\mu_D = \left\langle \left\{ \gamma^{\mathcal{F}}, \mathbb{P}^D \right\} \right\rangle [L] \quad \text{and} \quad A(\theta) = \left\langle \left\{ \exp \left[\left\langle \left\{ \gamma^{\mathcal{F}}, \theta \right\} \right\rangle \left[X_{[d]} \right] \right], \nu \right\} \right\rangle [\varnothing] \,.$$

The Maximum Entropy Problem for Markov Logic Networks is

$$\operatorname{argmax}_{\mathbb{P}} \mathbb{H} [\mathbb{P}] \quad \text{subject to} \quad \left\langle \left\{ \mathbb{P}, \gamma^{\mathcal{F}} \right\} \right\rangle [L] = \mu_D [L] \tag{38}$$

Corollary 7 (of Theorem 20)**.** *Among all distributions $\mathbb{P}$ of $\bigtimes_{k\in[d]}[2]$ satisfying $\langle\{\mathbb{P},\gamma^{\mathcal{F}}\}\rangle[L] = \langle\{\mathbb{P}^D,\gamma^{\mathcal{F}}\}\rangle[L]$ the Markov Logic Network with formulas $\mathcal{F}$ and weights θ being the solution of the maximum likelihood problem has minimal entropy.*

We notice, that the solution of the maximum entropy problem is thus a Markov Logic Network. This is remarkable, because this motivates our restriction to Markov Logic Networks as those distributions with maximal entropy given satisfaction rates of formulas in $\mathcal{F}$.

When having a prior $\mathbb{P}[(\mathcal{F},\theta)]$ over the Markov Logic Networks we alternatively want to find the parameters $(\mathcal{F},\theta)$ solving the maximum a posteriori problem

$$\operatorname{argmax}_{(\mathcal{F},\theta)}\mathbb{P}^{(\mathcal{F},\theta)}[\{D(j)\}_{j\in[m]}]\cdot\mathbb{P}[(\mathcal{F},\theta)]\ . \tag{39}$$

10.2 Alternating Algorithms to Approximate the Backward Map

Here we develop the algorithms of Section 4.5.4 in the special case of MLNs.

Let us now discuss an implementation of the Alternating Moment Matching Algorithm 4 in case of Markov Logic Networks. To solve the moment matching condition at a formula f_l we refine Lemma 6 in the following.

Lemma 13. *Let there be a base measure ν, a formula selecting map $\mathcal{F} = \{f_l : l\in[p]\}$ and weights θ, and choose $l\in[p]$ such that $f_l \notin \{\mathbb{I}\left[X_{[d]}\right], 0\left[X_{[d]}\right]\}$. The moment matching condition relative to θ, $l\in[p]$ and $\mu_D[L=l]\in(0,1)$ is then satisfied, if*

$$\theta[L=l] = \ln\left[\frac{\mu_D[L=l]}{(1-\mu_D[L=l])}\cdot\frac{T[X_{f_l}=0]}{T[X_{f_l}=1]}\right]$$

where by $T[X_{f_l}]$ we denote the contraction

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

Proof. Noticing that since $\operatorname{im}(f_l)\subset[2]$

$$\mathrm{Id}|_{\operatorname{im}(f_l)} = e_1[X_{f_l}]$$

the moment matching condition is by Lemma 6 satisfied if

$$\langle\{W^l, e_1, T\}\rangle[\varnothing] = \langle\{W^l, T\}\rangle[\varnothing]\cdot\mu_D[L=l]\ .$$

This is equal to

$$\exp[\theta_l]\cdot T[X_{f_l}=1] = (\exp[\theta_l]\cdot T[X_{f_l}=1] + T[X_{f_l}=0])\cdot\mu_D[L=l]\ .$$

Rearranging the equations this is equal to

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

We notice that the right side is well defined, since we have by assumption $\mu_D[L=l], (1-\mu_D[L=l])\neq 0$ and $T[X_{f_l}=0], T[X_{f_l}=1]\neq 0$ since Markov Logic networks are positive distributions and $f_l\notin\{\mathbb{I}\left[X_{[d]}\right], 0\left[X_{[d]}\right]\}$. □

In the case $\mu_D[L=l]\in\{0,1\}$ the moment matching conditions are not satisfiable for $\theta_f\in\mathbb{R}$. But, we notice, that in the limit $\theta_f\to\infty$ (respectively $-\infty$) we have

$$\mu_l\to 1\quad\text{(respectively 0)}\,,$$

and the moment matching can be satisfied up to arbitrary precision. In Chapter 9 we will allow infinite weights and interpret the corresponding factors by logical formulas. As a consequence, we will able to fit graphical models, which we will call hybrid networks on arbitrary satisfiable mean parameters.

The cases $T[X_{f_l}=1]=0$, respectively $T[X_{f_l}=1]=0$ only appear for nontrivial formulas when the distribution is not positive. This is not the case for Markov Logic Networks, but will happen when formulas are added as cores of a Markov Network. This situation will be investigated in Chapter 9.

Algorithm 7 Alternating Weight Optimization (AWO)

for $f \in \mathcal{F}$ **do**
  Compute
$$T\left[X_f\right] = \langle \{f\} \rangle \left[X_f\right]$$
  Set
$$\theta[L = l] = \ln\left[\frac{\mu_D\left[L = l\right]}{\left(1 - \mu_D\left[L = l\right]\right)} \cdot \frac{T\left[X_{f_l} = 0\right]}{T\left[X_{f_l} = 1\right]}\right]$$
end for
while Stopping criterion is not met **do**
  for $f \in \mathcal{F}$ **do**
    Compute
$$T\left[X_{f_l}\right] = \left\langle \{\rho^{f_l} \,:\, l \in [p]\} \cup \{W^{\tilde{l}} : \tilde{l} \in [p], \tilde{l} \neq l\} \cup \{\nu\} \right\rangle \left[X_{f_l}\right]$$
    Set
$$\theta[L = l] = \ln\left[\frac{\mu_D\left[L = l\right]}{\left(1 - \mu_D\left[L = l\right]\right)} \cdot \frac{T\left[X_{f_l} = 0\right]}{T\left[X_{f_l} = 1\right]}\right]$$
  end for
end while

Since the likelihood is concave (see Koller Book), there are not local maxima the coordinate descent could run into and coordinate descent will give a monotonic improvement of the likelihood.

We suggest an alternating optimization by Algorithm 7, solving the moment matching equation iteratively for all formulas $f \in \mathcal{F}$ and repeat the optimization until a convergence criterion is met.

In the initialization phase of Algorithm 7, each parameters is initialized relative to a uniform distribution. The algorithm would be finished, if the variables X_f are independent. This would be the case, if the Markov Logic Network consists of atomic formulas only. When they fail to be independent, the adjustment of the weights influence the marginal distribution of other formulas and we need an alternating optimization. This situation corresponds with couplings of the weights by a partition contraction, which does not factorize into terms to each formula.

Solving Equation **??** requires inference of a current model by answering the query to the formula $\tilde{f}$. This can be a bottleneck and circumvented by approximative inference, see e.g. CAMEL [Ganapati et al.].

Remark 24 (Grouping of coordinates with trivial sum)**.** *When having a set of coordinates, such that the coordinate functions are binary and sum to the trivial tensor, one can find simultaneous updates to the canonical parameters, such that the partition function is staying invariant. Given a parameter θ^t we compute*
$$\mu^t = \left\langle \mathbb{P}^{(\phi,\theta^t)}, \phi \right\rangle [L]$$
and build the update
$$\theta^{t+1} = \theta^t + \ln\left[\mu^D\right] \mu^t \,.$$
Then, θ^{t+1} satisfies the moment matching equations for all coordinates in the set.

The assumptions are met when taking all features to any hyperedge in a Markov Network seen as an exponential family. In that case, the update algorithm is refered to as Iterative Proportional Fitting Wainwright and Jordan (2008). Further, when activating both f and $\neg f$.

10.3 Forward and backward mappings of MLNs in closed form

We recall from Chapter 4, that while forward mappings are always in closed form by contractions, backward mapping in general fail. We here investigate specific examples, where closed forms can be derived for both.

We have formulated parameter estimation as a maximum entropy problem constrained to matching expected sufficient statistics. Let us discuss situations, where the forward and backward mappings are available in closed form and parameter estimation can thus be solved by application of the inverse on the expected sufficient statistics with respect to the empirical distribution. When the backward map $B^{(\phi,\nu)}$ is available in closed form, we directly get optimal parameters by the inversion acting on the satisfaction rate and can avoid iterative algorithms of parameter estimation.

10.3.1 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 Section 8.3 how to represent any distribution as a MLN of maxterms (respectively minterms), see Theorem 37.

We use the tuple enumeration of the maxterms and minterms by $\bigtimes_{k\in[d]}[2]$ introduced in Section 5.3.3. With respect to this enumeration the canonical parameters and mean parameters are tensors in $\bigotimes_{k\in[d]}\mathbb{R}^2$.

This is since the sufficient statistics is the identity!

The mean parameters for the minterm and the maxterm exponential family have interpretations

$$\mu\left[L_{[d]}=x_{[d]}\right]=\mathbb{P}\left[x_{[d]}\right]$$

and therefore after a relabeling of categorical variables to selection variables$\mu=\mathbb{P}$.

For maxterms we have analogously

$$\mu\left[L_{[d]}=x_{[d]}\right]=1-\mathbb{P}\left[x_{[d]}\right]$$

and $\mu=\mathbb{I}\left[\right]-\mathbb{P}$.

Theorem 48. *Given the Markov Logic Networks to the formula sets*

$$\mathcal{F}_\wedge:=\{Z^\wedge_{x_0,\dots,x_{d-1}}\,:\,x_0,\dots,x_{d-1}\in\bigtimes_{k\in[d]}[2]\}\quad\textit{and}\quad\mathcal{F}_\vee:=\{Z^\vee_{x_0,\dots,x_{d-1}}\,:\,x_0,\dots,x_{d-1}\in\bigtimes_{k\in[d]}[2]\}$$

of all minterms, respectively of all mapterms, the forward mapping are

$$F^\wedge(\theta)=\langle\exp\left[\theta\right]\rangle\left[X_{[d]}|\varnothing\right]\quad\textit{and}\quad F^\vee(\theta)=\langle\exp\left[-\theta\right]\rangle\left[X_{[d]}|\varnothing\right]\ ,$$

where in a slight abuse of notation we assigned the variables $X_{[d]}$ to the canonical parameters θ.

Possible choices of the backward mappings are

$$B^\wedge(\mu)=\ln\left[\mu\right]\quad\textit{and}\quad B^{\mathcal{F}_\vee}(\mu)=-\ln\left[\mu\right]\ .$$

Proof. For the minterms we use that

$$\mathcal{F}_\wedge[X_{[d]},X_{\mathcal{F}_\wedge}]=\delta\left[X_{[d]},X_{\mathcal{F}_\vee}\right]$$

and get

$$F^\wedge(\theta)=\left\langle\exp\left[\langle\{\mathcal{F}_\wedge,\theta\}\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right]=\langle\exp\left[\theta\right]\rangle\left[X_{[d]}|\varnothing\right]\ .$$

We notice that for any μ in the image of the forward map we have

$$F^\wedge(B^\wedge(\mu))=\mu$$

Therefor, $B^{\mathcal{F}_\wedge}$ is indeed a backward mapping to the exponential family of minterms.

For the maxterms we use that

$$\mathcal{F}_\vee[X_{[d]},X_{\mathcal{F}_\vee}]=\mathbb{I}\left[X_{[d]},X_{\mathcal{F}_\vee}\right]-\delta\left[X_{[d]},X_{\mathcal{F}_\vee}\right]$$

and get

$$\begin{aligned}F^\vee(\theta)&=\left\langle\exp\left[\langle\{\mathcal{F}_\wedge,\theta\}\rangle\left[X_{[d]}\right]\right]\right\rangle\left[X_{[d]}|\varnothing\right]\\&=\left\langle\{\exp\left[\langle\{\mathbb{I},\theta\}\rangle\left[X_{[d]}\right]\right],\exp\left[-\langle\theta\rangle\left[X_{[d]}\right]\right]\}\right\rangle\left[X_{[d]}|\varnothing\right]\\&=\langle\exp\left[-\theta\right]\rangle\left[X_{[d]}|\varnothing\right]\end{aligned}$$

where we used, that $\exp\left[\langle\{\mathbb{I},\theta\}\rangle\left[X_{[d]}\right]\right]$ is a multiple of $\mathbb{I}\left[X_{[d]}\right]$ and is thus eliminated in the normation. For any $\mu\in\operatorname{im}(F^\vee)$ we have

$$F^\vee(B^\vee(\mu))=\mu$$

and $B^\wedge$ is thus a backward map for the exponential family of maxterms. □

Any positive probability distribution can thus be fitted by minterms when we choose $\theta=\ln\left[\mathbb{P}\right]$, respectively by maxterms when we choose $\theta=\ln\left[\mathbb{P}\right]$. Thus, we have identified a subset of 2^d formulas, which is rich enough to fit any distribution.

10.3.2 Atomic formulas

Let us now derive a closed form backward mapping for the statistic

$$\mathcal{F}_{[d]} := \{X_k : k \in [d]\} .$$

The mean parameters coincide with the queries on the atomic formulas, that is the marginal

$$\mu [L = k] = \mathbb{P} [X_k = 1] .$$

Theorem 49. *Given a Markov Logic Network with the statistic $\mathcal{F}_{[d]}$ of atomic formulas, the forward mapping from canonical parameters to mean parameters is the coordinatewise sigmoid, that is*

$$F^{[d]}(\theta [L]) = \frac{\exp [\theta [L]]}{\mathbb{I} [L] + \exp [\theta [L]]}$$

where the quotient is performed coordinatewise.

A backward mapping is the coordinatewise logit, that is

$$B^{[d]}(\mu [L]) = \ln \left[\frac{\mu [L]}{\mathbb{I} [L] - \mu [L]} \right] .$$

Proof. We have for any $\theta [L] \in \mathbb{R}^d$

$$\mathbb{P}^{(\mathcal{F}_{[d]},\theta)} \left[X_{[d]} \right] = \bigotimes_{k \in [d]} \langle \exp [\theta [L = k] \cdot X_k] \rangle [X_k | \varnothing] .$$

For any $k \in [d]$ it therefore holds, that

$$\begin{aligned} F^{[d]}(\theta [L])[L = k] &= \left\langle \left\{ X_k, \mathbb{P}^{(\mathcal{F}_{[d]},\theta)} \left[X_{[d]} \right] \right\} \right\rangle [\varnothing] \\ &= \langle \{ X_k, \langle \exp [\theta [L = k] \cdot X_k] \rangle [X_k | \varnothing] \} \rangle [\varnothing] \\ &= \frac{\exp [\theta [L = k]]}{1 + \exp [\theta [L = k]]} . \end{aligned}$$

Since the coordinatewise logit is the inverse function of the coordinatewise sigmoid the map

$$B^{[d]}(\mu [L])[L = k] = \ln \left[\frac{\mu [L = k]}{1 - \mu [L = k]} \right]$$

satisfies for any μ in the image of the forward map

$$F^{[d]}(B^{[d]}(\mu)) = \mu$$

and is therefore a backward map. □

In a selection tensor networks they are represented by a single neuron with identity connective and variable selection to all atoms. To architectures: Redo the discussions on these examples.

The maximum likelihood estimator of a positive probability distribution by the MLN of atomic formulas is therefore the tensor product of the marginal distributions.

Remark 25. *By Independence Decomposition we reduce to a system of atomic MLN. The minterms of such MLNs are the literals. By redundancy (literals sum up to $\mathbb{I}$), it suffices to take only the positive or the negative literal.*

10.4 Parameter Estimation in Hybrid Networks

Modify alternating weight optimization to deal with situations $\mu [L = l] \in \{0, 1\}$: Add those as facts as base measures.

When there are facts, also situations $T [X = 1] \in \{0, 1\}$ can appear. It that case the formula is entailed or contradicted by the facts, and dropping should be considered in both cases.

The max entropy - max likelihood duality still holds for hybrid logic networks as we show next.

Theorem 50. *Given a set of formulas $\tilde{\mathcal{F}}$ and $\tilde{\mu}$, with coordinates $\tilde{\mu}_l \in [0,1]$ in the closed interval $[0,1]$. If the corresponding maximum entropy problem is feasible, its solution is a hybrid logic network with*

- $\mathcal{KB} = \{f_l : l, \mu[L = l] = 1\} \cup \{\neg f_l : l, \mu_l = 0\}$
- $\mathcal{F} = \{f_l : l, \mu_l \in (0,1)\}$
- *θ being the backward map evaluated at the vector μ consisting of the coordinates of $\tilde{\mu}$ not in $\{0,1\}$*

Proof. Feasible distributions have a density with base measure by $\mathcal{KB}$, we therefore reduce the set of distributions in the argmax to those with density to the base measure. The max entropy is a max entropy problem with respect to that base measure, where we only keep the constraints to the mean parameters different from $\{0,1\}$ (those are trivially satisfied). The statement then follows from the generic property (see Sec3.1 in Wainwright and Jordan (2008)). □

11 Structure Learning

Change loss to be function of distribution, not of canparam.

After having investigated how to parameterize exponentially many formulas by formula selecting networks, let us now develop objectives to be optimized.

We will use the effective selection tensor network representation of exponentially many formulas described in Chapter 7.

11.1 Overparametrization

Another exponent in the number of possible subsets of formulas!

Taking $\mathcal{F}$ to consist of all propositional formulas, we get a massive overparametrization: The essential statistics maps to a $2^{\left(2^d\right)}$ dimensional real vector space. All possible distributions of the d atomic variables are mapped to an $2^d - 1$ dimensional submanifold, where also the essential statistics maps to.

Thus, to identify probabilistic knowledge bases, we need to drastically restrict the shape of formulas allowed. It is in principle impossible to decide which formulas to be activated, based only on statistics and not on prior assumptions.

When having d atoms, there are 2^d states in the factored system. Since each state can either be a model of a formula or not, there are

$$|\mathcal{F}| = 2^{\left(2^d\right)}$$

formulas. Having, for example, $d = 10$, then $|\mathcal{F}| > 10^{308}$.

One regularization is by allowing only a small number of formulas to be active. We add active formulas in a greedy manner.

A further regularization strategy is the restriction of the size of the possible formulas to maintain interpretability. Thus, we choose small formula selection networks.

11.2 Learning in the minterm family

We approach structure learning as constrained parameter estimation in the naive exponential family (see Example 6), which coincides with the minterm family $\mathcal{F}_\wedge$ and has statistic $\phi = \delta\left[X_{[d]}, L_{[d]}\right]$.

By Theorem 37 all positive distributions are member of the minterm markov logic network. This expressivity result was generalized to arbitrary distributions, when allowing for formulas as basemeasures by Theorem 46.

Finding the distribution maximizing the likelihood of data would then be the empirical distribution. In this case we would have $\mu_D\left[L_{[d]} = x_{[d]}\right] = \mathbb{P}^D\left[X_{[d]} = x_{[d]}\right]$ and the maximum likelihood distribution is found by the problem

$$\operatorname{argmax}_{\theta \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}} \left\langle \left\{\theta, \mathbb{P}^D\right\} \right\rangle [\varnothing] - A(\theta)$$

which is solved at $\theta = \ln\left[\mathbb{P}^D\right]$ with $\mathbb{P}^{(\delta, \ln\left[\mathbb{P}^D\right])} = \mathbb{P}^D$. This follows from $\mathcal{L}_D\left(\mathbb{P}^{(\delta,\theta)}\right) = \mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^D || \mathbb{P}^{(\delta,\theta)}\right]$, which is by Gibbs inequality minimized at $\mathbb{P}^{(\delta,\theta)} = \mathbb{P}^D$, which is the case for $\theta = \ln\left[\mathbb{P}^D\right]$.

We here allow for $\ln[0] = -\infty$, with the convention of $\exp[-\infty] = 0$, to handle datasets where specific worlds are not represented. Better: Use Theorem 46 with basemeasure dropping non appearing data.

To avoid this overfitting situation, we regularize by restricting the parameter to be a set $\Theta \subset \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ and state

$$\operatorname{argmax}_{\theta\in\Theta} \left\langle \left\{\theta, \mathbb{P}^D\right\}\right\rangle [\varnothing] - A(\theta)\,. \tag{$\mathrm{P}_{\Theta,\mathbb{P}^D}$}$$

When the $\mathcal{F}$ is known we take Θ as the linear hull

$$\Theta^{\mathcal{F}} = \operatorname{span}(\{f : f \in \mathcal{F}\})\,.$$

Then, the problem is the parameter estimation problem studied in Chapter 10. To see this, we reparametrize by the coefficient vectors of the elements in the span, which are then understood as the canonical parameter of a

The problem of structure learning arises, when the set of formulas is to be learned, i.e.

$$\Theta^{\mathcal{H}} = \bigcup_{\mathcal{F}\in\mathcal{H}} \Theta^{\mathcal{F}}\,.$$

In this case, the problem in general fails to be convex.

11.3 Greedy Learning

It can be impracticle to learn all formulas at once, since the set $\mathcal{H}$ often grows combinatorically, for example when choosing as a powerset of formulas. 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. We in this section assume a current model $\tilde{\mathbb{P}}$, which is a generic positive distribution not necessarily a Markov Logic Network.

Alternative discussion: Can use current distribution as base measure and apply moment matching as first order condition.

11.3.1 Gradient heuristic and the proposal distribution

We now derive a heuristic of choosing features based on a variation of parameter vectors in naive exponential families.

For the naive exponential family

$$\mathcal{L}_D\left(\mathbb{P}^{(\mathrm{Id},\tilde{\theta})}\right) = \frac{1}{m}\sum_{j\in[m]} \ln\left[\mathbb{P}^{(\mathrm{Id},\tilde{\theta})}[X_{[d]} = D(j)]\right] = \left\langle \mathbb{P}^D, \gamma^{\mathrm{Id}}, \tilde{\theta}\right\rangle [\varnothing] - \ln\left[\left\langle \exp\left[\left\langle \gamma^{\mathrm{Id}}, \tilde{\theta}\right\rangle [X_{[d]}]\right]\right\rangle [\varnothing]\right]$$

we have the gradient

$$\frac{\partial \mathcal{L}_D\left(\mathbb{P}^{(\mathrm{Id},\tilde{\theta})}\right)}{\partial\tilde{\theta}} = \mathbb{P}^D - \mathbb{P}^{(\mathrm{Id},\tilde{\theta})}\,.$$

Given a feature $f[X_{[d]}]$ we vary the naive parameters by a function on $\theta \in \mathbb{R}$ by

$$\mathrm{Id}(\theta) = \mathrm{Id}(0) + \theta \cdot f$$

and get a likelihood gradient of

$$\begin{aligned}\frac{\partial \mathcal{L}_D\left(\mathbb{P}^{(\mathrm{Id}(\theta),\tilde{\theta})}\right)}{\partial\theta} &= \left\langle \left\{ \frac{\partial \mathcal{L}_D\left(\mathbb{P}^{(\mathrm{Id},\tilde{\theta})}\right)}{\partial\tilde{\theta}}|_{\tilde{\theta}(0)}, \frac{\partial\tilde{\theta}(\theta)}{\partial\theta}\right\}\right\rangle [\varnothing] \\ &= \left\langle \mathbb{P}^D, f\right\rangle [\varnothing] - \left\langle \mathbb{P}^{(\mathrm{Id},\tilde{\theta})}, f\right\rangle [\varnothing]\,.\end{aligned}$$

The gradient shows the typical decomposition into a positive and a negative phase. While the positive phase comes from the data term and prefers directions of large data support, the negative phase originates in the partition function and draws the gradient away from directions already supported by the current model $\mathbb{P}^{(\mathrm{Id},\tilde{\theta})}$. 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.

We can calulate all these feature gradients with the usage of a selection network. Given a set $\mathcal{H}$ of features we vary Id as

$$\mathrm{Id}(\theta) = \mathrm{Id}(0) + \left\langle \left\{ \theta, \gamma^{\mathcal{H}} \right\} \right\rangle \left[X_{[d]} \right] .$$

Now the gradient of the loss along the variation is

$$\begin{aligned} \frac{\partial \mathcal{L}_D \left(\mathbb{P}^{(\mathrm{Id}(\theta), \tilde{\theta})} \right)}{\partial \theta} &= \left\langle \left\{ \frac{\partial \mathcal{L}_D \left(\mathbb{P}^{(\mathrm{Id}, \tilde{\theta})} \right)}{\partial \tilde{\theta}} |_{\tilde{\theta}(0)}, \frac{\partial \tilde{\theta}(\theta)}{\partial \theta} \right\} \right\rangle [\varnothing] \\ &= \left\langle \left\{ \mathbb{P}^D, \gamma^{\phi} \right\} \right\rangle [\varnothing] \, L - \left\langle \left\{ \mathbb{P}^{(\mathrm{Id}, \tilde{\theta})}, \gamma^{\phi} \right\} \right\rangle [\varnothing] \, L \, . \end{aligned}$$

We want to choose the formula, which is best aligned with the gradient of the log-likelihood, that is using a formula selecting map $\mathcal{H}$

$$\mathrm{argmax}_{l \in [p]} \left\langle \left\{ \mathbb{P}^D, \gamma^{\mathcal{H}} \right\} \right\rangle [\varnothing] \, L = l - \left\langle \left\{ \mathbb{P}^{(\mathrm{Id}, \tilde{\theta})}, \gamma^{\mathcal{H}} \right\} \right\rangle [\varnothing] \, L = l \, .$$

This method is known as grafting.

When learning Markov Logic Networks, our features are propositional formulas. Given a set $\mathcal{F}$ of formulas we search for the best alignment

$$\mathrm{argmax}_{f \in \mathcal{F}} \left\langle \left\{ f, \mathbb{P}^D - \tilde{\mathbb{P}} \right\} \right\rangle [\varnothing] \tag{$\mathrm{P}_{\mathcal{F}, \mathbb{P}^D, \tilde{\mathbb{P}}}$}$$

We can choose selection architectures to efficiently parametrize

To take the steepest ascent formula among the formulas selectable by $\mathcal{F}$ we thus search for the maximal coordinate in the tensor network $\frac{\partial \mathcal{L}_D(\theta)}{\partial \theta}|_{\theta=0}$, that is solve

$$\mathrm{argmax}_{l_0 \dots l_{n-1}} \left\langle \left\{ \frac{\partial \mathcal{L}_D(\theta)}{\partial \theta}|_{\theta=0}, e_{l_0 \dots l_{n-1}} \right\} \right\rangle [\varnothing] \tag{40}$$

This is thus equivalent to the problem $\mathrm{P}_{\mathcal{F}, \mathbb{P}^D, \tilde{\mathbb{P}}}$, when taking all formulas selectable by $\mathcal{F}$ as the hypothesis Γ.

Let us now understand the likelihood gradient as the energy tensor of a probability distribution, which we call the proposal distribution.

Definition 52 (Proposal Distribution)**.** *Let there be a base distribution $\tilde{\mathbb{P}}$, a targeted distribution $\mathbb{P}^D$ and a formula selecting map $\mathcal{H}[X_{[d]}, L]$. The proposal distribution at inverse temperature $\beta > 0$ is the distribution of L defined by*

$$\left\langle \exp \left[\left\langle \left\{ \beta \cdot (\mathbb{P}^D - \tilde{\mathbb{P}}), \mathcal{H} \right\} \right\rangle [L] \right] \right\rangle [L | \varnothing] \, .$$

The proposal distribution is the member of the exponential family with statistics $\mathcal{H}$ and parameter $\beta \cdot (\mathbb{P}^D - \tilde{\mathbb{P}})$.

The proposal distribution is in the exponential family with sufficient statistic by the formula selecting map $\mathcal{H}$, namely the member with the canonical parameters $\theta = \mathbb{P}^D - \tilde{\mathbb{P}}$. Of further interest are tempered proposal distributions, which are in the same exponential family with canonical parameters $\beta \cdot (\mathbb{P}^D - \tilde{\mathbb{P}})$ where $\beta > 0$ is the inverse temperature parameter.

As Markov Logic Networks, the proposal distributions are in exponential families with the sufficient statistic defined in terms of formula selecting maps. While Markov Logic Networks contract the maps on the selection variables L, the proposal distributions contract them along the categorical variables X to define energy tensors.

We can then understand finding the largest coordinate as a search for the mode of this proposal distribution.

Theorem 51. *Grafting is the MAP query on the proposal distribution to any $\beta > 0$.*

To solve grafting, we thus need to answer a MAP query, for which we can apply the methods introduced in Chapter 4, such as Gibbs Sampling or Mean Field Approximations in combination with annealing.

11.3.2 Gain Heuristic

In the gain heuristic, only the parameters of the new formula are optimized and the others left unchanged. This is providing upper bounds on the likelihood increase.

More precisely this is Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ in the case

$$\Theta = \ln\left[\tilde{\mathbb{P}}\right] + \cup_{f\in\mathcal{F}}\mathrm{span}(f)\,.$$

Let us choose a formula $f \in \mathcal{F}$ and consider Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ in the case

$$\Theta^f = \ln\left[\tilde{\mathbb{P}}\right] + \mathrm{span}(f)\,.$$

This is parameter estimation on the exponential family with the single feature f and the base measure $\tilde{\mathbb{P}}$. Therefore we can apply the theory of Chapter 4 and characterize the solution by the θ satisfying the moment matching condition

$$\left\langle \tilde{\mathbb{P}}, \langle \exp[\theta]\rangle \left[X_{[d]}|\varnothing\right]\right\rangle [\varnothing] = \left\langle \mathbb{P}^D, f\right\rangle [\varnothing]\ .$$

We state the solution of this condition in the next theorem.

Lemma 14. *Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ in the case*

$$\Theta^f = \ln\left[\tilde{\mathbb{P}}\right] + \mathrm{span}(f)$$

is solved at

$$\hat{\theta_f} = \ln\left[\frac{Z_1}{Z_0}\frac{\mu_f^D}{1-\mu_f^D}\right]$$

where

$$Z_0 = \left\langle \left\{\tilde{\mathbb{P}}, \rho^f\right\}\right\rangle [X_f = 0] \quad , \quad Z_1 = \left\langle \left\{\tilde{\mathbb{P}}, \rho^f\right\}\right\rangle [X_f = 1] \quad \textit{and} \quad \mu_f^D = \left\langle \mathbb{P}^D, f\right\rangle [\varnothing]\ .$$

Proof. Follows from Lemma 13. □

We can apply this

Lemma 15. *Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ in the case*

$$\Theta = \ln\left[\tilde{\mathbb{P}}\right] + \cup_{f\in\mathcal{F}}\mathrm{span}(f)$$

is solved at any

$$\hat{\theta} = \tilde{\mathbb{P}} + \theta_{\hat{f}} \cdot \hat{f}$$

where the formula $\hat{f}$ is in

$$\hat{f} \in \mathrm{argmax}_{f\in\mathcal{F}}\mathrm{D}_{\mathrm{KL}}\left[\left\langle\left\{\mathbb{P}^D, f\right\}\right\rangle [\varnothing] \,||\, \left\langle\left\{\tilde{\mathbb{P}}, f\right\}\right\rangle [\varnothing]\right]$$

and $\theta_{\hat{f}}$ is the weight of $\hat{f}$ in the solution of Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ with $\Gamma = \tilde{\mathbb{P}} + \mathrm{span}(f)$. Here we denote by $\mathrm{D}_{\mathrm{KL}}[p_1||p_2]$ the Kullback-Leibler divergence between Bernoulli distributions with parameters $p_1, p_2 \in [0,1]$, that is

$$\mathrm{D}_{\mathrm{KL}}[p_1||p_2] = p_1 \cdot \ln\left[\frac{p_1}{p_2}\right] + (1-p_1)\cdot\ln\left[\frac{(1-p_1)}{(1-p_2)}\right]$$

Proof. By plugging in the above Lemma we get, that the KL divergence is the improvement of the objective to each f. The Problem is solved at the best improvement, where the KL divergence is maximal. □

One therefore takes the formula, which marginal distribution in the current model and the targeted distribution are differing at most, measured in the KL divergence.

One optimization method would thus be the computation of the mean parameters to both distribution, building the coordinatewise KL divergence and choosing the maximum. Since we need to evaluate each coordinate, this can be intractable for large sets of formulas.

Further improvement of the model can be achieved by iteratively optimizing the other weights as well, since their corresponding moment matching conditions might be violated after the integration of a new formula. This would require the computation of backward mappings for each candidate formula, for which we only have an alternating approach in general.

11.3.3 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 8, which is a greedy algorithm adding iteratively the currently best feature.

Algorithm 8 Greedy Structure Learning

Initialize

$$\tilde{\mathbb{P}} \leftarrow \mathbb{I}\left[X_{[d]}\right] \quad , \quad \mathcal{F} = \varnothing$$

while Stopping criterion is not met **do**

Structure Learning: Compute a (approximative) solution $\hat{f}$ to Problem $\mathrm{P}_{\Theta,\mathbb{P}^D}$ and add the formula to $\mathcal{F}_t$, i.e.

$$\mathcal{F}_t \leftarrow \mathcal{F}_{t-1} \cup \{\hat{f}\}$$

Weight Estimation: Estimate the best weights for the added formula and recalibrate the weights of the previous formulas, by calling Algorithm 7.

$$\tilde{\mathbb{P}} \leftarrow \mathbb{P}^{\mathcal{F}_t,\theta^t}$$

end while

When having used the same learning architecture multiple times, the energy of the corresponding formulas are all representable by a formula selecting architecture. Their energy term is therefore a contraction of the selecting tensor with a parameter tensor θ in a basis CP decomposition with rank by the number of learned formulas. When mutiple selection architectures have been used, the energy is a sum of such contractions. Let us note, that this representation is useful after learning, when performing energy-based inference algorithms on the result. During learning, one needs to instantiate the proposal distribution, which requires instantiation of the probability tensor. However, one could alternate data energy-based and use this as a particle-based proxy for the probability tensor.

12 Probabilistic Success Guarantees

When drawing data independently from a random distribution, we in this chapter derive guarantees, that the

12.1 Fluctuations

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.

12.1.1 Fluctuation of the empirical distribution

When drawing random states $D(j) \in \bigtimes_{k\in[d]}[m_k]$ by a distribution $\mathbb{P}^*$, we use the one-hot encoding to forward each random state to the random tensor

$$e_{D(j)}\left[X_{[d]}\right] .$$

The expectation of this random tensor is

$$\mathbb{E}\left[e_{D(j)}\right] = \sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[m_k]} \mathbb{P}^*[X_{[d]} = x_{[d]}] e_{x_0,\ldots,x_{d-1}}\left[X_{[d]}\right] = \mathbb{P}^*[X_{[d]}] .$$

The empirical distribution is then the average of independent random one-hot encodings, namely the random tensor

$$\mathbb{P}^D = \frac{1}{m}\sum_{j\in[m]} e_{D(j)} .$$

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

When the marginal of each datapoint is $\mathbb{P}^*$, the expectation of the empirical distribution is

$$\mathbb{E}\left[\mathbb{P}^D\right] = \frac{1}{m}\sum_{j\in[m]} \mathbb{E}\left[e_{D(j)}\right] = \mathbb{P}^* .$$

From the law of large numbers it follows, that in the limit of $m \to \infty$ at any coordinate $x \in \bigtimes_{k\in[d]}[m_k]$ almost everywhere

$$\mathbb{P}^D[X_{[d]} = x_{[d]}] \to \mathbb{E}\left[\mathbb{P}^D[X_{[d]} = x_{[d]}]\right] = \mathbb{P}^*[X_{[d]} = x_{[d]}] \,.$$

At finite m the empirical distribution differs from the by the difference

$$\mathbb{P}^D - \mathbb{P}^*$$

which we call a fluctuation tensor.

12.1.2 Fluctuation tensors and their widths

Let us now investigate random tensors, which result from the forwarding of the fluctuation of the empirical distribution by sufficient statistics.

Definition 53. *Given a statistic ϕ, $m \in \mathbb{N}$ and and a dataset we define the fluctuation tensor as the random tensor*

$$\eta^{\phi,\,\mathbb{P}^*,D} = \left\langle\left\{\mathbb{P}^D - \mathbb{P}^*, \phi\right\}\right\rangle[L]$$

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 naive 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.

12.1.3 Naive Exponential Family

This is the minterm exponential family!

In case of the naive exponential family, we have $\phi = \delta\left[X_{[d]}, L\right]$ and the fluctuation tensor is

$$\eta^{\delta,\,\mathbb{P}^*,D} = \mathbb{P}^D - \mathbb{P}^* \,.$$

This fluctuation tensor is related to tensor encodings of multinomial distributions, which we now define as multinomial random tensors.

Definition 54. *A multinomial random tensor is the sum of the one-hot encodings of independent random variables Z_j each distributed by $\mathbb{P}$*

$$Z^{\mathbb{P},m} = \sum_{j\in[m]} e_{Z_j} \,.$$

Lemma 16. *The fluctuation $\mathbb{P}^D - \mathbb{P}^*$ is a by $\frac{1}{m}$ rescaled centered multinomial random tensor with parameters $\mathbb{P}^*$ and m.*

Proof. By the above construction we have

$$\mathbb{P}^D - \mathbb{P}^* = \frac{1}{m}\left(e_{D(j)}\left[X_{[d]}\right] - \mathbb{E}\left[e_{D(j)}\left[X_{[d]}\right]\right]\right) \,.$$

□

12.1.4 Mean parameter in Markov Logic Networks

The mean parameter of the M-projection of the empirical distribution on the family of Markov Logic Networks with statistic $\mathcal{H}$ is the random tensor

$$\mu^D = \left\langle\left\{\mathcal{H}, \mathbb{P}^D\right\}\right\rangle[L] \,.$$

The expectation of this random tensor is

$$\mathbb{E}\left[\mu^D\right] = \left\langle\left\{\mathcal{H}, \mathbb{E}\left[\mathbb{P}^D\right]\right\}\right\rangle[L] = \left\langle\left\{\mathcal{H}, \mathbb{P}^*\right\}\right\rangle[L] = \mu^* \,,$$

where we used that the expectation and contraction operation can be commuted due to the multilinearity of contractions.

The fluctuation of this mean parameter is

$$\mu^D - \mathbb{E}\left[\mu^D\right] = \left\langle\left\{\mathcal{H}, \mathbb{P}^D - \mathbb{P}^*\right\}\right\rangle[L] \,.$$

We notice, that this is the fluctuation tensor $\eta^{\mathcal{H},\mathbb{P}^*,D}$.

12.1.5 Energy tensor in proposal distributions

The fluctuation tensor appears as a fluctuation of the energy of the proposal distribution. For the expected energy it holds that

$$\mathbb{E}\left[E^{\mathcal{H}^T,\mathbb{P}^D-\tilde{\mathbb{P}}}\right] = \mathbb{E}\left[\left\langle\left\{\mathcal{H}^T,\mathbb{P}^D-\tilde{\mathbb{P}}\right\}\right\rangle[L]\right] = \left\langle\left\{\mathcal{H}^T,\mathbb{E}\left[\mathbb{P}^D-\tilde{\mathbb{P}}\right]\right\}\right\rangle[L] = \left\langle\left\{\mathcal{H}^T,\mathbb{P}^*-\tilde{\mathbb{P}}\right\}\right\rangle[L] = \mathbb{E}\left[E^{\mathcal{H}^T,\mathbb{P}^*-\tilde{\mathbb{P}}}\right] .$$

The fluctuation of this random tensor is

$$\mathbb{E}\left[E^{\mathcal{H}^T,\mathbb{P}^D-\tilde{\mathbb{P}}}\right] - \mathbb{E}\left[E^{\mathcal{H}^T,\mathbb{P}^*-\tilde{\mathbb{P}}}\right] = \mathbb{E}\left[E^{\mathcal{H}^T,\mathbb{P}^D-\mathbb{P}^*}\right]$$

and coincides with $\eta^{\mathcal{H}^T,\mathbb{P}^*,D}$.

12.1.6 Binary Features

In all the above example we have statistics consistent of binary features. In this case the marginal distributions of the coordinates of $\eta^{\phi,\,\mathbb{P}^*,D}$ are scaled and centered binomials, which we investigate now.

Lemma 17. *Let ϕ be a statistic and let for $l \in [p]$ ϕ_l be a binary feature, i.e. let $\mathrm{im}\,(\phi_l) \subset \{0,1\}$. Then, the marginal distribution of the coordinate $\eta^{\phi,\,\mathbb{P}^*,D}[L=l]$ is*

$$\frac{1}{m}\left(B\left(p^{(l)},m\right)-p^{(l)}\right) ,$$

where by $B\left(p^{(l)},m\right)$ we denote the binomial distribution with mean parameter

$$p^{(l)} = \langle\{\phi_l,\mathbb{P}^*\}\rangle\,[\varnothing] .$$

Proof. We notice that when forwarding a random sample $D(j)$ of $\mathbb{P}^*$ is the random tensor

$$e_{D(j)}\left[X_{[d]}\right]$$

and since $\mathrm{im}\,(\phi_l) \subset \{0,1\}$ the contraction

$$\left\langle\left\{\phi_l, e_{D(j)}\left[X_{[d]}\right]\right\}\right\rangle[\varnothing]$$

is a random variable taking values in $\{0,1\}$. The variable therefore follows a Bernoulli distribution with mean parameter

$$p^{(l)} = \mathbb{E}\left[\left\langle\left\{\phi_l, e_{D(j)}\left[X_{[d]}\right]\right\}\right\rangle[\varnothing]\right] = \langle\{\phi_l,\mathbb{P}^*\}\rangle\,[\varnothing] .$$

□

12.1.7 Widths of random tensors

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 55. *Given a set $\Gamma \subset \bigotimes_{k\in[d]}\mathbb{R}^{m_k}$ and η a random tensor taking values in $\bigotimes_{k\in[d]}\mathbb{R}^{m_k}$ we define the width as the random variable*

$$\omega_\Gamma(\eta) = \sup_{\theta\in\Gamma}\left|\langle\{\theta,\eta\}\rangle\,[\varnothing]\right| .$$

12.2 Expected and Empirical Risk Minimization

We take a frequentist approach here and study the distributions of estimated parameters depending on random data.

12.2.1 Maximum Likelihood Estimation on random data

We here investigate the statistical errors of the maximum likelihood estimators. The empirical distribution is understood as an estimation of an underlying distribution. When sampling by independent copies of the underlying distribution, its mean is the underlying distribution. However, due to fluctuations around this mean the solution of estimation problems with respect to the empirical distribution does in general not coincide with the solution given the underlying distribution.

To be more precise, when generating data $\{D(j)\}_{j\in[m]}$ by independent copies of a distribution $\mathbb{P}^*$ we define the minimization problems

$$\hat{\theta} = \text{argmin}_{\theta\in\Gamma}\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^\theta\right] \tag{$P_{\{D(j)\}_{j\in[m]},\Gamma}$}$$

and

$$\theta^* = \text{argmin}_{\theta\in\Gamma}\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^\theta\right] \tag{$P_{\mathbb{E},\Gamma}$}$$

where by $\mathbb{P}^\theta$ we denote the element of an exponential family with sufficient statistics ϕ and parameter θ.

We have at each hypothesis $\theta \in \Gamma$

$$\mathbb{E}\left[\mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^\theta\right]\right] = \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^\theta\right] \tag{41}$$

and thus, the objective in Problem $P_{\{D(j)\}_{j\in[m]},\Gamma}$ converges in the limit $m \to \infty$ by the law of large numbers at each hypothesis $\theta \in \Gamma$ to the objective in Problem $P_{\mathbb{E},\Gamma}$.

To use this insight in the derivation of bounds of the distance of $\hat{\theta}$ and θ^*, we need to quantifying this convergence

- Non-asymptotically: Since we typically have access to limited amounts of data, that is finite m, we need to quantify the concentration in non-asymptotic cases.
- Uniform: Since the problems are optimized at extreme situations, the convergence of the objective has to happen uniformally at multiply $\theta \in \Gamma$.

12.2.2 Solution of the Expected Problem

Let us first investigate the solution of Problem $P_{\mathbb{E},\Gamma}$, to get a reference for Problem $P_{\{D(j)\}_{j\in[m]},\Gamma}$.

The solution does not change when manipulating the objective as

$$\text{argmin}_{\theta\in\Gamma}\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^\theta\right] = \text{argmin}_{\theta\in\Gamma}\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^\theta\right] - \mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^*\right] \tag{42}$$

We notice that the objective of the right side problem is the Kullback-Leibler divergence

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^* || \mathbb{P}^\theta\right] \tag{43}$$

Thus, Problem $P_{\mathbb{E},\Gamma}$ coincides with the approximation of $\mathbb{P}^*$ by an element $\mathbb{P}^\theta$ with $\theta \in \Gamma$ in the Kullback-Leibler divergence.

We decompose:

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{\theta^*} || \mathbb{P}^\theta\right] = \log\left[\frac{\mathcal{Z}(\theta)}{\mathcal{Z}(\theta^*)}\right] + \frac{1}{\mathcal{Z}(\theta^*)}\left\langle \theta^* - \theta, \exp\left[\theta^*\right]\right\rangle \tag{44}$$

Some insights can be drawn based on this decomposition:

- The first term vanishes, when both partition functions are the same. We can always adjust θ by constant offsets on all coordinates (of course without changing the distribution), such that the partition functions are equal. This is done by the map
$$\theta \to \theta + \lambda \cdot \mathbb{I}$$
where we choose $\lambda \in \mathbb{R}$ by
$$\lambda = \frac{\mathcal{Z}(\theta^*)}{\mathcal{Z}(\theta)}$$
to ensure a vanishing first term.
- In the second term the differences in coordinates are weighted by exponentiated solution θ^*. Where the probability mass is small errors in θ have a small influence on the Kullback-Leibler divergence.

Theorem 52. *Let us assume $\mathbb{P}^* = \mathbb{P}^{\theta^*}$ for some $\theta^* \in \Gamma$. Then one solution of Problem $P_{\mathbb{E},\Gamma}$ coincides with θ^* up to constant offsets.*

Proof. This follows directly from the Gibbs inequality, since

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{\theta^*} || \mathbb{P}^\theta\right] \leq 0 \tag{45}$$

with equality if only if $\mathbb{P}^{\theta^*}$ and $\mathbb{P}^\theta$ are equal. The uniqueness follows from Theorem **??** □

12.2.3 Recovery Guarantee based on Widths

Theorem 53. *Let us assume $\theta^* \in \Gamma$ and that $\mathcal{Z}(\theta)$ is constant among $\theta \in \Gamma$. Then for any solution $\hat{\theta}$ of the empirical problem we have*

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{\theta^*}||\mathbb{P}^{\hat{\theta}}\right] \leq \omega_\Gamma(\eta)\,. \tag{46}$$

Proof. First we notice

$$\mathrm{argmin}_{\theta\in\Gamma}\mathcal{L}\theta = \mathrm{argmin}_{\theta\in\Gamma}\mathcal{L}\theta - \mathcal{L}\theta^* \tag{47}$$

When $\theta^* \in \Gamma$ the minimum of the empirical loss with respect to θ^* is negative since

$$\mathcal{L}\hat{\theta} - \mathcal{L}\theta^* \leq \mathcal{L}\theta^* - \mathcal{L}\theta^* \leq 0 \tag{48}$$

We separate expectations and fluctuations and get

$$\mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{\theta^*}||\mathbb{P}^{\hat{\theta}}\right] \leq \mathrm{D}_{\mathrm{KL}}\left[\mathbb{P}^{\theta^*}||\mathbb{P}^{\hat{\theta}}\right] - (\mathcal{L}\hat{\theta} - \mathcal{L}\theta^*) \leq \omega_\Gamma\,. \tag{49}$$

□

The supremum of the differences between expected and empirical risks is the width of the fluctuation tensor, as we state next.

Lemma 18. *For any Γ and D we have*

$$\omega_\Gamma(\eta^{\delta,\mathbb{P}^*,D}) = \sup_{\theta\in\Gamma}\mathbb{H}\left[\mathbb{P}^D,\mathbb{P}^\theta\right] - \mathbb{H}\left[\mathbb{P}^*,\mathbb{P}^\theta\right]$$

Proof. Using the decomposition of cross entropy in the naive exponential family

$$\mathbb{H}\left[\mathbb{P}^D,\mathbb{P}^\theta\right] = \langle\mathbb{P},\ln\left[\mathbb{P}^*\right]\rangle\left[-\right]A(\ln\left[\mathbb{P}^*\right])\,.$$

□

Corollary 8. *At the solution $\hat{\theta}$ of Problem $P_{\{D(j)\}_{j\in[m]},\Gamma}$ we have*

$$\mathbb{H}\left[\mathbb{P}^D,\mathbb{P}^{\hat{\theta}}\right] - \mathbb{H}\left[\mathbb{P}^*,\mathbb{P}^{\hat{\theta}}\right] \leq \omega_\Gamma(\eta^{\delta,\mathbb{P}^*,D})\,.$$

12.3 Guarantees for Mode of the Proposal Distribution

Let us now derive probabilistic guarantees, that the mode of the proposal distribution at the empirical and the generating distribution are equal.

Definition 56. *The max gap of a tensor V is the quantity*

$$\delta(V) = \min_{i\neq i^{max}} V[X=i^{max}] - V[X=i]$$

where

$$i^{max} \in \mathrm{argmax}_i V[X=i]\,.$$

If multiple maxima exist, the

Theorem 54. *Whenever the energy tensor of the expected proposal distribution has a gap of δ, then for every $\epsilon > 0$ the mode of the expected proposal distribution coincides with the empirical proposal distribution with probability at least $1 - \exp\left[-\frac{1}{\epsilon^2}\right]$, provided that*

$$m > C\frac{\left(\sum_{k\in[d]}\ln\left[m_k\right]\right)}{\delta^2}$$

where C is a universal constant.

To proof the theorem we first use a deterministic recovery guarantee involving the width of the fluctuation tensor and then apply the width bound of Theorem 105.

Lemma 19. *Whenever*

$$\delta(\langle\{\mathbb{P}^*, \mathcal{H}\}\rangle [L]) > 2 \cdot \omega_{\{e_{x_0,\ldots,x_{d-1}}:x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[m_k]\}}(\eta^{\mathcal{H},\mathbb{P}^*,D}),$$

then the mode of the proposal distribution to the empirical distribution coincides with the mode of the proposal distribution to the generating distribution.

Proof. If different, then the expected objective at the solutions of the empirical and expected is at least δ. But at the empirical, the difference it as most twice the width different, so that is a contradiction to the assumption. □

Proof of Theorem 54. Given the assumed bound, the sub-gaussian norm of the width is upper bounded by $C_2 \cdot \delta$, thus for any $\epsilon > 0$ we have

$$\omega_{\{e_{x_0,\ldots,x_{d-1}}:x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[m_k]\}}(\eta^{\mathcal{H},\mathbb{P}^*,D}) < 2\delta$$

with probability at least $1 - \exp\left[-\frac{1}{\epsilon^2}\right]$. The claim thus follows with Lemma 19. □

Example 12 (Gap of a MLNs with single formulas)**.** *Let there be the MLN of a maxterm f with d variables, and let $\mathcal{F}$ be the maxterm selecting tensor, then*

$$\delta(E^{(\mathcal{F},\mathbb{P}^{(\{f\},\theta_f)}-\langle\mathbb{I}\rangle[X_{[d]}|\varnothing])}) = \frac{1}{2^d - 1 + \exp[-\theta_f]}$$

If $\theta_f > 0$ we have an exponentially small gap. Thus, for the above Lemma to apply, the width needs to be exponentially in d small.

Let there be the MLN of a minterm f with d variables, then

$$\delta(E^{(\mathcal{F},\mathbb{P}^{(\{f\},\theta_f)}-\langle\mathbb{I}\rangle[X_{[d]}|\varnothing])}) = \frac{1}{1 + (2^d - 1) \cdot \exp[-\theta_f]}$$

For large θ_f and d, the gap tends to 1.

12.4 Guarantees for Parameter Estimation

This is mean parameter fluctuation interpretation of the random tensor.

Lemma 20. *For any $\mathcal{H}$ and D drawn from $\mathbb{P}^*$ we have*

$$\left\|\mu^D - \mu^*\right\|_2 = \omega_{\mathcal{S}}(\eta^{\mathcal{H},\mathbb{P}^*,D}),$$

where $\mu^D = \left\langle\left\{\mathcal{H}, \mathbb{P}^D\right\}\right\rangle [L]$ and $\mu^* = \langle\{\mathcal{H}, \mathbb{P}^*\}\rangle [L]$.

We can thus apply the sphere bounds.

Theorem 55. *For any $\epsilon \in (0, 1)$ we have the following with probability at least $1 - \epsilon$. Let $\hat{\theta}$ and $\tau > 0$, then*

$$\left|\mathbb{H}\left[\mathbb{P}^*, \mathbb{P}^{\hat{\theta}}\right] - \mathbb{H}\left[\mathbb{P}^D, \mathbb{P}^{\hat{\theta}}\right]\right| \leq \tau \cdot \left\|\hat{\theta}\right\|_2$$

provided that

$$m \geq \frac{\langle\{\mu^*\}\rangle [\varnothing] - \left\langle\left\{(\mu^*)^2\right\}\right\rangle [\varnothing]}{\epsilon\tau^2}.$$

Proof. We have by Cauchy Schwartz

$$\left|\left\langle\left\{\mu^D - \mu^*, \hat{\theta}\right\}\right\rangle [\varnothing]\right| \leq \left\|\mu^D - \mu^*\right\|_2 \cdot \left\|\hat{\theta}\right\|_2$$

and with Lemma 20

$$\left|\left\langle\left\{\mu^D - \mu^*, \hat{\theta}\right\}\right\rangle [\varnothing]\right| \leq \omega_{\mathcal{S}}(\eta^{\mathcal{H},\mathbb{P}^*,D}) \cdot \left\|\hat{\theta}\right\|_2.$$

We show in Part III that in Theorem 106 that

$$\omega_{\mathcal{S}}(\eta^{\mathcal{H},\mathbb{P}^*,D}) \leq \tau$$

when m satisfies the assumed lower bound, from which the claim follows. □

13 Statistical Models of First Order Logic (FOL)

We now extend the tensor representation to structured representations, whereas the above was on factored representation of systems. The models/events in this situation are precise relations between objects.

13.1 World Tensors

We introduce here object enumerating variables O are valued in $[r]$, with $|A| = r$ and an enumeration of A by a_o.

Formulas in first order logic can contain variables, which are placeholder for specific individuals. Given a model and an assignment of objects to the arguments of a formula, the truth of the formula can be interpreted. This truth interpretation defines thus for any model a tensor, which we call the grounding tensor.

Let us assume, that we have a function-free theory with d predicates, where are predicates all of the same arity n. We then formalize a world in the following based on a selection variable L selecting a specific predicate and term variables $O_{[n]} = O_0, \ldots, O_{n-1}$ representing choices of objects from a given set A.

Definition 57 (FOL World)**.** *Given a set of objects $A = \{a_o : o \in [r]\}$ enumerated by $[r]$ and a finite set $\{g_0, \ldots, g_{d-1}\}$ of n-ary predicates a world is a Boolean tensor*

$$W[L, O_{[n]}] : [d] \times \left(\bigtimes_{l \in [n]} [r] \right) \to [2] \,. \tag{50}$$

We interpret the world tensor as encoding in the coordinate $W(k, o_0, \ldots, o_{n-1})$, whether the substitution of the variables in the kth predicate by the individuals $a_{o_0}, \ldots, a_{o_{n-1}}$ is satsifed on a corresponding world, that is

$$g_k(a_{o_0}, \ldots, a_{o_{n-1}}) = W(k, o_0, \ldots, o_{n-1}) \,.$$

When the assumptions are not met, we can do the following tricks. Functions are turned to predicates by their relation interpretation. If there are predicates of different arity in the theory, we can trivially extend them to $n$ary predicates by tensor products with the trivial tensor $\mathbb{I}$. This can be done by a tensor product with $e_0\,[O]$, where we add an object a_0 as a placeholder for predicates with smaller arity.

While in first order logics, depending on the chosen semantics, worlds can have infinite sets of objects, we here only treat worlds with finite objects.

13.1.1 Special Case of Propositional Logics

Propositional Logics is the case of $n = 0$. In that case, worlds are

$$W : [d] \to [2]$$

and represented by the tuples $x_k = W(k)$ as before.

Compared with propositional formulas, the grounding tensor does not take as input a specific world, but is defined on a given world. We show in this chapter, how both tensor interpretations can be transformed, i.e. by extracting samples from a FOL world W interpretated as an empirical distribution over PL worlds, and by generating FOL worlds by a set of samples generated from a PL distribution.

13.1.2 Enumeration of worlds

We here enumerate worlds coinciding in their domains. This is typical in database semantics, where only those worlds are considered which domains have a one-to-one map to the constant symbols appearing in a knowledge base. As above we use term variables O to represent objects in the domain A of a world.

Propositional worlds have been enumerated by indices of d Booleans, that is for a world $W : [d] \to [2]$ we take the index

$$x_0, \ldots, x_{d-1} \quad \text{where} \quad x_k = W(k) \,.$$

When we want to enumerate the first order logic worlds to a fixed set of objects A, we flatten the tensor W and get indices

$$\{x_{k,o_0,\ldots,o_{n-1}} : k \in [d], o_0, \ldots, o_{n-1} \in [r]\} \quad \text{where} \quad x_{k,o_0,\ldots,o_{n-1}} = W(k, o_0, \ldots, o_{n-1}) \,.$$

When restricting to worlds coinciding in their domain, we still have a factored respresentation of the system, since we can enumerate the possible worlds by a cartesian product. However, the number of categorical variables representing the world is $d \cdot r^n$ and tensor representations, even in sparse formats, are not feasible due to the large order required.

13.1.3 Random World Tensors

Random worlds with fixed domains A are probability distributions among possible tensors.

All possible world tensors given a set of objects A are enumerated by a Boolean to each coordinate of the tensor, indicating whether the corresponding assertion holds in the world. Random worlds are then probability distributions

$$\mathbb{P} : \bigtimes_{k\in[d],\, o_0,\ldots,o_{n-1}\in[r]} [2] \to \mathbb{R}\,.$$

We notice, that by definition these probability distributions are distributions of $d \cdot r^n$ Booleans with $2^{(d\cdot r^n)}$ many states. We will later in this chapter investigate methods to handle such high-dimensional distributions in the formalism of exponential families.

13.2 Formulas in first order logics

Given a FOL World W, arbitrary formulas in first order logics are interpreted in terms of the satisfactions of their groundings.

Definition 58 (Grounding of FOL Formulas)**.** *Given a specific world W, with domain A enumerated by $[r]$, the grounding of a formula q with n_q variables is the tensor*

$$q|_W[O_q] : \bigtimes_{l\in[n_q]} [r] \to [2]\,.$$

We interpret the coordinates as encoding, whether the substitution of the variables in the formula is satisfied given a world W, that is

$$q|_W\left(o_0,\ldots,o_{n_q}\right) = \begin{cases} 1 & \textit{if } q\left(a_{o_0},\ldots,a_{o_{n_q}}\right) = 1 \textit{ given the world } W \\ 0 & \textit{else} \end{cases}\,.$$

Slicing the grounding tensor of a FOL Formula amounts to substitution of the respective variable by the constant at the enumeration index.

The above definition of formulas can be interpreted as a map from worlds to grounding tensors

$$q : W \to q|_W\,.$$

When interpreting this map as a basis encoding, formulas are tensors in the tensor space

$$\left(\bigotimes_{k\in[d],o_0,\ldots,o_{n-1}\in[r]} \mathbb{R}^2\right) \otimes \left(\bigotimes_{k\in[d],o_0,\ldots,o_{n_q}\in[r]} \mathbb{R}^2\right)\,.$$

Atomic formulas are stored in the slices to the first axis of W and we have

$$g_k|_W = \left\langle \{W[L, O_{[n]}], e_k\,[L]\}\right\rangle \left[O_{[n]}\right]\,. \tag{51}$$

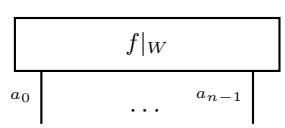


Figure 25: Relational encoding of the grounding tensor to a formula q with n arguments on world W.

13.2.1 Syntactical Decomposition of quantifier-free formulas

In order to have a sound semantic, the grounding of FOL formulas is determined by the syntax of the formula, i.e. a decomposition of the formula into connectives and quantifiers acting on atomic formulas.

Quantifier-free formulas are connectives acting on atomic formulas. We can describe them as in the case of propositional logics in the ρ-formalism. While the atomic formulas where delta tensors copying states, they are more involved here.

Theorem 56. *For any connective $\circ$ and formulas q_1 and q_2 we have*

$$(q_1 \circ q_2)|_W [O_{[n]}] = \left\langle \left\{ \rho^{q_1|_W} \left[O_{[n]}, X_{q_1}\right], \rho^{q_2|_W} \left[O_{[n]}, X_{q_2}\right], \rho^{\circ} \left[X_{q_1 \circ q_2}, X_{q_1}, X_{q_2}\right], e_1 \left[X_{q_1 \circ q_2}\right] \right\} \right\rangle \left[O_{[n]}\right] . \quad (52)$$

Proof. By the semantic interpretation of the groundings, which has to be sound. □

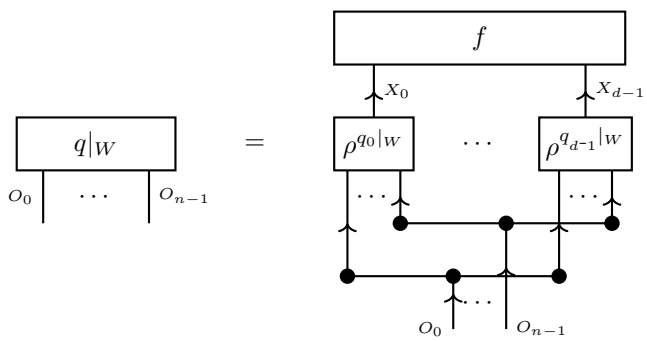
Here, variables can be shared by the connected formulas, therefore the variables in the combined formula are unions of the possible not disjoint variables of the connected formulas.

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 59. *Given a formula q in first order logic, we say that a propositional formula $f\left[X_{[d]}\right]$ is the propositional equivalent to q given atomic formulas q_k in first order logic, when for any world W we have*

$$q|_W \left[O_q\right] = \left\langle \left\{ \rho^{q_k|_W} \left[O_{q_k}, X_k\right] : k \in [d] \right\} \cup \left\{ f\left[X_{[d]}\right] \right\} \right\rangle \left[O_q\right] \quad (53)$$

We depict the relation of a grounding tensor to a propositional formula



13.2.2 Quantifiers

The variables quantified over are dropped i.e. $O_{\exists_{O_\exists} q} = O_q / O_\exists$.

Existential and universal quantifiers appear in generic first order logic. They are not representable as linear transform of the respective quantifier-free formula. Here we represent them as a compositions of contractions with the greater zero function ψ defined as

$$\psi : \mathbb{R} \to [2] \quad , \quad \psi\left(x\right) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & else \end{cases} \quad (54)$$

Theorem 57. *For any formula q with variables $O_{[n]}$ we have*

$$\left(\exists_{a_{o_0}} q|_W\right) \left[O_1, \ldots, O_{n-1}\right] = \psi\left(\left\langle \left\{ q|_W \right\} \right\rangle \left[O_1, \ldots, O_{n-1}\right]\right) \quad (55)$$

and

$$\left(\forall_{a_{o_0}} q|_W\right) \left[O_1, \ldots, O_{n-1}\right] = \psi\left(\left\langle \left\{ q|_W \right\} \right\rangle \left[O_1, \ldots, O_{n-1}\right] - r \cdot \mathbb{I}\left[O_1, \ldots, O_{n-1}\right]\right) . \quad (56)$$

Proof. We proof the claimed equalities to arbitrary slices of the remaining variables, which amount to arbitrary substitutions of the formulas. For any indices $o_1, \ldots, o_{n-1}$ we have

$$\left\langle \left\{ q|_W \right\} \right\rangle \left[O_1 = o_1, \ldots, O_{n-1} = o_{n-1}\right] = \sum_{o_0 \in [r_0]} q|_W \left[O_0 = o_0, \ldots, O_{n-1} = o_{n-1}\right] .$$

The existential quantification with respect to the zeroth variable is satified, when for at least one $o_0 \in [r_0]$ we have $q|_W \left[O_0 = o_0, \ldots, O_{n-1} = o_{n-1}\right] = 1$, which is equal to

$$\sum_{o_0 \in [r_0]} q|_W \left[O_0 = o_0, \ldots, O_{n-1} = o_{n-1}\right] \geq 1$$

and

$$\psi\left(\sum_{o_0\in[r_0]} q|_W[O_0=o_0,\ldots,O_{n-1}=o_{n-1}]\right)=1\,.$$

Combining both equations we get

$$\left(\exists_{a_{o_0}} q|_W\right)[O_1=o_1,\ldots,O_{n-1}=o_{n-1}]=\psi\left(\langle\{q|_W\}\rangle\,[O_1=o_1,\ldots,O_{n-1}=o_{n-1}]\right)$$

and arrive at the first claim.

To show the second claim, we notice that the universal quantification with respect to the zeroth variable is satisfied, when for all $o_0\in[r_0]$ we have $q|_W[O_0=o_0,\ldots,O_{n-1}=o_{n-1}]=1$. This is equal to

$$\sum_{o_0\in[r_0]} q|_W[O_0=o_0,\ldots,O_{n-1}=o_{n-1}]=r$$

and

$$\psi\left(\sum_{o_0\in[r_0]} q|_W[O_0=o_0,\ldots,O_{n-1}=o_{n-1}]-r\right)=1\,.$$

It follows that

$$\begin{aligned}&\left(\forall_{a_{o_0}} q|_W\right)[O_1=o_1,\ldots,O_{n-1}=o_{n-1}]\\&\quad=\psi\left(\langle\{q|_W\}\rangle\,[O_1=o_1,\ldots,O_{n-1}=o_{n-1}]-r\cdot\mathbb{I}\,[O_1=o_1,\ldots,O_{n-1}=o_{n-1}]\right)\end{aligned}$$

which establishes the second claim. □

13.2.3 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, that is

$$\ell_0\left(q|_W\right)\leq r^{n_q}\,.$$

In this case the basis CP decomposition (see Section 19.1.2) enables a representation of the grounding with significantly smaller storage demand, see Theorem 92. We depict the CP decomposition of a formula grounding in Figure 26.

Most logical syntaxes make use of the ℓ_0-sparsity and explicitly store only the known assertions. The not specified assertions have different interpretation based on the assumptions made. Under the Closed World Assumption all unspecified assertions are assumed to be false.

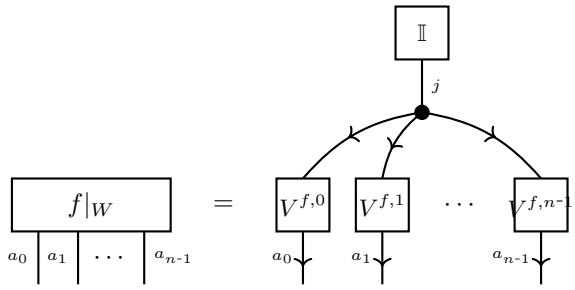


Figure 26: Basis CP Decomposition of the grounding of q.

13.2.4 Example: Queries

Queries are formulas p, which are asked against a specific world W. Each argument is of such formulas are called projection variables. The answer of a query is a list of solution mappings from the projection variables to objects in the world, such that the query formula is true.

Answering a query thus corresponds with finding the basis CP Decomposition (see Section 19.1.2).

There are situations, in which the CP Decomposition comes with unnecessary storage overhead compared with other formats. To be more precise, let us show how Join operations on groundings are represented by contractions.

Definition 60. *Let there be a world with individual sets J_v for $v \in \mathcal{V}$ and groundings $q_l|_W$*

$$q_l|_W : \bigtimes_{v \in \mathcal{V}_l} \to [2]$$

defined on subsets of the individuals sets $\bigtimes_{v \in \mathcal{V}} J_v$. Then their Join *is defined as*

$$\text{Join}\,(q_0, \ldots, q_{p-1})|_W = \prod_{l \in [p]} q_l|_W \,.$$

We can understand the Join of groundings by a factor graph, where each grounding is a factor connecting the variables decorating node sets $\mathcal{V}_l$.

Theorem 58 (Joins by Hadamard products)**.** *For any* Join *of grounded formulas we have*

$$\text{Join}\,(q_0, \ldots, q_{p-1})|_W = \langle \{q_0|_W, \ldots, q_{p-1}|_W\} \rangle\, [\mathcal{V}] \,.$$

Proof. Later by effective directed calculus. □

Thus, we can store the Join by at most linear in the number of joined formulas demand. Storing the result by solution maps on the other side can create storage overheads as we show next.

Theorem 59. *Given a grounding $q|_W$ on a world then the minimal rank of a basis CP Decomposition storing $q|_W$ is the number of tuples $\{a_v \,:\, v \in \mathcal{V}\}$ such that*

$$q|_W(\{a_v \,:\, v \in \mathcal{V}\}) = 1 \,.$$

Proof. Later in binary+directed Contraction Calculus, where we also define ℓ_0-norm □

Example 13. *For example, when a query with many basic graph patterns with pairwise different projection variables. The global CP Decomposition would come here with an exponential storage overhead compared with storage as a tensor product of CP Decompositions to each Basic graph pattern.*

Remark 26 (Distinguishing from probabilistic queries)**.** *Let us distinguish the discussion here from those of queries in probabilistic reasoning, which have two main differences. First, we ask queries against all possible pairs of variables, instead of asking the probability of satisfaction of a specific formula. Second, since we made the epistemologic assumption of knowing possibilities and not probabilities in logics, a query is answered by a truth value. We then only output in the shape of solution mappings the variable assignments where the query formula is true. Thus, the queries here can be thought of as a batch of probabilistic queries with Boolean answers. Probabilistic queries can furthermore be understood in terms of the data extraction process described in this section. We can ask the query in probabilistic form (decomposed into atomic formulas) on the resulting empirical distribution. This results in the ratio of the worlds satisfying the query among those worlds satisfying the extraction query p.*

13.3 Representation of Knowledge Graphs

Let us now represent a specific fragment of FOL, namely Description Logics which Knowledge Bases are often refered to as Knowledge Graphs.

13.3.1 Representation as unary and binary predicates

Atomic formulas in knowledge bases following the OWL standard (thus, especially knowledge graphs) are binary (owl:ObjectProperties) and unary (owl:Class). We enumerate the predicates by $[d]$, the objects in the domain A by $[r]$, and extend the unary predicates to binaries by tensor product with $e_0\,[O]$ (see discussion above) A Knowledge Graph on the set A of constants (owl:NamedIndividuals) is the tensor

$$kg : [d] \times [r] \times [r] \to [2] \,.$$

13.3.2 Representation as ternary predicate

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 kg with individuals A

$$\text{RDF}|_{kg} : [r] \times [r] \times [r] \to [2]$$

where

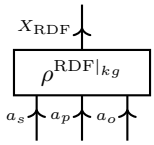
$$\mathrm{RDF}|_{kg}(o_s, o_p, o_o) = \begin{cases} 1 & \text{if triple } \langle o_s, o_p, o_o \rangle \text{ is in Knowledge Graph } kg \\ 0 & \text{else} \end{cases} .$$

Sparse representation of the grounding tensor to a knowledge graph is of central importance, as investigated in Bigerl et al. (2020). We here do basis CP for sparse representation. Slicing the tensor ρ^{kg} along the predicate axis retrieves specific information about roles and can be efficiently be performed on these formats.

The role rdf:typehas 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 rdf:typeas a placeholder for unitary formulas in a space of binary formulas.

Approximations of grounding tensors by decompositions leads to embeddings of the individuals such as Tucker, Complex, RESCAL (see Nickel et al. (2016)).

For our purposes of evaluating logical formulas such as SPARQL queries we use the relational encoding of the groundings, which are depicted by



13.3.3 SPARQL Queries

The SPARQL query language is a syntax to express first order logic formulas q. Given a specific world W, the execution of query is the interpretation $q|_W$, typical represented in a sparse basis CP format where each slice represents a solution mapping.

Triple Patterns Formulas which are of specific importance are triple patterns:

- Unary triple pattern with one variable, representing a formula with a single projection variable. For the example $\langle ?a_0, a, C \rangle$ see Figure 27a.
- Binary triple pattern with two variables, representing a formula with two projection variables. For the example $\langle ?a_0, R, ?a_1 \rangle$ see Figure 27b.

These triples are compositions of slicing operation, here seen as a map from single or two projection variables to the triples $(a_{o_s}, a_{o_p}, a_{o_o})$, with the RDF formula.

These slicing operations are performed by contractions with tensors $\psi^{\langle \dots \rangle}$, for unary triple patterns of the shape

$$\psi : ([d] \times [r] \times [r]) \times [r] \to [2]$$

and for binary

$$\psi : ([d] \times [r] \times [r]) \times [r] \times [r] \to [2]$$

defined by identity and basis vectors depending on the type of triple.

In the example $\langle ?a_0, a, C \rangle$ we have

$$\psi^{\langle ?a_0, a, C \rangle} = \delta^{a_s, ?a_0} \otimes e_0 \otimes \delta^{a_o, ?a_1}$$

The composition $\psi(\psi^T)$ of the matrification of the tensor ψ is a projection. That means that applying $\psi(\psi^T)$ is the same map as applying once.

Basic Graph Patterns Generic SPARQL queries are compositions of triple patterns by logical connectives. Statements in SPARQL can be translated into Propositional Logics combining the triple patterns:

SPARQL	**Propositional Logics**
$\{f_1, f_2\}$	$f_1 \wedge f_2$
UNION$\{f_1, f_2\}$	$f_1 \vee f_2$

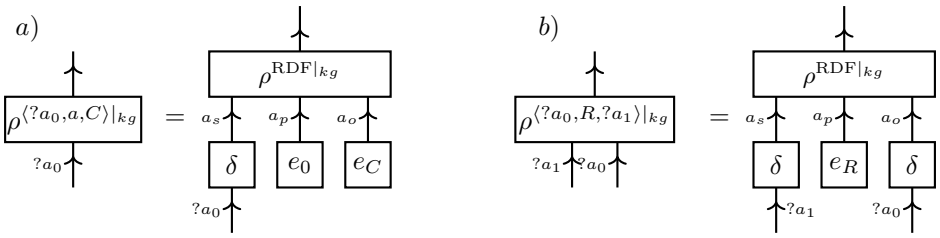


Figure 27: Triple patterns of SPARQL as tensor networks. a) Example of unary triple pattern $\langle ?a_0, a, C\rangle$ specifying whether an individual a_1 is a member of class C. b) Example of a binary triple pattern $\langle ?a_0, R, ?a_1\rangle$ specifying whether individuals a_1 and a_2 have a relation R. By e_0, e_C, e_R we denote the one-hot encodings of the enumeration of the resources $rdf : type, C$ and R.

A SPARQL query p consistent of multiple triple pattern is the composition of the triple patterns groundings. We can represent its groundings by a contraction of the relational encodings of basic graph patterns and the respective logical connectives. Alternatively, the more direct effective calculus developed in Section 17.4 can be applied. The latter way is especially compelling, since large parts of typical SPARQL queries are mere conjunctions of the triple patterns, which is reflected in SPARQL syntax (default interpretation of instruction-free lists of patterns are conjunctions, while differing connectives need to be specified by additional instructions).

Further SPARQL features are

- FILTER$\{\cdot\}$ does not depend on triple patterns (e.g. numeric inequalities, regex functions on strings). We can regard it as another basic formula, which does not result from a slicing of the RDF grounding tensor. Besides that, we can understand it as formulas and include it in compositions.
- OPTIONAL$\{\cdot\}$ would result in $\mathbb{I}$ leg vectors, when there is a missing variable assignment resulting.

13.4 Probabilistic Relational Models

So far we have studied Markov Logic Networks in Propositional Logics as probability distributions over worlds. In FOL they define probability distributions over relations in worlds with a fixed set of objects. More generally, such models are probabilistic relational models (see for an overview Getoor and Taskar (2019).

We in this section show, when and how we can interpret likelihoods of Markov Logic Networks in First Order Logic in terms of samples of a Markov Logic Network in Propositional Logics.

13.4.1 Markov Logic Networks in FOL

Following **?** we define Markov Logic Networks as templates for distributions, which instantiate random worlds when choosing a set of objects A. 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.

Definition 61 (Markov Logic Networks in FOL)**.** *A Markov Logic Network is a template of probability distributions defined by a set $\mathcal{Q}$ of FOL formulas with maximal arity n, which is weighted by a function $\theta : \mathcal{Q} \to \mathbb{R}$. The Markov Logic Network instantiated for a given set of objects A and a base measure ν is the random world, which is a member of the exponential family with sufficient statistics*

$$\phi_l(W) = \langle \{q_l|_W\}\rangle [\varnothing]$$

and canonical parameters θ.

The statistics

$$\langle \{q_l|_W\}\rangle [\varnothing]$$

can be interpreted as the number of substitutions to a formula, such that the formula ist satisfied. Each substitution satisfying a formula adds a factor of $\exp[\theta_l]$ to the probability of the respective world before normalization.

When constructing a world tensor to a theory with predicates of different order, we already argued that we extend the arity of predicates by tensor products with e_0. To define random world tensors, we then restrict the corresponding

base measure to be supported only on those worlds where the extended predicates hold only at the individual a_0 at the extended axis.

We choose extraction formulas q_k such that any formula in the FOL MLN has a propositional equivalent (see Definition 59). The statistic map is then a formula selecting tensor as in the propositional logic case contracted with the groundings of q_k.

13.4.2 Importance Formula

Analogous to a guard formula in (Koller and Friedman, 2009, Definition 6.11)!

We want to reduce the number of object tuples influencing the probability distribution in order to arrive at an interpretation of FOL MLNs as likelihoods to datasets of propositional MLNs.

To this end, we mark pairs of objects relevant to the distributions by an 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 Definition 67)

$$\underline{p} = \sum_{j\in[m]} \left(\bigotimes_{l\in[n]} e_{o^j_l}[O_l] \right) .$$

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 MLN to those world coinciding with groundings of the importance formula coinciding with $\underline{p}$ by designing a base measure

$$\nu_{\underline{p}} = \begin{cases} 1 & \text{if} \quad p|_W = \underline{p} \\ 0 & \text{else} \end{cases} .$$

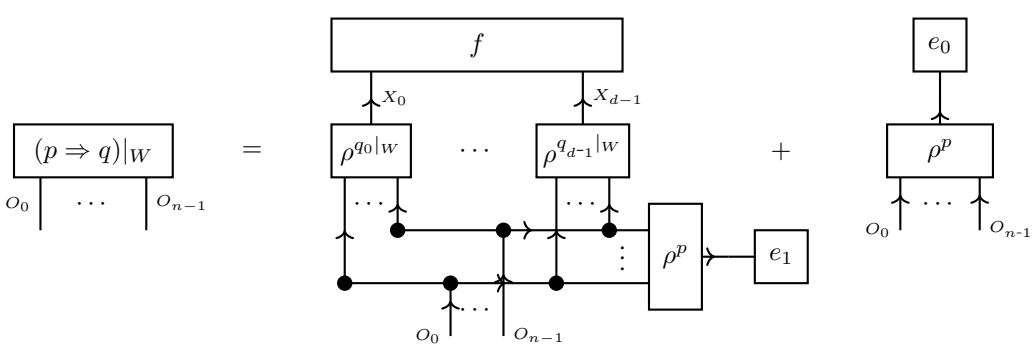
The base measure restricts the Markov Logic Network to be those worlds, where $p|_W$ is given by a fixed tensor $\underline{p}$. This amounts to assuming that $p|_W$ represents certain information about a FOL world, where other formulas are uncertain.

To reduce the likelihood of a world to we make the assumption that all formulas in a Markov Logic Network are of the form

$$q_l(a_{o_0}, \dots, a_{o_{n-1}}) = \big(p(a_{o_0}, \dots, a_{o_{n-1}}) \Rightarrow \tilde{q}(a_{o_0}, \dots, a_{o_{n-1}})\big) \tag{57}$$

that is a rule with the importance formula being the premise. When this assumption holds, we can think of the importance formula as a conditions on individuals to satisfy a statistical relation given by $\tilde{q}$.

We depict the assumption, that any formula is of the form (57) in the diagram



where the second summand depends only on the query p and therefore does not appear in the likelihood.

13.4.3 Decomposition of the log likelihood

Given a FOL MLN, the probability of a world W with $p|_W = \underline{p}$ is

$$\mathbb{P}^{(\mathcal{Q}|_A, \theta, \nu_{\underline{p}})}[W] = \frac{1}{\mathcal{Z}\left(\mathcal{Q}|_A, \theta, \nu_{\underline{p}}\right)} \exp\left[\sum_{q\in\mathcal{Q}} \theta_q \left\langle (p \Rightarrow \tilde{q})|_W \right\rangle [\varnothing] \right]$$

where the partition function is

$$\mathcal{Z}\Big(\mathcal{Q}|_A, \theta, \nu_{\underline{p}}\Big) = \sum_{W\,:\,p|_W = \underline{p}} \exp\left[\sum_{q\in\mathcal{Q}} \theta_q \left\langle (p \Rightarrow \tilde{q})|_W \right\rangle [\varnothing]\right]$$

Let us now decompose the statistics into constant and varying terms. We have

$$\left\langle (p \Rightarrow \tilde{q})|_W \right\rangle [\varnothing] = \left\langle p \wedge \tilde{q}|_W \right\rangle [\varnothing] + \left\langle \neg p|_W \right\rangle [\varnothing] \ ,$$

where the the second term is constant among the supported worlds and the first can be enumerated by the satisfied substitutions of p, that is

$$\left\langle p \wedge \tilde{q}|_W \right\rangle [\varnothing] = \sum_{j\in[m]} \tilde{q}|_W \left[O_{[n]} = o^j_{[n]}\right] \ .$$

Using these insights we decompose a normalized log likelihood as

$$\frac{1}{m} \ln \left[\mathbb{P}^{(\mathcal{Q}|_A, \theta, \nu_{\underline{p}})}[W]\right] = \frac{1}{m} \sum_{j\in[m]} \tilde{q}|_W \left[O_{[n]} = o^j_{[n]}\right] - \frac{1}{m} \ln \left[\frac{\mathcal{Z}\Big(\mathcal{Q}|_A, \theta, \nu_{\underline{p}}\Big)}{\exp\left[\left\langle \theta \right\rangle [\varnothing] \cdot \left\langle \neg p|_W \right\rangle [\varnothing]\right]}\right] \tag{58}$$

We notice a similarity with the likelihood in the case of MLNs in propositional logic. When we interpret each pair $a_{o_0}, \ldots, a_{o_{n-1}} \in A$ satisfying $p(a_{o_0}, \ldots, a_{o_{n-1}}) = 1$ as a datapoint, and choose the formulas $\mathcal{F}$ from atomic formulas connected by propositional connectives, both approaches are equivalent. However, the partition function couples multiple samples, and prevents a straight forward interpretation as an empirical dataset. We in the next section present assumptions on the tuples satisfying p, which lead to a factorization of the partition function.

13.4.4 Interpretation as Likelihood of Propositional Dataset

Theorem 60. *Let $\nu_{\underline{p}}$ be a base measure of worlds such that the vectors*

$$\left(q_0|_{\tilde{W}} \left[O_0 = o^j_0, \ldots, O_{n-1} = o^j_{n-1}\right], \ldots, q_{d-1}|_{\tilde{W}} \left[O_0 = o^j_0, \ldots, O_{n-1} = o^j_{n-1}\right]\right) \tag{59}$$

for $j \in [m]$ are independent and identical distributed by a distribution μ, when distributing $\tilde{W}$ by $\nu_{\underline{p}}$.

Let there further be a set of formulas $\mathcal{Q}$, which formulas $q \in \mathcal{Q}$ are representable by

$$q = p \Rightarrow \tilde{q}$$

and we find a propositional formula f with

$$\tilde{q}|_W = \left\langle \{\rho^{q_k|_W} \ : \ k \in [d]\} \cup \{f\} \right\rangle \left[O_{[n]}\right] \ .$$

We then have for the likelihood of any by $\nu_{\underline{p}}$ supported world W that

$$\frac{1}{m} \ln \left[\mathbb{P}^{(\mathcal{Q}|_A, \theta, \nu_{\underline{p}})}[W]\right] = \frac{1}{m} \ln \left[\mathbb{P}^{((\mathcal{F}, \theta), \mu)}[D]\right] - \frac{1}{m} \sum_{j\in[m]} \ln \left[\mu[X_{[d]} = D(j)]\right]$$

where μ is the distribution of the random vectors (59) and by D we denote the dataset

$$D = \left\{ \left(q_0|_W \left[O_0 = o^j_0, \ldots, O_{n-1} = o^j_{n-1}\right], \ldots, q_{d-1}|_W \left[O_0 = o^j_0, \ldots, O_{n-1} = o^j_{n-1}\right]\right) \ : j \in [m] \right\}$$

and $\mathcal{F}$ is the set of propositional formulas to $q \in \mathcal{Q}$.

To show the theorem, we show first in the following lemma the factorization of the partition function of the FOL MLN.

Lemma 21. *Given the assumptions of Theorem 60, we have*

$$\frac{\mathcal{Z}\Big(\mathcal{Q}|_A, \theta, \nu_{\underline{p}}\Big)}{\exp\left[\left\langle \theta \right\rangle [\varnothing] \cdot \left\langle \neg p|_W \right\rangle [\varnothing]\right]} = \left(\mathcal{Z}((\mathcal{F}, \theta), \mu)\right)^m \ .$$

Proof. We have

$$\begin{aligned}
\mathcal{Z}\left(\mathcal{Q}|_A, \theta, \nu_{\underline{p}}\right) &= \mathbb{E}_{W\sim\nu_{\underline{p}}}\left[\exp\left[\sum_{q\in\mathcal{Q}} \theta_q \left\langle (p \Rightarrow \tilde{q})|_W \right\rangle [\varnothing]\right]\right] \\
&= \exp\left[\langle\theta\rangle [\varnothing] \cdot \langle \neg p|_W\rangle [\varnothing]\right] \cdot \mathbb{E}_{W\sim\nu_{\underline{p}}}\left[\exp\left[\sum_{q\in\mathcal{Q}} \theta_q \sum_{j\in[m]} \tilde{q}|_W \left[O_{[n]} = o^j_{[n]}\right]\right]\right] \\
&= \exp\left[\langle\theta\rangle [\varnothing] \cdot \langle \neg p|_W\rangle [\varnothing]\right] \cdot \mathbb{E}_{W\sim\nu_{\underline{p}}}\left[\prod_{j\in[m]} \exp\left[\sum_{q\in\mathcal{Q}} \theta_q \cdot \tilde{q}|_W \left[O_{[n]} = o^j_{[n]}\right]\right]\right] \\
&= \exp\left[\langle\theta\rangle [\varnothing] \cdot \langle \neg p|_W\rangle [\varnothing]\right] \cdot \prod_{j\in[m]} \mathbb{E}_{W\sim\nu_{\underline{p}}}\left[\exp\left[\sum_{q\in\mathcal{Q}} \theta_q \cdot \tilde{q}|_W \left[O_{[n]} = o^j_{[n]}\right]\right]\right]
\end{aligned}$$

Here we used, that since the substitutions of the atom formulas at the respective object tuples are independent, also the variables

$$\exp\left[\theta_q \cdot \tilde{q}|_W \left[O_{[n]} = o^j_{[n]}\right]\right]$$

for $j \in [m]$ are independent.

Since each $\tilde{q}|_W \left[O_{[n]}\right]$ depends only on the random value $x^j_k = q_k[O_{[n]}]$ we have

$$\begin{aligned}
\mathbb{E}_{W\sim\nu_{\underline{p}}}\left[\exp\left[\sum_{q\in\mathcal{Q}} \theta_q \cdot \tilde{q}|_W \left[O_{[n]} = o^j_{[n]}\right]\right]\right] &= \sum_{x_{[d]}\in[2]^d} \mathbb{E}_{W\sim\nu_{\underline{p}}}\left[\forall k \in [d] \,:\, q_k[O_{[n]}] = x_k\right] \\
&\qquad \cdot \exp\left[\sum_{f\in\mathcal{F}} \theta_f \cdot f\left[X_{[d]} = x_{[d]}\right]\right] \\
&= \sum_{x_{[d]}\in[2]^d} \mu[X_{[d]} = x_{[d]}] \cdot \exp\left[\sum_{f\in\mathcal{F}} \theta_f \cdot f\left[X_{[d]} = x_{[d]}\right]\right] \\
&= \mathcal{Z}((\mathcal{F}, \theta), \mu)\,.
\end{aligned}$$

Combining the above, we arrive at the claim. □

Proof of Theorem 60. Use Equation 58 and the Lemma 21. Note that we need to correct the likelihood by the averalge log basemeasure on the data, since that term is appearing in the likelihood of a MLN. □

Let us now investigate, when the assumptions of independent data can be matched.

Lemma 22. *Let p and $q_0, \ldots, q_{d-1}$ be quantor and constant free and let the index tuples of the support of $\underline{p}$ be pairwise disjoint. Then the vectors* (59) *are pairwise independent.*

Proof. Then we can reduce each sample as dependent only on an independent random world with domain by the respective objects. Quantor and constant-free is needed that this reductions is possible. □

Situations where atom base μ measures are not uniform:

- extraction formula being a) conjunctions of predicates: Probability that they are satisfied decreases b) disjunctions of predicates: Probability that they are satisfied increases
- extraction formula coinciding with importance formula: Always satisfied
- extraction formulas contradicting each other, more general not independent from each other

13.4.5 Approximation by Independent Samples

In general, we cannot assume that the $p, q_0, \ldots, q_{d-1}$ do not influence each other.

We approximate the partition function into factors to each solution map of p. This amounts to the assumption, that the atoms created to each tuple are independent.

For example, when solution maps share the resources, which form the arguments of an atom extraction query, they coincide on this atom and are thus dependent.

If the expectations of each sample with respect to the marginalized distributions coincide, the average of empirical distribution also coincides with these (by linearity). When the creation of samples has sufficient mixing properties, the empirical distribution converges to this expectation in the asymptotic case of large numbers of samples.

13.4.6 Extraction of Samples from FOL Knowledge Bases

The decomposition of the likelihood suggests the following approach to generate samples from groundings:

- Define for $k \in [d]$ queries q_k generating the the atoms X_k: Predicates along with assignment of variables / constants to its positions.
- Define a query formula p, which we decompose in the basis CP decomposition for later interpretation of datapoints
- Contract the groundings of each formula f^k with the grounding of p to build a data core

13.4.7 Representation by Tensor Networks

Let us now understand the extraction process as a relation between a tuple of individuals and the extracted world in the factored system of atoms X_k.

$$\mathcal{R} = \{(a_{o_0}, \ldots, a_{o_{n-1}}, X_0, \ldots, X_{d-1}) \,:\, p(a_{o_0}, \ldots, a_{o_{n-1}}) = 1 \,,\, \forall_{k\in[d]} \,:\, X_k = q_k(a_{o_0}, \ldots, a_{o_{n-1}})\}$$

Towards constructing the encoding of this we enumerate the individuals in the set A and use in the following the respective one-hot encoding

$$e : A \to \mathbb{R}^{|A|} \,.$$

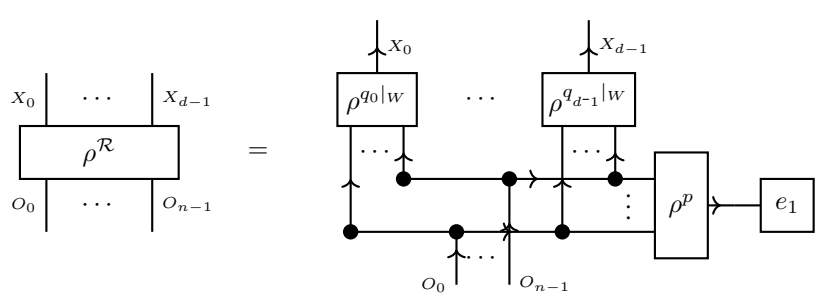
The combination of the queries p and $\{q_0, \ldots, q_{d-1}\}$ by the relational encoding

$$\rho^{\mathcal{R}} \subset \left(\bigotimes_{l\in[n]} \mathbb{R}^r \right) \otimes \left(\bigotimes_{k\in[d]} \mathbb{R}^2 \right)$$

defined by

$$\rho^{\mathcal{R}} = \left\{ (e_{a_{o_0},\ldots,a_{o_{n-1}}}, X_0, \ldots, X_{d-1}) \,:\, p(a_{o_0}, \ldots, a_{o_{n-1}}) = 1 \,,\, \forall_{k\in[d]} \,:\, X_k = q_k(a_{o_0}, \ldots, a_{o_{n-1}}) \right\} \,.$$

We can represent the encoding $\rho^{\mathcal{R}}$ by the diagram



Here the contraction of ρ^p with the truth vector e_1 represents the matching condition posed by p when extracting pairs of individuals.

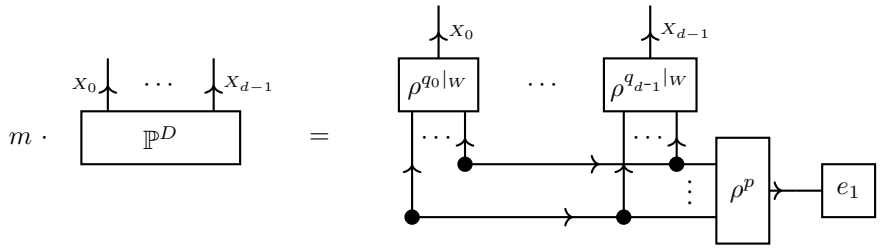
The empirical distribution is then the normalized contraction leaving only the legs to the extracted atomic formulas open, that is

$$\mathbb{P}^D = \frac{\left\langle \{\rho^{\mathcal{R}}\} \right\rangle \left[X_{[d]}\right]}{\left\langle \{\rho^{\mathcal{R}}\} \right\rangle [\varnothing]} .$$

Here the number of extracted data is the denominator

$$m = \left\langle \{\rho^{\mathcal{R}}\} \right\rangle [\varnothing] .$$

We depict this by



13.4.8 Basis CP Decomposition of extracted data

To connect with the empirical distribution introduced in Section 3.8 we now show how the dataset D extracted from the interpretations of the formulas $p, q_0, \ldots, q_{d-1}$ on a FOL world W can be represented by tensor networks.

Each datacore is then a contraction with the grounding of a formula, which is contracted with the grounding of the extraction query in the basis CP decomposition,

$$\rho^{D_k} = \left\langle \left\{ \rho^{q_k|_W}, p|_W \right\} \right\rangle [J, X_k]$$

see Figure 28.

Theorem 61. *We have*

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

and thus

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

Proof. Using that $p|_W$ is binary and the core can be copied (ref to theorem in binary calculus). □

Towards efficient calculation of the data cores, we build a basis CP decomposition of $p|_W$, where we further demand $\sigma = \mathbb{I}$. This is a collection of basis leg cores $V^{\underline{p},l}$ such that

$$\underline{p}[O_{[n]}] = \left\langle \left\{ V^{\underline{p},l}[J, O_l] \,:\, l \in [n] \right\} \right\rangle \left[O_{[n]}\right] .$$

The log-likelihood of a MLN is then the contraction of the data cores with the representation of the MLN wrt the atomic formulas q_k.

13.4.9 Representation with auxiliary term variables

When many atom extraction formulas differ only by a constant, we can replace the constant by an auxiliary term variable. The atoms are then the atomizations of this variable (see Section 9.4), treated as a categorical variable, with respect to the constant in the extraction query. The advantages are that we can avoid the ρ-formalims and directly model the categorical distributions.

13.4.10 Generic Tensor Network Decomposition of Extracted Data

More naturally: Formulas are compositions of predicates. When quantor free then a tensor network.

Remark 27 (Alternative Representation of empirical distributions)**.** *In many applications such as the computation of log-likelihoods we can use any representation of the empirical distribution by tensor networks. It is thus not necessary to compute the data cores as above, unless one requires a list of the extracted samples.*

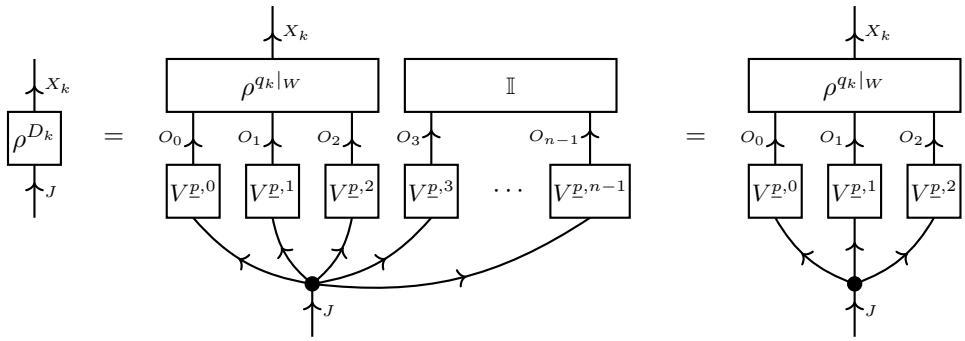


Figure 28: Generation of Datacores given a formula f, whose variables (here a_1, a_2, a_3) are selected from an extraction query p. Variables, which are not appearing in the formula f are trivialized over (here $a_4, \ldots, a_n$). For consistency we denoted the index $x_{f(k)}$ by x_k.

13.4.11 Design of the Formulas

Most intuitive when labeling individuals by classes. Extraction formulas $q_0, \ldots, q_{d-1}$ can then be defined by subclasses of the member of a class and relations between objects of different classes. We then choose $\mathcal{F}$ as more involved formulas decomposed into connectives acting on these atoms. The extraction query p is then designed based on class memberships to ensure, that the arguments of the formulas are always of specific classes.

We propose to

- Execute an extraction query to get pairs of individuals (the pairDf).
- Propositionalize the FOL Formulas independently on each tuple taking the individuals as a set of constant and filtering on the possible properties of each individuals. (Can understand as adding knowledge that most of the relations do not hold)
- Understand each such generated knowledge base as datapoint and average over them to get the empirical distribution to be fit.
- Fit a MLN describing the statistical relations of unseen results of the extraction query, based on likelihood maximation.

13.5 Generation of FOL worlds

Definition 62 (Reproduction of Empirical Distributions)**.** *Given an empirical distribution $\mathbb{P}^D \in \bigotimes_{k\in[d]} \mathbb{R}^2$, we say that a triple $(W, p, q_{[d]})$ of a FOL world W an importance formula p and extraction formulas q_d reproduces $\mathbb{P}^D$, when*

$$\mathbb{P}^D = \left\langle \{p|_W\} \cup \{\rho^{q_k|_{kg}\,:\,k\in[d]}\} \right\rangle \left[X_{[d]}|\varnothing\right] \ .$$

13.5.1 Samples by single objects

The first reproduction scheme identifies the datapoints with objects $A = [m]$

Theorem 62. *Given a dataset D the world $W[L, O]$*

$$W[L, O] = \sum_{k\in[d]} \sum_{j\in[m]\,:D_k(j)=1} e_k\,[L] \otimes e_j\,[O]$$

reproduces with the trivial importance query and extraction queries coinciding with the predicates the dataset D.

13.6 Samples by pairs of objects

We instantiate multiple objects for each datapoint, one for each variable of the importance formula, i.e. $A = [m] \times [n]$

13.7 Example: Generation of Knowledge Graphs

So far we have discussed, how MLNs for FOL Knowledge Bases such as Knowledge Graphs can be built by extracting data. Conversely, any binary tensor can be interpreted as a Knowledge Graph. To be more precise, we follow the intuition that the ones coordinates mark possible worlds compatible with the knowledge about a factored system. Each possible world can then be encoded in a subgraph of the Knowledge Graph representing the world.

This amounts to an "inversion" of the data generation process described in the subsection above.

Definition 63 (Reproduction of Empirical Distributions)**.** *Given an empirical distribution $\mathbb{P}^D \in \bigotimes_{k\in[d]} \mathbb{R}^2$, we say that a triple $(kg, p, \{q_0, \ldots, q_{d-1}\})$ of a Knowledge Graph kg and queries p, q_k reproduces $\mathbb{P}^D$, when*

$$\mathbb{P}^D = \left\langle \{p|_{kg}\} \cup \{\rho^{q_k|_{kg}\,:\,k\in[d]}\} \right\rangle \left[X_{[d]}|\varnothing\right] \ .$$

Any distribution with rational coordinates is an empirical distribution, since each coordinate can be interpreted as the frequency of the respective world in the data D.

13.7.1 Samples by single resources

TBox: The categorical variables of the factored system are the coarsest classes. We define atomic formulas by the state indicators of each categorical variable as in Section 9.4. Each such atomic formula corresponds with a sub-class of the classes. By definition, each collection of state indicators define thus pairwise disjoint subclasses.

ABox: The samples are represented by single individuals in the Knowledge Graph. Their sub-class memberships corresponding with the categorical variables of the system are instantiated whenever the atom is true in the sample.

Theorem 63. *Let there any empirical distribution $\mathbb{P}^D \in \bigotimes_{k\in[d]} \mathbb{R}^2$ and $m \in \mathbb{N}$ such that $\operatorname{im}\left(m \cdot \mathbb{P}^D\right) \subset \mathbb{N}$. Then the tuple $(kg, p, \{q_0, \ldots, q_{d-1}\})$ defined by*

$$kg = \bigcup_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \{(s_{j,x_0,\ldots,x_{d-1}} \quad \text{rdf:type} \quad C) : j \in [m \cdot \mathbb{P}^D(x_0, \ldots, x_{d-1})]\} \tag{60}$$

$$\bigcup_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \{(s_{j,x_0,\ldots,x_{d-1}} \quad \text{rdf:type} \quad C_k) : j \in [m \cdot \mathbb{P}^D(x_0, \ldots, x_{d-1})], k \in [d], x_k = 1\} \tag{61}$$

p= SELECT { ?x } WHERE { ?x rdf:type *C .}*

q_k = SELECT { ?x } WHERE { ?x rdf:type *C_k .}*

reproduces $\mathbb{P}^D$.

Proof. With respect to any enumeration of the resources of kg we have

$$p|_{kg} = \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]} \sum_{j\in[m\cdot\mathbb{P}^D(x_0,\ldots,x_{d-1})]} e_{s_{j,x_0,\ldots,x_{d-1}}} \tag{62}$$

and

$$q_k|_{kg} = \sum_{x_0,\ldots,x_{d-1}\in\times_{k\in[d]}[2]\,:\,x_k=1} \sum_{j\in[m\cdot\mathbb{P}^D(x_0,\ldots,x_{d-1})]} e_{s_{j,x_0,\ldots,x_{d-1}}} \ . \tag{63}$$

Summing over the resource variables of these tensors in a contraction we get

$$\left\langle \{p|_{kg}\} \cup \{\rho^{q_k|_{kg}\,:\,k\in[d]}\} \right\rangle [\{X_0, \ldots, X_{d-1}\}] = \sum_{k\in[d]} m \cdot \mathbb{P}^D(x_0, \ldots, x_{d-1}) \cdot e_{x_0,\ldots,x_{d-1}} = m \cdot \mathbb{P}^D \tag{64}$$

and therefore

$$\left\langle \{p|_{kg}\} \cup \{\rho^{q_k|_{kg}\,:\,k\in[d]}\} \right\rangle [\{X_0, \ldots, X_{d-1}\}|\varnothing] = \mathbb{P}^D \ . \tag{65}$$

□

In this simple Knowledge Graph, Description Logic is expressive enough to represent any formula q composed of the formulas $q_0, \ldots, q_{d-1}$.

13.7.2 Samples by pairs of resources

Remark 28 (Refinement of the Samples)**.** *We can split each sample node into a pair of individuals. For this we need to specify, which each class membership will be encoded in a unary or binary attribute of the splitted individuals. This specification is possible based on the extraction query and the atomic formulas.*

Taking any importance query p, which has no permutation symmetries, we can instantiate each projection variable for each sample and prepare the links according to the triple patterns. When the atom queries $q_0, \dots, q_{d-1}$ have different triple patterns compared with p, we instantiate those in cases where $x_k = 1$.

Label individuals $a_{j,l}$ by dataindex and variable index and

Theorem 64. *Let there be queries $p, q_0, \dots, q_{d-1}$, a Knowledge Graph kg containing individuals $a_{j,l}$ and a data map D. If*

$$p|_{kg} = \sum_j \bigotimes_{l \in [n]} e_{j,l}$$

and for any $k \in [d]$

$$q_k|_{kg} = \sum_{j: D^k(j)=1} \bigotimes_{l \in q_k} e_{j,l}$$

Proof. It can be shown that the contraction is

$$\sum_j \bigotimes_{k \in [d]} e_{D^k(j)} \,.$$

□

Theorem 65. *The Knowledge Graph*

$$\mathrm{RDF}|_{kg} = \sum_{t \in p} \sum_j t(a_{j,l}) + \sum_{k \in [d]} \sum_{t \in q_{k \in [d]}} \sum_{j: D^k(j)=1} t(a_{j,l})$$

reproduces the data, if the summed tensors are pairwise orthogonal. Here we denote by t(..) the one-hot encoding of the tuple when mapping the corresponding individuals on the projection variables of the triple pattern t.

13.7.3 General orthogonal projection approach

Given any triples, such that their composed projections to pairwise different triple patterns vanish. One reproducing KG is then the sum of the projection of any reproducing KG.

Theorem 66. *Let $p, q_0, \dots, q_{d-1}$ contain of triple patterns such that for $t \neq \tilde{t}$*

$$P_t \circ P_{\tilde{t}} = 0 \,.$$

For any KG reproducing the data D also the KG

$$\sum_t P_t \mathrm{RDF}|_{kg}$$

is a reproducing KG.

Proof. By coinciding on all triple patterns t since

$$P_t \left(\sum_{\tilde{t}} P_{\tilde{t}} \mathrm{RDF}|_{kg} \right) = P_t P_t \mathrm{RDF}|_{kg} = P_t \mathrm{RDF}|_{kg} \,.$$

Then the extraction contraction on kg and $\sum_t P_t \mathrm{RDF}|_{kg}$ coincides. □

From this perspective, it is obvious that any triple in the orthogonal complement of all projections can be added and the KG stays reproducing.

13.8 Discussion

Models of FOL are called Probabilistic Relational Models. Extensions are models that also handle structural uncertainty, i.e. distributions of worlds with varying A.

Part III
Analysis of the Contraction Calculus

Based on the logical interpretation we often handle tensor calculus with specific tensors. Often, they are binary (that is their coordinates are in $\{0, 1\}$ corresponding with a Boolean), and sparse (that is having a decomposition with less storage demand). We investigate it in this part in more depth the properties of such tensors, which where exploited in the previous parts.

14 Coordinate Calculus

In the previous chapters, information to states has been stored in coordinates of a tensor. To distinguish from other schemes of calculus, we call this scheme of storing and retrieving information the coordinate calculus. We in this chapter investigate in more depth, which operations can be performed based on such tensors and proof the applied properties.

Add summations and further operations?

Discuss things like $\exp[T]$ as coordinatewise operations, by circles in the factor graph diagrams.

14.1 One-hot encodings as basis

Lemma 23 (Basis of tensor spaces)**.** *The image of the one-hot encoding map is a linear basis of the tensor space* $\bigotimes_{k\in[d]} \mathbb{R}^{m_k}$*. Any element* $f \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ *has a decomposition*

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

We notice that the coordinates are the weights to the basis elements in the one-hot decomposition.

Proof. For any $\tilde{x}_0, \ldots, \tilde{x}_{d-1} \in \bigtimes_{k\in[d]}[m_k]$ we have

$$f\left[X_{[d]} = \tilde{x}_{[d]}\right] = \sum_{x_0,\ldots,x_{d-1}} f[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}] \cdot e_{x_0,\ldots,x_{d-1}}\left[X_{[d]} = \tilde{x}_{[d]}\right] = f\left[X_{[d]} = \tilde{x}_{[d]}\right] \cdot e_{x_0,\ldots,x_{d-1}}\left[X_{[d]} = \tilde{x}_{[d]}\right]$$

□

Definition 64. *Given any real-valued function*

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

we define the coordinate encoding by

$$T^f = \sum_{x_0,\ldots,x_{d-1}\in\bigtimes_{k\in[d]}[m_k]} f(x_0, \ldots, x_{d-1}) \cdot e_{x_0,\ldots,x_{d-1}} \,.$$

Theorem 67 (Coordinate Calculus)**.** *Given any tensor* $T[X_0, \ldots, X_{d-1}]$ *can retrieve its coordinate indexed by* $x_0, \ldots, x_{d-1}$ *as*

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

Proof. We use the decomposition in Lemma 23 and have

$$\begin{aligned}\left\langle\{T, e_{x_0,\ldots,x_{d-1}}\}\right\rangle[\varnothing] &= \sum_{\tilde{x}_0,\ldots,\tilde{x}_{d-1}} T\left[X_0 = \tilde{x}_0, \ldots, X_{d-1} = \tilde{x}_{d-1}\right] \cdot \left\langle\{e_{\tilde{x}_0,\ldots,\tilde{x}_{d-1}}, e_{x_0,\ldots,x_{d-1}}\}\right\rangle[X_0, \ldots, X_{d-1}] \\ &= \sum_{\tilde{x}_0,\ldots,\tilde{x}_{d-1}} T\left[X_0 = \tilde{x}_0, \ldots, X_{d-1} = \tilde{x}_{d-1}\right] \cdot \delta_{(\tilde{x}_0,\ldots,\tilde{x}_{d-1}),(x_0,\ldots,x_{d-1})} \\ &= T\left[X_0 = x_0, \ldots, X_{d-1} = x_{d-1}\right] ,\end{aligned}$$

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 Theorem 67.

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 contraction and in particular coordinate retrieval.

Theorem 68. *Given a tensor network $\mathcal{T}^{\mathcal{G}}$ on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, disjoint subsets $A, B \subset \mathcal{V}$ and $x_B \in [m_B]$, we have*

$$\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle[X_A, X_B = x_B] = \left\langle\{\langle\{T^e\}\rangle\left[X_{e/B}, X_B = x_B\right] \,:\, e \in \mathcal{E}\}\right\rangle[X_A] \,.$$

Proof. Using a delta tensor, which copies basis vector when contracting with one. □

If we retrieve a single coordinate of a tensor, we have the situation $A = \varnothing$, $B = \mathcal{V}$. In that case, Theorem 68 shows, that the coordinate is the product of the coordinates of the cores.

14.2 Selection Encodings

Selection encodings as introduced in Definition 15 are best understood in terms of linear mapping interpretations of tensors. We will first provide by basis encodings a generic relation between the coordinatewise tensor definitions in this work and linear maps.

We then show the utility of this perspective in the representation of composed linear functions. The results are applicable in the exponential family theory, in the tensor representation of energies and means.

14.2.1 Tensors as linear maps

The state sets $\bigtimes_{k\in[d]}[m_k]$ can be interpreted as an enumeration of basis elements e_x of the tensor space $\bigotimes_{k\in[d]}\mathbb{R}^{m_k}$.

Along this interpretation, tensors have an interpretation as maps between tensor spaces.

Any tensor and any partition of its variables into two sets can be interpreted as the basis elements of a linear map between the tensor spaces of the respective variables.

Tensor valued functions on state sets $\bigtimes_{k\in[d]}[m_k]$ are an intermediate representation.

Definition 65. *Let there be two tensor spaces V_1 and V_2 with basis by sets $\mathcal{M}_1 \subset V_1$ and $\mathcal{M}_2 \subset V_2$ of cardinality m_1 and m_2, which are enumerated by variables O_1 and O_2. The basis encoding of a linear map $f \in \mathbb{L}(V_1, V_2)$ is the tensor*

$$\beta^f[O_1, O_2] \in \mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2} \,.$$

Basis encodings are standard linear algebra tools, where matrices are understood as linear maps between vector spaces.

Theorem 69. *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 chosen bases in the spaces. We have*

$$\beta^{F^2\circ F^1}[O_1, O_3] = \left\langle\left\{\beta^{F^2}[O_2, O_3], \beta^{F^1}[O_1, O_2]\right\}\right\rangle[O_1, O_3] \,.$$

Proof. By basis decompositions and representations of linear maps as contractions. □

A typical instance is matrix multiplication, where matrices understood as representations of linear maps.

14.2.2 Selection encodings

Selection encodings (see Definition 15) are related to basis encodings of linear maps as we show in the next theorem.

Theorem 70. *Given a function*

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

we define a linear map $F^f \in \mathbb{L}(\bigotimes_{k\in[d]} \mathbb{R}^{m_k}, \bigotimes_{s\in[n]} \mathbb{R}^{p_s})$ *by the basis elements to* $x_1 \in \bigtimes_{k\in[d]}[m_k]$ *and* $x_2 \in \bigtimes_{s\in[n]}[p_s]$ *by*

$$F^f(e_{x_1}, e_{x_2}) = \langle \{f(e_{x_1}), e_{x_2}\} \rangle [\varnothing] \ .$$

We then have

$$\gamma^f = \beta^{F^f} \ .$$

While relational encoding works for maps from $\bigtimes_{k\in[d]}[m_k]$ to arbitrary sets (which are enumerated), selection encodings as introduced in Definition 15 require and exploit that their image is embedded in a tensor space.

Given a selection encoding of a function, the function is retrieved by slicing with respect to the

$$f(x) = \gamma^f [X = x, L] \ .$$

More generally, we show in the next Lemma how to construct to any tensor and any partition of its variables functions by slicing operations, such that the tensor is the selection encoding of the function.

Lemma 24. *Let* $T[X_{\mathcal{V}}]$ *be a tensor in* $\bigotimes_{v\in\mathcal{V}} \mathbb{R}^{m_v}$ *and let* A, B *be a disjoint partition of* $\mathcal{V}$*, that is* $A \dot\cup B = \mathcal{V}$*. Then the function*

$$f : \bigtimes_{v\in A} [m_v] \to \bigotimes_{v\in B} \mathbb{R}^{m_v}$$

with coordinates

$$f(x_A) = T[X_A = x_A, X_B]$$

obeys

$$\gamma^f [X_{\mathcal{V}}] = T[X_{\mathcal{V}}] \ .$$

Here we have renamed the selection variables L_A *by the categorical variables* X_A*.*

Proof. From Theorem 69 using the basis encoding equivalence of Theorem 70. □

Example 14 (Markov Logic Networks and Proposal Distributions)**.** *While the statistic of MLN (namely* $\mathcal{H}$*) and the proposal distribution (namely* $\mathcal{H}^T$*) have a common selection encoding, both result from the inverse selection encoding described in Lemma 24. We can construct* $\mathcal{H}^T$ *by first building the selection encoding to* $\mathcal{H}$ *and then applying the construction of Lemma 24 with* $A = L$ *and* $B = X_{[d]}$*.*

We use selection encodings to express compositions of functions, based on the next theorem.

Theorem 71 (Selection Encoding for Linear Compositions)**.** *Let* ϕ *be a tensor valued function from* $\bigtimes_{k\in[d]}[m_k]$ *to* $\mathbb{R}^p$ *with image coordinates* ϕ_l *and let* f *be a tensor Then*

$$\left(\sum_{l\in[p]} f[L_\phi = l] \cdot \phi_l\right) [X_{[d]}] : \bigtimes_{k\in[d]} [m_k] \to \mathbb{R}$$

is represented as

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

Proof. The representation holds, since for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have

$$\left\langle \left\{\gamma^\phi \left[X_{[d]}, L_\phi\right], f[L_\phi]\right\}\right\rangle \left[X_{[d]} = x_{[d]}\right] = \sum_{l\in[p]} f[L_\phi = l] \cdot \phi_l(x_{[d]}) \, .$$

□

This theorem shows, that while relation encodings can represent any composition with another function by a contractions, selection encodings can be used to represent linear transforms. To see this, we interpret ϕ and f in Theorem 71 as basis decompositions of linear maps.

14.3 Discussion

Representations of linear maps is the typical application of tensors, reason for refering to tensor networks as multilinear algebra.

15 Directed Tensor Calculus

We in this chapter investigate the properties of tensors, which where arising in probabilistic and logical reasoning. We observed already before, that

- Conditional probability tensors are directed tensors.
- Logical formulas are binary tensors.

Thus, the set of tensors, which are both directed and binary is of much interest. We will show, that they are equal to the set of relational encodings of functions.

15.1 Directed Tensors

Directionality as defined in Definition 12 represents the constraints on the structure of tensors: Summing over outgoing trivializes the tensor.

Definition 66 (Directed Hypergraph)**.** *A directed hyperedge is a hyperedge, which node set is split into disjoint sets of incoming and outgoing nodes. We say a hypercore T^e decorating a directed hyperedge respects the direction, when it is a conditional probability tensor with respect to the direction of the hyperedge. The hypergraph is acyclic, when there is no nonempty cycle of node tuples (v_1, v_2), such that v_1 is an incoming node and v_2 an outgoing node of the same hyperedge.*

There can be multiple ways to direct a tensor, which an extreme example being the Dirac Delta Tensors. Further example are relational encodings of invertible functions.

Example 15 (Dirac Delta Tensors)**.** *Given a node set e colored with a constant dimension m Diracs Delta Tensors are the tensors*

$$\delta^{e,m} \in \bigotimes_{v \in e} \mathbb{R}^m$$

with coordinates

$$\delta^{e,m}_{x_0,\ldots,x_{d-1}} = \begin{cases} 1 & \textit{if} \quad x_0 = \ldots = x_{d-1} \\ 0 & \textit{else} \end{cases} . \tag{66}$$

The contractions with respect to subsets $\tilde{\mathcal{V}} \subset e$ are

$$\langle \{\delta^{e,m}\} \rangle [X_{\tilde{\mathcal{V}}}] = \begin{cases} m & \textit{if} \quad \mathcal{V} = \varnothing \\ \mathbb{I} & \textit{if} \quad \left|\tilde{\mathcal{V}}\right| = 1 \\ \delta^{\tilde{\mathcal{V}},m} & \textit{else} \end{cases} . \tag{67}$$

Thus are directed for any orientation of the respective edge with exactly one incoming variable.

Lemma 25. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a hypergraph and $\mathcal{T}^{\mathcal{G}}$ a tensor network on $\mathcal{G}$. We build a graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}} \cup \Delta^{\mathcal{G}})$ and a tensor network $\mathcal{T}^{\tilde{\mathcal{G}}}$ by*

- *Recolored Edges $\tilde{\mathcal{E}} = \{\tilde{e} : e \in \mathcal{E}\}$ where $\tilde{e} = \{v^e : v \in e\}$, which decoration tensor $T^{\tilde{e}}$ has same coordinates as T^e*
- *Nodes $\tilde{\mathcal{V}} = \bigcup_{e \in \mathcal{E}} \tilde{e}$*
- *Delta Edges $\Delta^{\mathcal{G}} = \left\{\{v\} \cup \{v^e : e \ni v\} : v \in \mathcal{V}\right\}$ each of which decorated by a delta tensor $\delta^{\{v^e : e \ni v\}}$*

Then we have

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

Proof. For any $x_{\mathcal{V}}$ we have

$$\begin{aligned}\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] &= \left\langle \{T^{\tilde{e}}[X_{\{v^e : v \in e\}}] : e \in \mathcal{E}\} \cup \{\delta^{\{v\} \cup \{v^e \,:\, e \ni v\}}[X_{\{v^e : e \ni v\}}, X_v = x_v] : v \in \mathcal{V}\} \right\rangle [\varnothing] \\ &= \left\langle \{T^{\tilde{e}}[X_{\{v^e : v \in e\}} = x_{\{v : v \in e\}}] : e \in \mathcal{E}\} \right\rangle [\varnothing] \\ &= \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \ ,\end{aligned}$$

which establishes the claim. □

15.1.1 Normation

Normed tensors (see Definition 13) are directed and directed tensors invariant under normation wrt their incoming and outgoing variable, as we show next.

Theorem 72. *For any tensor network $\mathcal{T}^{\mathcal{G}}$ on variables $\mathcal{V}$ that can be normed with respect to $\mathcal{V}^{\text{in}}$ and $\mathcal{V}^{\text{out}}$, the normation is directed with $\mathcal{V}^{\text{in}}$ incoming and $\mathcal{V}^{\text{out}}$ outgoing.*

Proof. We have for any incoming state $x_{\mathcal{V}^{\text{in}}} \in \bigtimes_{v \in \mathcal{V}^{\text{in}}} m_v$ that

$$\left\langle \{ \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\mathcal{V}^{\text{in}} | \mathcal{V}^{\text{out}} \right], e_{x_{\mathcal{V}^{\text{in}}}} \} \right\rangle [\varnothing] = \frac{\left\langle \mathcal{T}^{\mathcal{G}} \cup \{e_{x_{\mathcal{V}^{\text{in}}}}\} \right\rangle [\varnothing]}{\left\langle \mathcal{T}^{\mathcal{G}} \cup \{e_{x_{\mathcal{V}^{\text{in}}}}\} \right\rangle [\varnothing]} \ .$$

By Definition 12, $\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle \left[\mathcal{V}^{\text{out}} | \mathcal{V}^{\text{in}} \right]$ is thus directed. □

The normation operation coincides in cases of non-negative tensors with the conditioning of a Markov Network representing a probability distribution.

15.1.2 Contraction of Directed Tensors

Let us now investigate, which contractions inherit the directionality of the tensors.

Theorem 73. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic hypergraph, such that each node $v \in e$ appears at most in one hyperedge as an outgoing variable and denote $\mathcal{V}^{\text{in}}$ as those nodes, which do not appear as outgoing variables. For any tensor network $\mathcal{T}^{\mathcal{G}}$ respecting the direction of $\mathcal{G}$ we have that*

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

that is $\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}}]$ is a directed tensor with $\mathcal{V}^{\text{in}}$ incoming and $\mathcal{V}/\mathcal{V}^{\text{in}}$ outgoing.

Proof. We show the theorem only for the case of hypergraphs, where variables are appearing at most in two hyperedges. If a hypergraph fails to satisfy this assumption, we apply Lemma 25 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. When orienting each edge

We show the theorem over induction on the number n of cores.

$n = 1$**:** It holds trivially, when $\mathcal{T}^{\mathcal{G}}$ consists of a single core

$n \to n-1$**:** Let us assume, that the claim holds for graphs with $n-1$ hyperedges and let $\mathcal{T}^{\mathcal{G}}$ be a tensor network with n hyperedges. Since the hypergraph is acyclic, we find an edge $e \in \mathcal{E}$ such that all outgoing nodes of e are not appearing as an incoming node in any edge. We then apply Theorem 77 and get

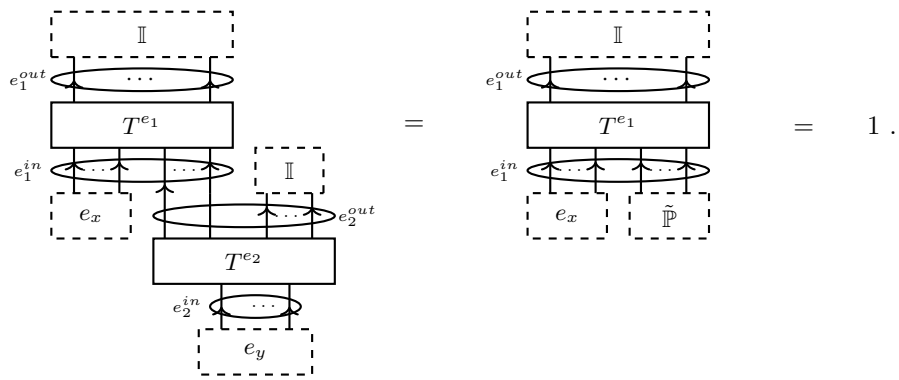
$$\begin{aligned}\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}^{\text{in}}}] &= \left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E}/\{e\})} \cup \{T^e\,[X_{e^{\text{in}}}, X_{e^{\text{out}}}]\} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &= \left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E}/\{e\})} \cup \{\langle \{T^e\} \rangle\,[X_{e^{\text{in}}}]\} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &= \left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E}/\{e\})} \cup \{\mathbb{I}\,[X_{e^{\text{in}}}]\} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \\ &\left\langle \mathcal{T}^{(\mathcal{V}, \mathcal{E}/\{e\})} \} \right\rangle [X_{\mathcal{V}^{\text{in}}}] \ .\end{aligned}$$

We then notice that the hypergraph $(\mathcal{V}, \mathcal{E}/\{e\})$ has $n-1$ hyperedges and each node appears at most once as an incoming and at most once as an outgoing node. Thus, we apply the assumption of the induction and get

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}^{\text{in}}}] = \left\langle \mathcal{T}^{(\mathcal{V},\mathcal{E}/\{e\})} \} \right\rangle [X_{\mathcal{V}^{\text{in}}}] = \mathbb{I}\,[X_{\mathcal{V}^{\text{in}}}] \ .$$

□

Schematically, the direction conservation property argument is depicted as:



16 Contraction Equations

We have observed, that many concepts and theorems in probability theory and logics can be understood as contraction equations. We first provide a summary of the used contraction equations.

16.1 Contraction equations in logical and probabilistic reasoning

Let us summarize the application of contractions and normation in the definition of

- Marginal probabilities (Definition 17, Theorem 1)

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

- Conditional probabilities (Definition 18, Theorem 2)

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

- The partition function of a Markov Networks

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

- The probability distribution of a Markov Network is (Definition 23)

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

Further the following properties have been defined by contraction equations:

- X_0 and X_1 are independent when (Definition 19, Theorem 6)

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

- X_0 and X_1 are called independent conditioned on X_2 when (Definition 20, Theorem 7)

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

16.2 Normation Equations

Theorem 74 (Normation as a Contraction Equation)**.** *For any on $\mathcal{V}^{\text{in}}$ normable tensor $T\left[X_{\mathcal{V}}\right]$, where $\mathcal{V}^{\text{in}} \dot\cup \mathcal{V}^{\text{out}} = \mathcal{V}$, we have*

$$\langle\{T\}\rangle\left[X_{\mathcal{V}}\right] = \langle\{\langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{out}}}|X_{\mathcal{V}^{\text{in}}}\right], \langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{in}}}\right]\}\rangle\left[X_{\mathcal{V}}\right] .$$

Proof. Let us choose indices $x_{\mathcal{V}^{\text{in}}}$ and $x_{\mathcal{V}^{\text{out}}}$. We have that

$$\langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}|X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right] = \frac{\langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}, X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right]}{\langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}\right]}$$

and therefor

$$\langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}, X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right] = \langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}|X_{\mathcal{V}^{\text{out}}} = x_{\mathcal{V}^{\text{out}}}\right] \cdot \langle\{T\}\rangle\left[X_{\mathcal{V}^{\text{in}}} = x_{\mathcal{V}^{\text{in}}}\right]$$

Since the equation holds for arbitrary indices, the claim is established. □

Theorem 75 (Generic Chain Rule)**.** *For any Tensor $T\left[X_{\mathcal{V}}\right]$ and any total order $\prec$ on the nodes $\mathcal{V}$ we have*

$$T\left[X_{\mathcal{V}}\right] = \left\langle\left\{\langle\{T\}\rangle\left[X_v|X_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}}\right] \,:\, v \in \mathcal{V}\right\}\right\rangle\left[X_{\mathcal{V}}\right]$$

Proof. We apply Theorem 74 on the tensor

$$\langle\{T\}\rangle\left[X_v, X_{\{\tilde{v}\,:\,v\prec\tilde{v}, \tilde{v}\neq v\}}|X_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}} = x_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}}\right] ,$$

where $v \in \mathcal{V}$ and $x_{\mathcal{V}}$ are chosen arbitrarly. For any $v \in \mathcal{V}$ we get

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

Applying this equation iteratively and making use of the commutation of contractions we get for any $v \in \mathcal{V}$

$$\langle\{T\}\rangle\left[X_v, X_{\{\tilde{v}\,:\,v\prec\tilde{v}, \tilde{v}\neq v\}}|X_{\{\tilde{v}\,:\,\tilde{v}\prec v, \tilde{v}\neq v\}}\right] = \left\langle\left\{\langle T\rangle\left[X_{\tilde{v}}|X_{\{\bar{v}:\bar{v}\prec\tilde{v}, \bar{v}\neq\tilde{v}\}}\right] \,:\, v \prec \tilde{v}\right\}\right\rangle\left[X_{\mathcal{V}}\right] .$$

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. □

16.3 Proof of Hammersley-Clifford Theorem

Different to the standard case, we show a version of Hammersley-Clifford for hypergraphs.

Lemma 26. *Let $T\left[X_{\mathcal{V}}\right]$ be a positive tensor and $y_{\mathcal{V}}$ an arbitrary index. Then we have*

$$T\left[X_{\mathcal{V}}\right] = \left\langle\left\{\left(\langle\{T\}\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right]\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} \,:\, \bar{\mathcal{V}} \subset \tilde{\mathcal{V}} \subset \mathcal{V}\right\}\right\rangle\left[X_{\mathcal{V}}\right] ,$$

where the exponentiation is performed coordinatewise and positivity of T ensures the well-definedness.

Proof. It suffices to show, that for an arbitrary index $x_{\mathcal{V}}$ be an arbitrary index we have

$$T\left[X_{\mathcal{V}} = x_{\mathcal{V}}\right] = \prod_{\tilde{\mathcal{V}}\subset\mathcal{V}} \prod_{\bar{\mathcal{V}}\subset\tilde{\mathcal{V}}} \left(\langle\{T\}\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right]\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} .$$

We do this by applying a logarithm on the right hand side and grouping the terms by $\bar{\mathcal{V}}$ as

$$\begin{aligned}
&\ln\left[\prod_{\tilde{\mathcal{V}}\subset\mathcal{V}} \prod_{\bar{\mathcal{V}}\subset\tilde{\mathcal{V}}} \langle\{T\}\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right]\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}}\right] \\
&= \sum_{\bar{\mathcal{V}}\subset\mathcal{V}} \ln\left[\langle\{T\}\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right]\right] \left(\sum_{\tilde{\mathcal{V}}\subset\mathcal{V}\,:\,\bar{\mathcal{V}}\subset\tilde{\mathcal{V}}} (-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}\right) \\
&= \sum_{\bar{\mathcal{V}}\subset\mathcal{V}} \ln\left[\langle\{T\}\rangle\left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}\right]\right] \left(\sum_{i\in[|\mathcal{V}|-|\bar{\mathcal{V}}|]} (-1)^i \binom{|\mathcal{V}| - |\bar{\mathcal{V}}|}{i}\right)
\end{aligned}$$

Now, by the generic binomial theorem we have that for $n \in \mathbb{N}, n \neq 0$

$$0 = (1-1)^n = \sum_{i \in [n]} (-1)^i \binom{n}{i} .$$

Therefore, the summands for $\bar{\mathcal{V}} \neq \mathcal{V}$ vanish and we have

$$\begin{aligned}
&\ln \left[\prod_{\tilde{\mathcal{V}} \subset \mathcal{V}} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle \{T\} \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} \right] \\
&= \ln \left[T \left[X_{\mathcal{V}} = x_{\mathcal{V}} \right] \right] \left(\sum_{i \in [0]} (-1)^i \binom{0}{i} \right) \\
&= \ln \left[T \left[X_{\mathcal{V}} = x_{\mathcal{V}} \right] \right] .
\end{aligned}$$

Applying the exponential function on both sides establishes the claim. □

Lemma 27. *Let T be a positive tensor, $\tilde{\mathcal{V}} \subset \mathcal{V}$ and arbitrary subset and $x_{\tilde{\mathcal{V}}}$ an arbitrary index. When there are $A, B \in \tilde{\mathcal{V}}$, such that*

$$\langle \{T\} \rangle \left[X_{A,B} | X_{\mathcal{V}/\{A,B\}} \right] = \left\langle \left\{ \langle \{T\} \rangle \left[X_A | X_{\mathcal{V}/\{A,B\}} \right], \langle \{T\} \rangle \left[X_B | X_{\mathcal{V}/\{A,B\}} \right] \right\} \right\rangle \left[X_{\tilde{\mathcal{V}}} \right]$$

then

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\langle \{T\} \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} = 1 .$$

Proof. We abbreviate

$$Z_{\bar{\mathcal{V}}} = \langle \{T\} \rangle \left[X_{\mathcal{V}/\bar{\mathcal{V}}} = x_{\mathcal{V}/\bar{\mathcal{V}}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] .$$

By reorganizing the sum over $\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}$ into $\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}/A \cup B$ we have

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(Z_{\bar{\mathcal{V}}} \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} = \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}/\{A,B\}} \left(\frac{Z_{\bar{\mathcal{V}}} \cdot Z_{\bar{\mathcal{V}} \cup \{A,B\}}}{Z_{\bar{\mathcal{V}} \cup \{A\}} \cdot Z_{\bar{\mathcal{V}} \cup \{B\}}} \right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} . \tag{68}$$

From the independence assumption it follows that for any index x

$$\begin{aligned}
&\langle \{T\} \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B \right] \\
&= \langle \{T\} \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}} \right] \\
&= \langle \{T\} \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B \right]
\end{aligned}$$

Applying this in each squares bracket term of (68) we get

$$\begin{aligned}
\frac{Z_{\bar{\mathcal{V}}}}{Z_{\bar{\mathcal{V}} \cup \{A\}}} &= \frac{\langle \{T\} \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B \right]}{\langle \{T\} \rangle \left[X_A = \ y_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = x_B \right]} \\
&= \frac{\langle \{T\} \rangle \left[X_A = x_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B \right]}{\langle \{T\} \rangle \left[X_A = \ y_A | X_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}} = x_{\mathcal{V}/\bar{\mathcal{V}} \cup \{A,B\}}, X_{\bar{\mathcal{V}}} = y_{\bar{\mathcal{V}}}, X_B = y_B \right]} \\
&= \frac{Z_{\bar{\mathcal{V}} \cup \{B\}}}{Z_{\bar{\mathcal{V}} \cup \{A,B\}}} .
\end{aligned}$$

Thus, each factor in (68) is trivial, which establishes the claim. □

We are finally ready to proof Theorem 11 based on the Lemmata above.

Theorem 76 (Theorem 11)**.** *Let $\mathbb{P}\left[X_{\mathcal{V}}\right]$ be a probability distribution and $\mathcal{G}$ a clique-capturing hypergraph, such that for A, B, C we have that X_A is independent of X_B conditioned on X_C, when C separates A and B in the hypergraph. Then there is a Markov Network on $\mathcal{G}$, which distribution is equal to $\mathbb{P}\left[X_{\mathcal{V}}\right]$.*

Proof of Theorem 11. By Lemma 26 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 Lemma 27 we then have

$$\prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\mathbb{P}\left[X_{\bar{\mathcal{V}}} = x_{\bar{\mathcal{V}}}, X_{\mathcal{V}/\bar{\mathcal{V}}} = y_{\mathcal{V}/\bar{\mathcal{V}}}\right]\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}} = 1\,.$$

We label by a function

$$\alpha : \{\tilde{\mathcal{V}} : \exists e \in \mathcal{E} : \tilde{\mathcal{V}} \subset e\} \to \mathcal{E}$$

the remaining node subsets by a hyperedge containing the subset. We build the tensor

$$T^e\left[X_e\right] = \prod_{\tilde{\mathcal{V}}\,:\,\alpha(\tilde{\mathcal{V}})=e} \prod_{\bar{\mathcal{V}} \subset \tilde{\mathcal{V}}} \left(\mathbb{P}\left[X_{\bar{\mathcal{V}}} = x_{\bar{\mathcal{V}}}, X_{\mathcal{V}/\bar{\mathcal{V}}} = y_{\mathcal{V}/\bar{\mathcal{V}}}\right]\right)^{(-1)^{|\tilde{\mathcal{V}}|-|\bar{\mathcal{V}}|}}\,.$$

and get, that

$$\begin{aligned}\mathbb{P}\left[X_{\mathcal{V}}\right] &= \left\langle\{T^e \,:\, e \in \mathcal{E}\}\right\rangle\left[X_{\mathcal{V}}\right] \\ &= \left\langle\{T^e \,:\, e \in \mathcal{E}\}\right\rangle\left[X_v|\varnothing\right]\,.\end{aligned}$$

We have thus constructed a Markov Network with trivial partition function, which contraction coincides with the probability distribution. □

16.4 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 77. *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network on a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let us now split the $\mathcal{G}$ into two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, such that $\mathcal{E}_1 \dot\cup \mathcal{E}_2 = \mathcal{E}$, $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$ and all nodes in $\mathcal{V}_2$ are contained in an hyperedge of $\mathcal{E}_2$. We then have*

$$\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right] = \left\langle\mathcal{T}^{\mathcal{G}_1} \cup \{\left\langle\mathcal{T}^{\mathcal{G}_2}\right\rangle\left[X_{\mathcal{V}_2 \cap (\mathcal{V}_1 \cup \tilde{\mathcal{V}})}\right]\}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\,.$$

Proof. For any index $x_{\tilde{\mathcal{V}}}$ we show that

$$\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}\right] = \left\langle\mathcal{T}^{\mathcal{G}_1} \cup \{\left\langle\mathcal{T}^{\mathcal{G}_2}\right\rangle\left[X_{\mathcal{V}_2 \cap (\mathcal{V}_1 \cup \tilde{\mathcal{V}})}\right]\}\right\rangle\left[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}\right]\,.$$

By definition we have

$$\begin{aligned}\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}\right] &= \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}}} \prod_{e \in \mathcal{E}} T^e\left[X_e = x_e\right] \\ &= \sum_{x_{\mathcal{V}/\tilde{\mathcal{V}}}} \left(\prod_{e \in \mathcal{E}_1} T^e\left[X_e = x_e\right]\right) \cdot \left(\prod_{e \in \mathcal{E}_2} T^e\left[X_e = x_e\right]\right) \\ &= \sum_{x_{\mathcal{V}_1/\tilde{\mathcal{V}}}} \sum_{x_{\mathcal{V}_2/(\tilde{\mathcal{V}} \cup \mathcal{V}_1)}} \left(\prod_{e \in \mathcal{E}_1} T^e\left[X_e = x_e\right]\right) \cdot \left(\prod_{e \in \mathcal{E}_2} T^e\left[X_e = x_e\right]\right) \\ &= \sum_{x_{\mathcal{V}_1/\tilde{\mathcal{V}}}} \left(\prod_{e \in \mathcal{E}_1} T^e\left[X_e = x_e\right]\right) \cdot \left(\sum_{x_{\mathcal{V}_2/(\tilde{\mathcal{V}} \cup \mathcal{V}_1)}} \prod_{e \in \mathcal{E}_2} T^e\left[X_e = x_e\right]\right)\,.\end{aligned}$$

When contracting the variables $X_{\mathcal{V}_2/(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}$ on $\mathcal{T}^{\mathcal{G}_2}$, the variables $X_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}$ are left open. We therefore have for any $x_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}$

$$\left\langle\left\{\mathcal{T}^{\mathcal{G}_2}\right\}\right\rangle\left[X_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}=x_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}\right]=\left(\sum_{x_{\mathcal{V}_2/(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}}\prod_{e\in\mathcal{E}_2}T^e\left[X_e=x_e\right]\right).$$

It follows with the above, that

$$\begin{aligned}\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\tilde{\mathcal{V}}}=x_{\tilde{\mathcal{V}}}\right]&=\sum_{x_{\mathcal{V}_1/\tilde{\mathcal{V}}}}\left(\prod_{e\in\mathcal{E}_1}T^e\left[X_e=x_e\right]\right)\cdot\left\langle\left\{\mathcal{T}^{\mathcal{G}_2}\right\}\right\rangle\left[X_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}=x_{\mathcal{V}_2\cap(\tilde{\mathcal{V}}\cup\mathcal{V}_1)}\right]\\&=\left\langle\mathcal{T}^{\mathcal{G}_1}\cup\left\{\left\langle\mathcal{T}^{\mathcal{G}_2}\right\rangle\left[X_{\mathcal{V}_2\cap(\mathcal{V}_1\cup\tilde{\mathcal{V}})}\right]\right\}\right\rangle\left[X_{\tilde{\mathcal{V}}}=x_{\tilde{\mathcal{V}}}\right].\end{aligned}$$

□

16.5 Support of Contractions

To state the next theorem we introduce the nonzero function $\chi:\mathbb{R}\to[2]$ by

$$\chi\left(x\right)=\begin{cases}0 & \text{if } x=0\\ 1 & \text{else}\end{cases}\tag{69}$$

Applied coordinatewise on tensors it marks the nonzero coordinates by 1.

We show that adding binary tensor cores to an contraction orders the results by the partial ordering introduced in Definition 41

Theorem 78 (Monotonicity of Tensor Contractions)**.** *Let $\mathcal{T}^{\mathcal{G}},\mathcal{T}^{\tilde{\mathcal{G}}}$ be tensor network of non-negative tensors and $X_{\tilde{\mathcal{V}}}$ an arbitrary set of random variables. Then we have*

$$\chi\left(\left\langle\mathcal{T}^{\mathcal{G}}\cup\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\right)\prec\chi\left(\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\right).$$

Proof. It suffices to show that for any $x_{\tilde{\mathcal{V}}}$ with

$$\chi\left(\left\langle\mathcal{T}^{\mathcal{G}}\cup\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle\left[X_{\tilde{\mathcal{V}}}=x_{\tilde{\mathcal{V}}}\right]\right)=1$$

we also have

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

For any $x_{\tilde{\mathcal{V}}}$ satisfying the first equation we find an extension $x_{\mathcal{V}}$ to all variables of the tensor networks such that

$$\left\langle\mathcal{T}^{\mathcal{G}}\cup\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle\left[X_{\mathcal{V}}=x_{\mathcal{V}}\right]>0$$

and it follows that

$$\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}=x_{\mathcal{V}}\right]>0\quad\text{and}\quad\left\langle\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle\left[X_{\mathcal{V}}=x_{\mathcal{V}}\right]>0\,.$$

But this already implies, that

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

□

Let us now state an equivalence of the contraction, when we add the result of the same contraction

Theorem 79 (Invariance under adding subcontractions)**.** *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network of non-negative tensors with variables $X_{\mathcal{V}}$ and let $\mathcal{T}^{\tilde{\mathcal{G}}}$ be a subset. Then we have for any subset $X_{\tilde{\mathcal{V}}}$ of $X_{\mathcal{V}}$*

$$\left\langle\mathcal{T}^{\mathcal{G}}\cup\left\{\chi\left(\left\langle\mathcal{T}^{\tilde{\mathcal{G}}}\right\rangle\left[X_{\tilde{\mathcal{V}}}\right]\right)\right\}\right\rangle\left[X_{\mathcal{V}}\right]=\left\langle\mathcal{T}^{\mathcal{G}}\right\rangle\left[X_{\mathcal{V}}\right].$$

Proof. For any $x_{\mathcal{V}}$ with

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

we also have

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{\chi \left(\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] \right)\} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 0 \,.$$

For any $x_{\mathcal{V}}$ with

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \neq 0$$

we have for the reduction $x_{\tilde{\mathcal{V}}}$ of the index $x_{\mathcal{V}}$ that

$$\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] \neq 0$$

and thus

$$\left\langle \mathcal{T}^{\mathcal{G}} \cup \{\chi \left(\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] \right)\} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \cdot \chi \left(\left\langle \mathcal{T}^{\tilde{\mathcal{G}}} \right\rangle [X_{\tilde{\mathcal{V}}}] \right) [X_{\tilde{\mathcal{V}}} = x_{\tilde{\mathcal{V}}}] = \left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] \,.$$

□

Remark 29. *Similar statements hold, when dropping the non-negativity assumption on the, but demanding that all variables are left open.*

17 Basis Calculus

Basis Calculus stores informations in the selection of basis elements, while coordinate calculus uses the coordinates to each index for storage. While coordinate calculus is more expressive, basis calculus can be exploited in sparse representations of composed functions.

17.1 Encoding of Subsets and Relations

Based on the concept of one-hot encodings of states we in this chapter develop the construction of encodings to sets, relations and functions.

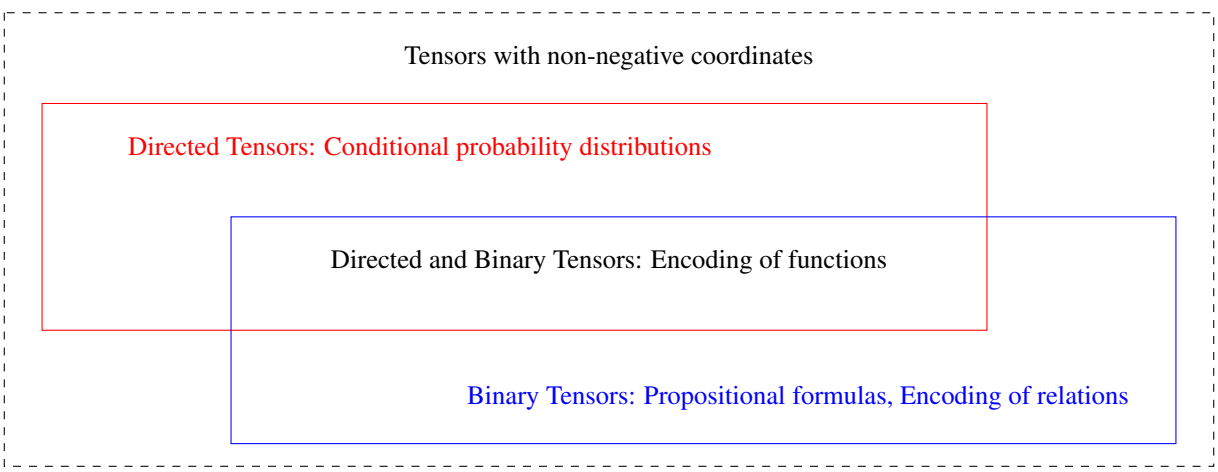


Figure 29: Sketch of the tensors with non-negative coordinates. We investigate in this chapter tensors, which are directed and binary.

Here we show how we can use binary numbers to encode the truth of set memberships.

Definition 67 (Subset Encoding)**.** *We say that an arbitrary set $\mathcal{M}$ is enumerated by a categorical variables $O_{\mathcal{M}}$ taking values in $[m_{\mathcal{M}}]$, when $m_{\mathcal{M}} = |\mathcal{M}|$ and there is a bijective function*

$$I : [m_{\mathcal{M}}] \to \mathcal{M} \,.$$

Given an set $\mathcal{M}$ enumerated by the variable $O_{\mathcal{M}}$, any subset $\mathcal{V} \subset \mathcal{M}$ is encoded by the tensor $e_{\mathcal{V}}[O]$ defined for $x \in [|\mathcal{M}|]$ as

$$e_{\mathcal{V}}[O = x] = \begin{cases} 1 & \text{if } I(x) \in \mathcal{V} \\ 0 & \text{else} \end{cases} \,. \tag{70}$$

In a one-hot basis decomposition we have

$$e_{\mathcal{V}}[O] := \sum_{x \in [|\mathcal{M}|] \,:\, I(x) \in \mathcal{V}} e_x\,[O]\ .$$

Encoding of subsets as vectors: Each coordinate associated with a possible element, $\{0, 1\}$ encoding whether in subset. The encodings is thus a binary tensor. Any subset encoding is a binary tensor.

Definition 68 (Relation Encoding)**.** *Given two finite sets $\mathcal{M}^{\text{in}}$, $\mathcal{M}^{\text{out}}$, a relation is a subset of their cartesian product*

$$\mathcal{R} \subset \mathcal{M}^{\text{in}} \times \mathcal{M}^{\text{out}}\,.$$

Given an enumeration of $\mathcal{M}^{\text{in}}$ and $\mathcal{M}^{\text{out}}$ by the categorical variables O_{in} and O_{out} and interpretation maps I^{in}, I^{out} , we define the encoding of this subset as the tensor $e_{\mathcal{R}}[O_{\text{in}}, O_{\text{out}}]$ with the coordinates

$$e_{\mathcal{R}}[O_{\text{in}} = x_{\text{in}}, O_{\text{out}} = x_{\text{out}}] = \begin{cases} 1 & \textit{if}\,(I^{\text{in}}(x_{\text{in}}), I^{\text{out}}(x_{\text{out}})) \in \mathcal{R} \\ 0 & \textit{else} \end{cases}\ . \tag{71}$$

The relation encoding has a decomposition into one-hot encodings as

$$e_{\mathcal{R}}[O_{\text{in}}, O_{\text{out}}] = \sum_{x_{\text{in}}, x_{\text{out}} \,:\, (I^{\text{in}}(x_{\text{in}}), I^{\text{out}}(x_{\text{out}})) \in \mathcal{R}} e_{x_{\text{in}}}\,[X_{\text{in}}] \otimes e_{x_{\text{out}}}\,[X_{\text{out}}]\ .$$

Relations are subsets of cartesian products and encodings of relations are the encodings of subsets by vectors. They have a matrix structure by the cartesian product, which can be further folded to tensors, when the sets itself are cartesian products.

Theorem 80. *The relational encoding is a bijection between the set of relations and the set of binary tensors.*

Proof. By definition, a relational encoding is the encoding of a subset and thus a binary tensor. Any matrification of a binary tensor marks by its 1 coordinates the elements of a relation. □

We can thus understand any matrification of a binary tensor as the encoding of a relation and vice versa.

17.1.1 Higher order relations

We can extend this contraction to relations of higher order, and arrive at encoding schemes usable for relational databases.

Definition 69. *Given sets $\mathcal{M}^k$ for $k \in [d]$, a d-ary relation is a subset of a their cartesian product, that is*

$$\mathcal{R} \subset \bigtimes_{k \in [d]} \mathcal{M}^k\,.$$

Given an enumeration of each set $\mathcal{M}^k$ by a variable O_k and an interpretation map I^k, we define the encoding of the relation as the tensor $e_{\mathcal{R}}[O_{[d]}]$ with coordinates

$$e_{\mathcal{R}}[O_0 = x_0, \ldots, O_{d-1} = x_{d-1}] = \begin{cases} 1 & \textit{if}\,(I^0(x_0), \ldots, I^{d-1}(x_{d-1})) \in \mathcal{R} \\ 0 & \textit{else} \end{cases}\ . \tag{72}$$

Example 16 (Propositional Formulas)**.** *Propositional formulas are equal to the subset encoding of their models. The sets $\mathcal{M}^k$ are all $[2]$ and are interpreted as the possible assignments to the boolean atoms.*

Example 17 (Relational Databases)**.** *Relational Databases can be encoded as tensors using the relation encoding scheme. Each column is thereby understood as a categorical variable, which values form the sets $\mathcal{M}^k$.*

Let us notice, that the dimensionality of the tensor space used for representing a relation is

$$\prod_{k \in [d]} |\mathcal{M}^k|$$

and therefore growing exponentially with the number of variables. Relations are however often sparse, in the sense that

$$|\mathcal{R}| << \prod_{k \in [d]} |\mathcal{M}^k|\ .$$

It is therefore often benefitially to choose sparse encoding schemes, for example by restricted CP formats (see Chapter 19) to represent $e_{\mathcal{R}}$.

17.2 Encoding of Functions

Real valued functions are directly tensors by definition.

17.2.1 Relational Encoding of Functions

In Definition 14 we have introduced the relational encoding of functions between the states of factored systems. We now generalize the representation scheme towards maps between arbitrary unstructured sets.

Definition 70 (Relational Encoding of Maps)**.** *Any map*

$$f : \mathcal{M}^{\text{in}} \to \mathcal{M}^{\text{out}}$$

can be represented by a relation

$$\mathcal{R}^f \coloneqq \left\{(x, f(x) \,:\, x \in \mathcal{M}^{\text{in}})\right\} \subset \mathcal{M}^{\text{in}} \times \mathcal{M}^{\text{out}} \,.$$

Given a enumeration of the sets by O_{in} and O_{out} we define the relational encoding of f as the tensor

$$\rho^f \left[O_{\text{in}}, O_{\text{out}}\right] = e_{\mathcal{R}^f} \left[O_{\mathcal{M}^{\text{in}}}, O_{\mathcal{M}^{\text{out}}}\right] \,.$$

Remark 30 (Reduction to images)**.** *When f maps into a set of infinite cardinality, we restrict $\mathcal{M}^{\text{out}}$ to the image of f and enumerate the image by a variable X_f. This scheme is applied, when f is itself a tensor, i.e. $\mathcal{M}^{\text{out}} = \mathbb{R}$. While the variable X_f can in general be of the same cardinality as the domain set $\mathcal{M}^{\text{in}}$, it will be in $[2]$ when considering binary tensors.*

We notice, that any relational representation of a function is also a directed tensor with incoming variables to the domain and outgoing variables to the image. It furthermore holds, that the set of directed and binary tensors is characterized by the relational encoding of functions. This is shown in the next theorem, by the claim that any binary tensor which is directed is the relational representation of a function.

Theorem 81. *Let $\mathcal{M}^{\text{in}}, \mathcal{M}^{\text{out}}$ be sets and $\mathcal{R} \subset \mathcal{M}^{\text{in}} \times \mathcal{M}^{\text{out}}$ a relation. If and only if there exists a map $f : \mathcal{M}^{\text{in}} \to \mathcal{M}^{\text{out}}$ such that $\mathcal{R} = \mathcal{R}^f$, the relational encoding ρ^f is a directed tensor with O_{in} incoming and O_{out} outgoing.*

Proof. When f is a function, we have for any $o_{\text{in}} \in [r_{\text{in}}]$

$$\sum_{o_{\text{out}} \in [r_{\text{out}}]} \rho^f \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}} = o_{\text{out}}\right] = \rho^f [O_{\text{in}} = o_{\text{in}}, O_{\text{out}} = f(o_{\text{in}})] = 1 \,.$$

Conversely let there be a relation $\mathcal{R}$, such that $\rho^{\mathcal{R}}$ is directed. To this end, we observe that for any $o_{\text{in}} \in [r_{\text{in}}]$ the tensor

$$e_{\mathcal{R}} \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}\right]$$

is a binary tensor with coordinate sum one and therefore a basis vector. It follows that the function $f : \mathcal{M}^{\text{in}} \to \mathcal{M}^{\text{out}}$

$$f(o_{\text{in}}) = e^{-1}(e_{\mathcal{R}} \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}\right])$$

is well-defined. We then have by construction

$$\rho^f = \sum_{o_{\text{in}} \in [r_{\text{in}}]} e_{o_{\text{in}}} \otimes e_{f(o_{\text{in}})} = \sum_{o_{\text{in}} \in [r_{\text{in}}]} e_{o_{\text{in}}} \otimes e_{\mathcal{R}} \left[O_{\text{in}} = o_{\text{in}}, O_{\text{out}}\right] = e_{\mathcal{R}}$$

and therefore $\mathcal{R} = \mathcal{R}^f$. □

We are specially interested in sets of states of a factored system, which amounts to the case in Definition 14. Those state sets have a decomposition into a cartesian product of d sets

$$\mathcal{M} = \bigtimes_{k \in [d]} [m_k] \,.$$

The most obvious enumeration of the set $\mathcal{M}$ 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 Definition 70 has an obvious generalization to this case with multiple enumeration variables.

Since the relational encoding of any map between factored systems is directed, it can be interpreted by a conditional probability tensor, as we state next.

Corollary 9. *The relational encoding ρ^f (see Definition 14) of a function f between factored systems is a conditional probability tensor, where the legs to the image system are the conditions and the legs to the target system the distribution legs.*

These are deterministic conditional probability tensors, in the sense that any slice with respect to the input variables is a basis tensor. Through contractions with distribution tensors (e.g. distributions in domain systems) they get stochastic. This is for example the case in the empirical distribution, which can be understood as the forwarding of the uniform distribution on the sample enumeration.

17.3 Calculus of relational encodings

17.3.1 Function Evaluation

Theorem 82 (Basis Calculus)**.** *Retrieving the value of the function at a specific state is then the contraction of the tensor representation with the one-hot encoded state. For any state indexed by $(x_0, \dots, x_{d-1})$ we have*

$$e_{f(x_0,\dots,x_{d-1})}\left[X_f\right] = \left\langle \{\rho^f, e_{x_0,\dots,x_{d-1}}\}\right\rangle \left[X_f\right] .$$

Thus, we can retrieve the function evaluation by the inverse one-hot mapping as

$$f(x_0, \dots, x_{d-1}) = e^{-1}(\left\langle \{\rho^f, e_{x_0,\dots,x_{d-1}}\}\right\rangle \left[X_f\right]) .$$

This generalizes Example **??** *to arbitrary maps between factored systems.*

Proof. By doing the summation in tensor product notation. □

We can thus use tensor contractions to calculate the values of functions. Since basis vectors being the one-hot encoding of the domain system are mapped to basis vectors being the encoding of the image system, we call these contraction basis calculus.

17.3.2 Composition of function

We have already used, that combination of propositional formulas by connectives can be represented by contractions. In a more general perspective, any composition of functions between factored systems is in its relational encoding the contraction of the encoded functions.

Theorem 83 (Composition of Functions)**.** *Let there be two maps between factored systems*

$$f \,:\, \bigtimes_{v\in\mathcal{V}_1} [m_v] \to \bigtimes_{v\in\mathcal{V}_2} [m_v]$$

and

$$g : \bigtimes_{v\in\mathcal{V}_2} [m_v] \to \bigtimes_{v\in\mathcal{V}_3} [m_v]$$

with the image system of f is the domain system of g. Then the relational encoding of the composition

$$g(f) : \bigtimes_{v\in\mathcal{V}_1} [m_v] \to \bigtimes_{v\in\mathcal{V}_3} [m_v]$$

satisfies

$$\rho^{g(f)} = \left\langle \{\rho^f, \rho^g\}\right\rangle \left[\mathcal{V}_1 \cup \mathcal{V}_3\right] .$$

Proof. By definition we have

$$\rho^{g(f)} = \sum_{i_1 \in \bigtimes_{v\in\mathcal{V}_1}[m_v]} e_{i_1} \otimes e_{g(f(i_1))} ,$$

where by i_1 we denote in a slide abuse of notation the tuple indexing the state of the domain factored systems of f. On the other side, we have that

$$\begin{aligned}
\langle\{\rho^f,\rho^g\}\rangle\,[X_{\mathcal{V}_1}X_{\mathcal{V}_3}] &= \sum_{\tilde{i}_2\in\times_{v\in\mathcal{V}_2}[m_v]}\left(\sum_{i_1\in\times_{v\in\mathcal{V}_1}[m_v]} e_{i_1}\cdot\left(e_{f(i_1)}\right)_{\tilde{i}_2}\right)\left(\sum_{i_2\in\times_{v\in\mathcal{V}_2}[m_v]}(e_{i_2})_{\tilde{i}_2}\cdot e_{g(i_2)}\right)\\
&= \sum_{\tilde{i}_2\in\times_{v\in\mathcal{V}_2}[m_v]}\left(\sum_{i_1\in\times_{v\in\mathcal{V}_1}[m_v]} e_{i_1}\cdot\delta_{f(i_1),\tilde{i}_2}\right)\left(\sum_{i_2\in\times_{v\in\mathcal{V}_2}[m_v]}\delta_{i_2\tilde{i}_2}\cdot e_{g(i_2)}\right)\\
&= \sum_{i_1\in\times_{v\in\mathcal{V}_1}[m_v]}\sum_{\tilde{i}_2\in\times_{v\in\mathcal{V}_2}[m_v]}\delta_{f(i_1),\tilde{i}_2}\cdot\left(e_{i_1}\otimes e_{g(i_2)}\right)\\
&= \sum_{i_1\in\times_{v\in\mathcal{V}_1}[m_v]} e_{i_1}\otimes e_{g(f(i_1))}\,.
\end{aligned}$$

Here we represented the contraction of the variables in $\mathcal{V}_2$ by the summation over another index i_2. In the last equation we used that the delta tensor does not vanish only for $\tilde{i}_2 = f(i_1)$. The claim follows, since both expressions are equal. □

We can use Theorem 83 iteratively to further decompose the function g. In this way, the relational encoding of a function consistent of multiple compositions can be represented as the contractions of all the functions. The relational encoding of propositional formulas for instance can in this way be represented as a contraction of the encodings of its logical connectives applied on the respective formula spaces.

17.3.3 Compositions with real functions

Follows from composition calculus above with the usage of

$$T = \langle\{\rho^T, Id|_{\mathrm{im}(T)}\}\rangle\,[X_{[d]}]\ .$$

We here investigate how the composition of a tensor

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

with arbitrary functions

$$h:\mathbb{R}\to\mathbb{R}$$

can be represented. This is for example relevant, when representing the distributions of an exponential family.

Our main strategy is in understanding the tensor T as a map to its finite image, seen as the enumerated states of a categorical variable building a factored system. We then use the relational encoding ρ^T of this map between factored systems.

By $h|_{\mathcal{M}}$ we further denote the restriction of a real function h to an enumerated set $\mathcal{M} = \{x_i\ :\ i\in[|\mathcal{M}|]\}\subset\mathbb{R}$, i.e. the vector

$$h|_{\mathcal{M}}:[|\mathcal{M}|]\to\ \mathbb{R}$$

defined for $i\in[|\mathcal{M}|]$ as

$$h|_{\mathcal{M}}(i) = h(x_i)\,.$$

Theorem 84. *We have for any tensor T and real function h (see Figure 30)*

$$h\circ T = \langle\{\rho^T, h|_{\mathrm{im}(T)}\}\rangle\,[X_{[d]}]\ .$$

Proof. We enumerate the image $\mathrm{im}(T)$ by $\{x_i\ :\ i\in[|\mathrm{im}(T)|]\}$. For arbitrary but fixed $x_0,\dots,x_{d-1}\in\bigtimes_{k\in[d]}[m_k]$ let $\tilde{i}$ be such that $T(x_0,\dots,x_{d-1}) = x_i$. Then we have for X_T denoting the image variable of ρ^T that

$$\langle\{\rho^T, e_{x_0,\dots,x_{d-1}}\}\rangle\,[\{X_T\}] = e_{\tilde{i}}$$

and

$$\langle\{\rho^T, h|_{\mathrm{im}(T)}, e_{x_0,\dots,x_{d-1}}\}\rangle\,[\varnothing] = \langle\{h|_{\mathrm{im}(T)}, e_{\tilde{i}}\}\rangle\,[\varnothing] = h(T(x_0,\dots,x_{d-1}))\,.$$

Since $x_0,\dots,x_{d-1}$ was chosen arbitrarly form $\bigtimes_{k\in[d]}[m_k]$, this shows that $h\circ T$ and $\langle\{\rho^T, h|_{\mathrm{im}(T)}\}\rangle\,[X_{[d]}]$ coincide on all inputs and are thus equal. □

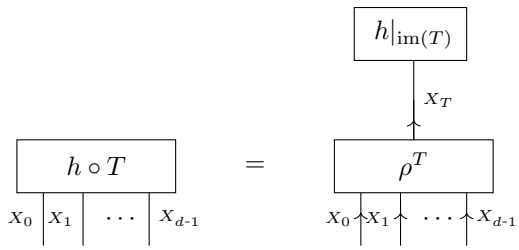


Figure 30: Representation of the composition of a tensor T with a real function h.

Corollary 10. *For any tensor $T\left[X_{[d]}\right]$ we have*

$$T\left[X_{[d]}\right] = \left\langle \rho^T, \mathrm{Id}|_{\mathrm{im}(T)} \right\rangle \left[X_{[d]}\right] .$$

Proof. Directly by using $h = \mathrm{Id}$. □

Corollary 11. *For any tensor T, which is directed with $X_{[d]}$ incoming, we have*

$$\mathbb{I}\left[X_{[d]}\right] = \left\langle \rho^T \right\rangle \left[X_{[d]}\right] .$$

Proof. Directly by using $h = \mathbb{I}$. □

Corollary 12. *Let U be a directed and binary tensor with incoming variables being $X_{[d]}$, and T a tensor, which variables are the outgoing variables of U. Let further $h : \mathbb{R} \to \mathbb{R}$ be any real function. Then*

$$h \circ \langle \{U, T\} \rangle \left[X_{[d]}\right] = \langle \{U, h \circ T\} \rangle \left[X_{[d]}\right] .$$

Proof. Since U is a directed and binary tensor, we find a map

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

such that $U = \rho^f$ and a map V such that $T = V|_{\mathrm{im}(f)}$. Then Theorem 84 implies that

$$\langle \{U, T\} \rangle \left[X_{[d]}\right] = V \circ f .$$

It follows that

$$h \circ \langle \{U, T\} \rangle \left[X_{[d]}\right] = h \circ V \circ f$$

and by another application of Theorem 84 that

$$\begin{aligned} h \circ V \circ f &= \left\langle \rho^f, h \circ V|_{\mathrm{im}(f)} \right\rangle \left[X_{[d]}\right] \\ &= \langle \{U, h \circ T\} \rangle \left[X_{[d]}\right] . \end{aligned}$$

The claim follows as a combination of both equations. □

Example 18 (Shannon entropy of empirical distribution)**.** *The Shannon entropy of an empirical distribution can be efficiently computed by contraction of the datatensor with itself along the atom indices, then applying a coordinatewise* $\ln$ *and averaging.*

This follows from commutations of contraction and coordinatewise contraction (see Corollary 12), using that the datacores are directed. To be more precise, let $\tilde{D}$ be a copy of D with identical image space and copied domain space. Then $\rho^{\tilde{D}_k}$ are tensor cores with identical outgoing legs to ρ^{D_k}, but different incoming legs. We have that

$$\begin{aligned} \mathbb{H}\left[\mathbb{P}^D\right] &= \left\langle \{\rho^{D_k} : k \in [d]\} \cup \{\frac{1}{m}\mathbb{I}\} \cup \{-\ln \left\langle \{\rho^{\tilde{D}_k} : k \in [d]\} \cup \{\frac{1}{m}\mathbb{I}\} \right\rangle \left[X_{[d]}\right]\} \right\rangle [\varnothing] \\ &= \left\langle \left\{ \frac{1}{m}\mathbb{I}, -\ln \left\langle \{\rho^{D_k}, \rho^{\tilde{D}_k} : k \in [d]\} \cup \{\frac{1}{m}\mathbb{I}\} \right\rangle \left[X_{[d]}\right] \right\} \right\rangle [\varnothing] \end{aligned}$$

where in the second equation we used Corollary 12.

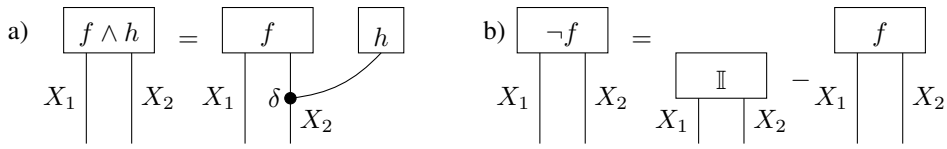


Figure 31: Decomposition schemes by effecitve calculus. a) Conjunction, b) Negations.

17.3.4 Decomposition in case of structured images

Here the introduction of multiple head variables, i.e. when

$$\mathcal{M}^{\text{out}} = \bigtimes_{k \in [d]} \mathcal{M}^k$$

This is the case for the empirical distributions!

When the image admits a cartesian representation, the relational encoding can be represented by a contraction of relational encodings to each image coordinate.

Theorem 85. *Let f be a function between factored systems*

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

and denote by

$$f^k : [m] \to [m_k]$$

the restrictions of f to axes of $\bigtimes_{k \in [d]} [m_k]$. We assign the variable X to the factored system in the domain system of f and the variables X_k for $k \in [d]$ to the image system of f.

We then have

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

Proof. We have

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

since for any $x \in [m]$

$$\rho^f \left[X = x, X_{[d]}\right] = \bigotimes_{k \in [d]} \rho^{f^k} [X = x, X_k] = \left\langle \{\rho^{f^k} [X, X_k] : k \in [d]\} \right\rangle \left[X = x, X_{[d]}\right] .$$

To get a representation in the basis CP format, we rename the variable X in the cores $\rho^{f^k} [X, X_k]$ by J and observe that for a trivial scalar core $\mathbb{I}[J]$

$$\left\langle \{\rho^{f^k} [X, X_k] : k \in [d]\} \right\rangle \left[X, X_{[d]}\right] = \left\langle \{\mathbb{I}[J]\} \cup \{\rho^{f^k} [J, X_k] : k \in [d]\} \cup \{\delta [X, J]\} \right\rangle \left[X, X_{[d]}\right] .$$

□

17.4 Effective Calculus

For specific functions, slices of the relational encodings with respect to head variables are basis vectors. In that case, we can perform basis calculus in the inverse direction than suggested by the directions of the tensors. We examplify this situation in the following theorem for relational encodings of logical conjunctions and negations.

Theorem 86. *For any formulas f, h we have*

$$\langle \{\rho^{\wedge} [X_{f \wedge h}, X_f, X_h], e_1 [X_{f \wedge h}]\} \rangle [X_f, X_h] = e_1 [X_f] \otimes e_1 [X_h] .$$

In particular, it holds that (see Figure 31a)

$$(f \wedge h)[X_{[d]}] = \langle \{f, h\} \rangle \left[X_{[d]}\right] .$$

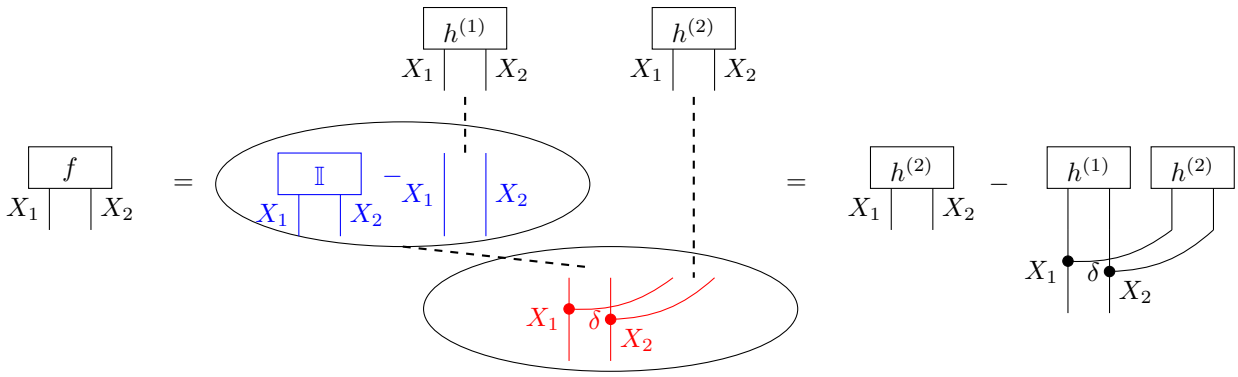


Figure 32: Example of a decomposition by effective calculus of a formula $f(X_1, X_2) = \neg h^{(1)}(X_1, X_2) \wedge h^{(2)}(X_1, X_2)$ into a sum of contractions.

Proof. We decompose

$$\rho^{\wedge}\left[X_{f\wedge h}, X_f, X_h\right] = e_1\left[X_{f\wedge h}\right] \otimes e_1\left[X_f\right] \otimes e_1\left[X_h\right] + e_0\left[X_{f\wedge h}\right]\left(\mathbb{I}\left[X_f, \mathbb{I}\left[X_h\right]\right] - e_1\left[X_f\right] \otimes e_1\left[X_h\right]\right)$$

and get the first claim as

$$\begin{aligned}\left\langle\left\{\rho^{\wedge}\left[X_{f\wedge h}, X_f, X_h\right], e_1\left[X_{f\wedge h}\right]\right\}\right\rangle\left[X_f, X_h\right] &= \left\langle\left\{e_1\left[X_{f\wedge h}\right] \otimes e_1\left[X_f\right] \otimes e_1\left[X_h\right], e_1\left[X_{f\wedge h}\right]\right\}\right\rangle\left[X_f, X_h\right] \\ &= e_1\left[X_f\right] \otimes e_1\left[X_h\right] .\end{aligned}$$

To show the second claim we use

$$\begin{aligned}(f \wedge h)[X_{[d]}] &= \left\langle\left\{\rho^f\left[X_f, X_{[d]}\right], \rho^h\left[X_h, X_{[d]}\right], \rho^{\wedge}\left[X_{f\wedge h}, X_f, X_h\right], e_1\left[X_{f\wedge h}\right]\right\}\right\rangle\left[X_{[d]}\right] \\ &= \left\langle\left\{\rho^f\left[X_f, X_{[d]}\right], \rho^h\left[X_h, X_{[d]}\right], \left(e_1\left[X_f\right] \otimes e_1\left[X_h\right]\right)\right\}\right\rangle\left[X_{[d]}\right] \\ &= \left\langle\left\{f, h\right\}\right\rangle\left[X_{[d]}\right] .\end{aligned}$$

□

A similar decomposition holds for negations, as we show next.

Theorem 87. *For any formula f we have*

$$\left\langle\left\{\rho^{\neg}\left[X_f, X_{\neg f}\right], e_1\left[X_{\neg f}\right]\right\}\right\rangle\left[X_f\right] = e_0\left[X_f\right] = \mathbb{I}\left[X_f\right] - e_1\left[X_f\right] .$$

and

$$\left\langle\left\{\rho^{\neg}\left[X_f, X_{\neg f}\right], e_0\left[X_{\neg f}\right]\right\}\right\rangle\left[X_f\right] = e_1\left[X_f\right] .$$

In particular, it holds that (see Figure 31b)

$$(\neg f)[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - f\left[X_{[d]}\right] .$$

Proof. Using that for two dimensional variables we have $\mathbb{I}\left[X\right] = e_0\left[X\right] + e_1\left[X\right]$. □

These theorems provide a mean to represent logical formulas by sums of one-hot encodings. Since any propositional formula can be represented by compositions of negations and conjunctions, they are universal. We further notice, that the resulting decomposition is a basis+ CP format, as further discussed in Chapter 19. In Figure 32 we provide an example of this decomposition.

17.5 Applications in Machine Learning

The neural paradigm of Machine Learning describes the relevance of sparse function to be effective models in the sense of learning and approximation.

Our model of the neural paradigm are tensor network decompositions, seen as decomposition of functions into smaller functions, which take each other as input. Summations along input axis are avoided, when having directed and binary tensor networks with basis calculus interpretation.

We have already observed in Theorem 82, that the value of discrete maps can be calculated by contractions of the directed binary relation encodings. This has been framed as Basis Calculus. What is more, tensor network decompositions into directed binary tensors correspond with representation of functions as compositions of smaller functions. We can understand each composition as marking a neuron in an architecture and thus have established a neural perspective on binary directed tensor networks.

18 Contraction Message Passing

In this chapter we introduce local contraction passed along tensor clusters to approximatively calculate global contractions. These message passing schemes provide tradeoffs between efficiency increases and exactness of the global contraction. We use the CP Decompositions to investigate the asymptotic behavior of the message passing algorithms.

18.1 Exact Contractions

We apply Theorem 77 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.

18.1.1 Construction of Cluster Graphs

Definition 71 (Cluster Graph)**.** *Given a tensor network $\mathcal{T}^{\mathcal{G}}$ a cluster partition is a partition of the tensor network into n clusters, by a function*

$$\alpha : \mathcal{E} \to [n] \,.$$

The clusters are with tensors decorated edge sets $C_i = \{e \,:\, \alpha(e) = i\}$ with variables $\mathcal{V}_i = \bigcup_{e \in C_i} e$. The clusters form a graph where edges between C_i and C_j exist, when the node sets $\mathcal{V}_i$ and $\mathcal{V}_j$ are not disjoint. In this case, we define separation sets $S_{i,j} = C_i \cup C_j$

Theorem 88. *Given a tensor network $\mathcal{T}^{\mathcal{G}}$ and a cluster graph. We then define for each cluster the node set*

$$\tilde{\mathcal{V}}_i = \bigcup_{j \neq i} \mathcal{V}_j$$

and have

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\tilde{\mathcal{V}}}] = \left\langle \{ \left\langle \mathcal{T}^{C_i} \right\rangle \left[\mathcal{V}_i \cap (\tilde{\mathcal{V}}_i \cup \tilde{\mathcal{V}}) \right] : i \in [n] \} \right\rangle [X_{\tilde{\mathcal{V}}}] \,.$$

Proof. By Theorem 77 applied for each cluster seen as a subgraph. □

18.1.2 Message Passing to calculate contractions

Having a hypergraph $\mathcal{G}$, we iteratively apply Theorem 77 and call the $\mathcal{G}_2$ a cluster. When iterating until $\mathcal{G}$ is empty, we get a cluster graph, where all tensors are assigned to a cluster.

When the cluster are a polytree, that is a union of disjoint trees, we define messages between neighbored clusters C_i and C_j with $C_j \prec C_i$ by the contractions

$$\delta_{j \to i} = \left\langle \{ \delta_{\tilde{j} \to j} \,:\, C_{\tilde{j}} \prec C_j \} \cup \mathcal{T}^{C_j} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] \,. \tag{73}$$

and

$$\delta_{j \leftarrow i} = \left\langle \{ \delta_{i \leftarrow \tilde{j}} \,:\, C_i \prec C_{\tilde{j}} \} \cup \mathcal{T}^{C_i} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] \,. \tag{74}$$

We note, that the messages are well defined by these recursive equations, exactly when the cluster graph is a polytree.

Lemma 28. *When the cluster graph is a tree, we have for neighbored clusters C_i and C_j with $C_j \prec C_i$*

$$\delta_{j \to i} = \left\langle \{ \mathcal{T}^{C_{\tilde{j}}} \,:\, C_{\tilde{j}} \prec C_j \} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right]$$

and

$$\delta_{j \leftarrow i} = \left\langle \{ \mathcal{T}^{C_{\tilde{j}}} \,:\, C_i \prec C_{\tilde{j}} \} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j} \right] \,.$$

Proof. By induction over the cardinality of the preceding clusters.

$n = 1$: Only a single cluster before, therefore trivial.

$n+1 \to n$: Assuming the statement holds for up to n preceding clusters, let there be $n+1$ preceding clusters. Then Theorem 77 splits contractions into terms, which are by inductive assumption the messages. □

Theorem 89. *When the cluster graph is a tree, then we have for each cluster C_i with neighbors $N(i)$*

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\mathcal{V}_i] = \left\langle \{\delta_{j\to i} \,:\, j \in N(i),\, C_j \prec C_i\} \cup \{\delta_{i\leftarrow j} \,:\, j \in N(i),\, C_i \prec C_j\} \cup \mathcal{T}^{C_i} \right\rangle [\mathcal{V}_i] \,. \tag{75}$$

Proof. By Theorem 77 we split into contractions of the clusters up and down of the respective neighbors and apply the above lemma. □

18.1.3 Variable Elimination Cluster Graphs

Remark 31 (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,*

18.1.4 Bethe Cluster Graphs

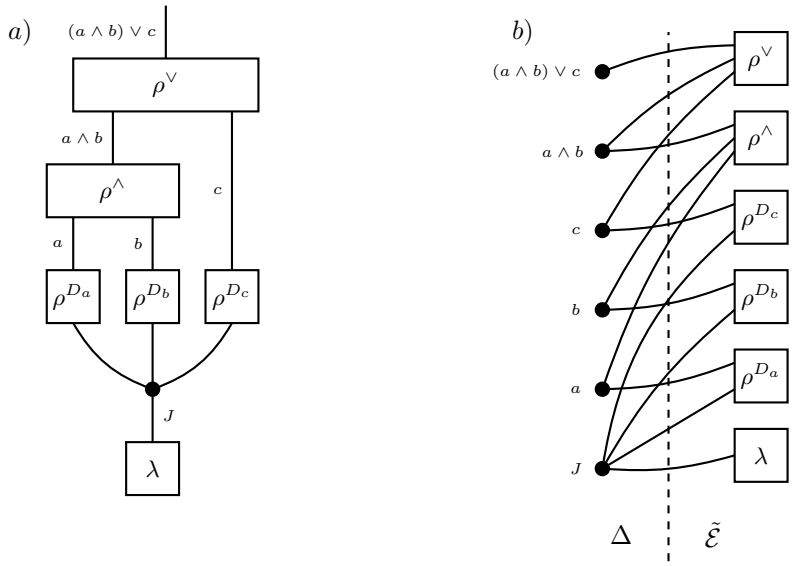


Figure 33: Example of a Bethe Cluster Graph. a) Example of a Tensor Network $\mathcal{T}^{\mathcal{G}}$, which represents the by λ averaged evaluation of the formula $(a \wedge b) \vee c$ on data D. b) Corresponding Bethe Cluster Hypergraph, which dual is bipartite by the sets Δ and $\tilde{\mathcal{E}}$.

By adding delta tensors to each node $v \in \mathcal{V}$ and defining its leg variables by v^e for $e \in \mathcal{E}$. We mark each such delta tensor by a cluster in $\Delta^{\mathcal{G}}$, as defined in the following (see also Figure 33).

Definition 72. *Given a tensor network $\mathcal{T}^{\mathcal{G}}$ on a decorated hypergraph $\mathcal{G}$, we define the Bethe Cluster Hypergraph $\tilde{\mathcal{G}}$ as $(\tilde{\mathcal{V}}, \tilde{\mathcal{E}} \cup \Delta^{\mathcal{G}})$ where we have*

- *Recolored Edges $\tilde{\mathcal{E}} = \{\tilde{e} \,:\, e \in \mathcal{E}\}$ where $\tilde{e} = \{v^e \,:\, v \in e\}$, which decoration tensor has same coordinates as T^e*
- *Nodes $\tilde{\mathcal{V}} = \bigcup_{e\in\mathcal{E}} \tilde{e}$*
- *Delta Edges $\Delta^{\mathcal{G}} = \big\{\{v^e \,:\, e \ni v\} \,:\, v \in \mathcal{V}\big\}$, each of which decorated by a delta tensor $\delta^{\{v^e \,:\, e\ni v\}}$*

By Lemma 25 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).

18.1.5 Computational Complexity

Tree-width here.

Naive execution of $\langle \mathcal{T}^{\mathcal{G}} \rangle \left[\tilde{\mathcal{V}}\right]$: $\prod_{v \in \mathcal{V}} m_v$ many products are built and summed up. When splitting contractions into local subcontractions, the product can be turned into sums with tremendous decrease in complexity.

18.2 Approximate Contractions

We ignore that cluster graphs are not trees and perform contraction message passing along neighbored clusters. For the contraction of basis tensor networks, this scheme still provides the exact contraction.

18.2.1 Exact Message Passing for Directed and Binary Contractions

A Tensor Network of directed and binary cores represents the evaluation of composed functions. In a Message Passing Perspective each component (let us call them neurons) can be evaluated, when the evaluation of the ancestor neurons are known.

Lemma 29. *Required? Basis vector factorization suffices? For any collection of categorical variables $X_{[d]}$ with identical dimension and any $x_0 \in [m_0]$ we have*

$$\langle \{\delta[X_0, \ldots, X_{d-1}]\} \rangle [X_0 = x_0, X_1, \ldots, X_{d-1}] = \bigotimes_{k \in \{1, \ldots, d-1\}} e_{x_0}[X_k] \ .$$

Proof. Directly by sum decomposition of delta tensors. □

Theorem 90. *Let $\mathcal{T}^{\mathcal{G}}$ be a tensor network on a directed acyclic hypergraph $\mathcal{G}$, such that each tensor is Boolean and directed, and such that each variable is appearing only once as an outgoing variable of a hyperedge. We build a cluster graph by storing each edge as a cluster and use the topological order on $\mathcal{G}$. Then*

$$\delta_{j \to i} = \langle \mathcal{T}^{\mathcal{G}} \rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j}\right] \ .$$

Proof. Lemma above needed? By using that each message is a basis vector (Using Theorem 73 in an induction argument) and can thus be splitted into the product of multiple copies.

Any hypercore, which is not a precessor to T^{e_i} can be omitted from the contraction by a root-stripping argument using its directionality. Therefore we have

$$\langle \mathcal{T}^{\mathcal{G}} \rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j}\right] = \left\langle \{T^{e_{\tilde{j}}} \ : \ e_{\tilde{j}} \prec e_i\} \right\rangle \left[X_{\mathcal{V}_i \cap \mathcal{V}_j}\right] \ .$$

We then replace each variable which is appearing more than once in outgoing legs by a delta tensor, which does not change the contraction by Lemma 25.

We then follow a leaf stripping argument and apply Theorem 77 iteratively on the remaining leaves. Along that line, the leave and its successors are contracted. The contraction is a basis vector and can therefore be represented as an outer product of basis vectors.

□

When replacing e_{x_0} by an arbitrary vector in Lemma 29 we have

$$\langle \{\delta[X_0, \ldots, X_{d-1}], V[X_0]\} \rangle [X_1, \ldots, X_{d-1}] \neq \bigotimes_{k \in \{1, \ldots, d-1\}} V[X_k] \ .$$

Therefore, the messages will in general differ from the exact contractions. To provide intuition of what happens in this case, let us take the following cases into account:

- $\lambda \cdot e_x$ sent multiple times: Result gets a factor of $\lambda^{\#\text{copies}}$ compared with the exact contraction.
- $e_x + e_{\tilde{x}}$: Result is the exact contraction added by the crossterms of sending e_x in one and $e_{\tilde{x}}$ in the other direction.

18.2.2 Case of Matrices

Here a toy example of cycling messages starting with $\mathbb{I}$. When normating the messages, the maximal singular vectors will be dominant.

We investigate the Bethe message passing for a tensor network consisting of a single matrix.

Theorem 91. *The stable messages are the linear subspace of the maximal singular values of the fixed core T.*

Proof. Having a Singular Value Decomposition of T and decompose the messages in the orthonormal system of the respective singular vectors. □

For the propagation of a and b on binary $f(a, b)$ starting with trivial messages of a and b the above theorem implies:

- Case of single possible world: Exact $f(a, b) \in \{a \wedge b, a \wedge \neg b, \neg a \wedge b, \neg a \wedge \neg b\}$ Messages are after first iteration exact
- Case of two possible worlds and $f(a, b) \in \{a \Leftrightarrow b, a \Leftrightarrow \neg b, \neg a \Leftrightarrow b, \neg a \Leftrightarrow \neg b\}$. In this situation any start message is stable and determines the other.
- Case of two possible worlds and $f(a, b) \in \{a, b, \neg a, \neg b\}$. In this situation one stable message is determined by the specified atom and the other is always stable.
- Case of three possible worlds: Approximative (exact: $1/3, 2/3$, approximative: $1-goldenratio, goldenratio$
- Case of four possible worlds: Exact ($\mathbb{I}$)

18.2.3 Case of Tensors

Let there now be a single tensor of arbitrary order. When the tensor is not ODECO, we cannot find a CP-Decomposition with leg vectors building an orthonormal system in the respective leg spaces. This prohibits direct application of the same techniques in the case of a matrix, which is always ODECO.

18.3 Basis Calculus

Message Passing of directed and binary message by relational encoding of functions can be interpreted as function evaluation. This is because any relational encoding of a function, the decomposition

$$\rho^f = \sum_{y \in \operatorname{im}(f)} \Big(\sum_{i : f(i) = y} e_i \Big) \otimes e_y$$

is a SVD of the matrification of ρ^f with respect to incoming and outgoing legs.

Passing a message e_i in direction thus gives the message $e_{f(i)}$.

Remark 32 (Basis Calculus as Message Passing)**.** *Given a tensor network of directed and binary tensor cores T^e, each representing a function f^e. When there are not directed cycles, we define the compositions of f^e to be the function f from the nodes $\mathcal{V}_1$ not appearing as incoming nodes to the nodes $\mathcal{V}_2$ not appearing as outgoing nodes in an edge. Choosing arbitrary $x_v \in [m_v]$ for $v \in \mathcal{V}_1$ we have*

$$\langle \{T^e \ : e \in \mathcal{E}\} \rangle \, [\mathcal{V}_2] = e_{f(x_v \, : \, v \in \mathcal{V}_1)} \, .$$

18.4 Applications

When queries share same parts, can perform their contraction using dynamic programming. For conditional probability queries, which variables are the clusters of a cluster tree, this results in belief propagation.

19 Sparse Tensor Representations

We in this chapter investigate, which sparsity notations enable tensors to be representable as contractions of tensor networks.

19.1 CP Formats

The CP Decomposition is one way to generalize the ranks of matrices to tensors. It is oriented on the Singular Value Decomposition of matrices, providing a representation of the matrix as a weighed sum of the tensor product of singular vectors. Given a tensor of higher order, each such tensor product is over multiple vectors,

Definition 73. *A CP Decomposition of rank p of a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ is a collections of tensors $\sigma[J]$ and $V^k[J, X_k]$ for $k \in [d]$, where J takes values in $[p]$, such that*

$$T\left[X_{[d]}\right] = \left\langle \{\sigma[J]\} \cup \{V^k[J, X_k] \,:\, k \in [d]\} \right\rangle \left[X_{[d]}\right] \,.$$

We say that the CP Decomposition is

- *directed, when for each k the core V^k is directed with J incoming and X_k outgoing.*
- *binary, when for each k the core V^k is binary.*
- *basis, where we demand both properties, that is for each $k \in [d]$ and $l \in [p]$*

$$V^k[J = l, X_k] \in \{\, e_{[x_k]}\,[X_k]\, x_k \in [m_k]\} \,.$$

- *basis+, when for each $k \in [d]$ and $l \in [p]$*

$$V^k[J = l, X_k] \in \{\, e_{[x_k]}\,[X_k]\, x_k \in [m_k]\} \cup \{\mathbb{I}\,[X_k]\} \,.$$

We denote by $\operatorname{rank}(T)$*, respectively* $\operatorname{rank}^{\text{bin}}(T)$, $\operatorname{rank}^{\text{bas}}(T)$ *and* $\operatorname{rank}^{\text{bas+}}(T)$ *the minimal cardinality such that T has a CP Decomposition with directed cores, respectively binary cores, basis cores and basis+ cores.*

We have by definition

$$T\left[X_{[d]}\right] = \sum_{l\in[p]} \sigma[J = l] \left(\bigotimes_{k\in[d]} V^k[J = l, X_k] \right) \,.$$

The right side can be seen as an alternative definition of CP Decompositions by summations of elementary tensors.

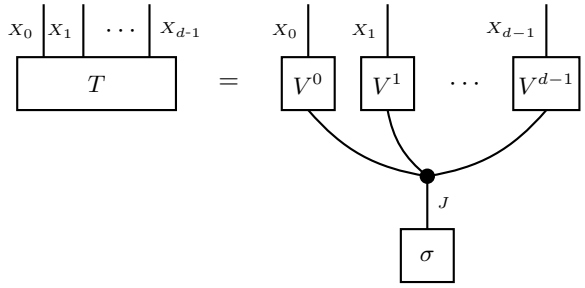


Figure 34: Tensor Network diagram of a generic CP decomposition (see Definition 73)

We introduce different notions of sparsities based on CP Decomposition with different properties of their leg cores.

19.1.1 Directed Leg Cores

This is the canonical CP Decomposition, where the vectors $V^{k,l}$ are interpreted as generalized singular vectors. Any CP decomposition can be transformed into a directed CP decomposition without enlarging the index set J, simply by diving the vectors by their norms and multiplying it to $\sigma[J = l]$.

We then have a partially directed Tensor Network representing the decomposed tensor. The only undirected core is σ, since we do not demand it to be normed. In many applications applications, however, also the σ is directed with a single outgoing leg (see for example the empirical distributions as discussed in Section 3.8). In that case, also the decomposed tensor is directed with outgoing legs.

19.1.2 Basis Leg Cores

Directed and binary leg cores have incoming slices being basis vectors, we thus call them basis CP Decomposition. This allows the interpretation of the directed and binary CP decomposition in terms of mapping to nonzero coordinates. We start by defining the number of nonzero coordinates of tensors by the ℓ_0-norm.

Definition 74. *The ℓ_0-norm counts the nonzero coordinates of a tensor by*

$$\ell_0(T) = \#\left\{x_0, \ldots, x_{d-1} : T_{x_0,\ldots,x_{d-1}} \neq 0\right\}.$$

The ℓ_0-norm is not a proper norm itself, but the limit of ℓ_p-norms (where $p \to 0$) of the flattened tensor (which are norms for $p \geq 1$).

The ℓ_0 norm is the number of nonzero coordinates. We understand the leg cores as the relational encoding of functions mapping to the slices of these coordinates given an enumeration. This is consistent with the previous analysis of Chapter 15, where we characterized binary and directed cores by the encoding of associated functions. Based on this idea, we can proof, that any tensor has a directed and binary CP decomposition with rand $\ell_0(T)$.

Theorem 92. *For any tensor T we have*

$$\operatorname{rank}^{\text{bas}}(T) = \ell_0(T).$$

Proof. We find a map

$$f : [\ell_0(T)] \to \bigtimes_{k\in[d]} [m_k],$$

which image is the set of nonzero coordinates of T. Denoting its image coordinate maps by f^k we have

$$T = \sum_{j\in[m]} \sigma[f(j)] \left(\bigotimes_{k\in[d]} e_{f^k(j)} \right).$$

This is a basis CP Decomposition with rank $\ell_0(T)$. Conversely, any basis CP Decomposition of T with dimension r would have at most r coordinates different from zero and thus $\ell_0(T) \leq r$. Thus, there cannot be a CP Decomposition with a dimension $r \leq \ell_0(T)$. □

The next theorem relates the basis CP decomposition with encodings of d-ary relations (see Definition 69).

Theorem 93. *If any only if $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ has a basis decomposition with slices $\{x^l_{[d]} : l \in [p]\}$ and trivial cores, it coincides with the encoding of the d-ary relation*

$$\mathcal{R} = \{x^l_{[d]} : l \in [p]\}.$$

If in addition $m_k = 2$, we can interpret basis CP decompositions as propositional formulas.

Theorem 94. *If $T \in \bigotimes_{k\in[d]} \mathbb{R}^2$ has a basis decomposition with slices $\{x^l_{[d]} : l \in [p]\}$ and trivial cores, it coincides with the propositional formula*

$$f\left[X_{[d]}\right] = \bigvee_{l\in[p]} Z^{\wedge}_{x^l_{[d]}},.$$

Proof. This is a generalization of Theorem 37, which follows from Theorem 23. □

The storage demand of any CP decomposition is at most linear in the dimension and the sum of its leg dimension. When we have a basis CP decomposition, this demand can be further improved. The basis vectors can be stored by its preimage of the one hot encoding $e.$, that is the number of the basis vector in $[m]$. This reduces the storage demand of each basis vector to the logarithms of the space dimension without the need of storing the full vector.

More precisely, we can store the CP Decomposition by the matrix

$$M[J, L] \in \mathbb{R}^{m\times(d+1)}$$

defined for $k \in [d]$

$$M[J = l, L = k] = e^{-1}(V^k[J = l, X_k])$$

and

$$M[J = l, L = d] = \sigma[J = l].$$

This is a typical tabular format to store relational databases.

19.1.3 Basis+ Leg Cores

The minimal rank of CP Decompositon is closely related to polynomial sparsity of the map T, which we will define next.

Definition 75. *A monomial decomposition of a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ is a set $\mathcal{M}$ of tuples (λ, A, x_A) where $\lambda \in \mathbb{R}, A \subset [d]$ and $x_A \in \bigtimes_{k\in A}[m_k]$ such that*

$$T\left[X_{[d]}\right] = \sum_{(\lambda,A,x_A)\in\mathcal{M}} \lambda \cdot \langle e_{x_A}\rangle \left[X_{[d]}\right] . \tag{76}$$

For any tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we define its polynomial sparsity as

$$\tilde{\ell}\left(T\right) = \min\left\{ |\mathcal{M}| \,:\, T\left[X_{[d]}\right] = \sum_{(\lambda,A,x_A)\in\mathcal{M}} \lambda \cdot \langle e_{x_A}\rangle \left[X_{[d]}\right] \right\}$$

We refer to the terms in a decomposition (76) in Definition 75 as monomials of binary variables, which are enumerated by pairs (k, x_k) and indicate whether the variable X_k is in state $x_k \in [m_k]$. Such indicators are represented by the one-hot encodings

$$e_{x_k}\left[X_k\right] .$$

The monomial of multiple such binary variables indicated, whether all variables labelled by a set A are in the state X_A, which is represented by

$$e_{x_A}\left[X_A\right] = \bigotimes_{k\in A} e_{x_k}\left[X_k\right] .$$

The states of the variables labeled by $k \in [d]/A$ are not specified in the monomial and the monomial is trivially extended to

$$\langle e_{x_A}\rangle \left[X_{[d]}\right] = e_{x_A}\left[X_A\right] \otimes \mathbb{I}\left[X_{[d]/A}\right] .$$

Theorem 95. *For any tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ we have*

$$\tilde{\ell}\left(T\right) = \mathrm{rank}^{\mathrm{bas+}}\left(T\right) .$$

When $x_k = 2$ for all $k \in [d]$, we also have

$$\mathrm{rank}^{\mathrm{bin}}\left(T\right) = \tilde{\ell}\left(T\right) .$$

Proof. To proof the first claim, we construct a basis+ CP decomposition given a monomial decomposition and vice versa. Let there be a tensor $T\left[X_{[d]}\right]$ with a monomial decomposition by $\mathcal{M}$ with $|\mathcal{M}| = m$ and let us enumerate the elements in $\mathcal{M}$ by $(\lambda^l, A^l, x^l_{A^l})$ for $l \in [p]$. We define for each $k \in [d]$ the tensors

$$V^k[J, X_k] = \left(\sum_{l\in[p]\,:\,k\in A} e_l\left[J\right] \otimes e_{x^l_k}\left[X_k\right] \right) + \left(\sum_{l\in[p]\,:\,k\notin A} e_l\left[J\right] \otimes \mathbb{I}\left[X_k\right] \right)$$

and

$$\sigma[J] = \sum_{l\in[p]} \lambda^l \cdot e_l\left[J\right]$$

and notice that

$$\begin{aligned} T\left[X_{[d]}\right] &= \sum_{l\in[p]} \lambda^l \cdot \left\langle e_{x^l_A} \right\rangle \left[X_{[d]}\right] \\ &= \sum_{l\in[p]} \left(\sigma[J=l] \cdot \bigotimes_{k\in[d]} V^k[J=l, X_k] \right) \\ &= \left\langle \{\sigma[J]\} \cup \{V^k[J, X_k] \,:\, k \in [d]\} \right\rangle \left[X_{[d]}\right] . \end{aligned}$$

By construction this is a basis+ CP decomposition with rank p. Since any monomial decomposition can be transformed into a basis+ CP decomposition with same rank we have

$$\tilde{\ell}\left(T\right) \geq \mathrm{rank}^{\mathrm{bas+}}\left(T\right) .$$

Let there now be a basis+ CP decomposition we define for each $l \in [p]$

$$A^l = \{k \in [d] : V^k[J = l, X_k] \neq \mathbb{I}[X_k]\} \quad \text{and} \quad x^l_A = \{e^{-1}(V^k[J = l, X_k]) : k \in A\}$$

where by $e^{-1}(\cdot)$ we denote the inverse of the one-hot encoding.

We notice that this is a monomial decomposition of $T\left[X_{[d]}\right]$ to the tuple set

$$\mathcal{M} = \{(\sigma[J = l], A^l, x^l_{A^l}) : l \in [p]\}.$$

It follows from this that

$$\tilde{\ell}(T) \leq \text{rank}^{\text{bas+}}(T)$$

and the first claim is shown.

The second claim follows from the observation, that whenever $x_k = 2$ for all $k \in [d]$ the binary CP decompositions with non-vanishing slices $V^k[J = l, X_k]$ for $k \in [d]$ and $l \in [p]$ are also basis+ CP decompositions and vice versa. □

Example 19 (Propositional Formulas)**.** *When all leg dimensions of a binary tensor T are 2, we can interpret T as a logical formula. We can use the binary CP decomposition of any tensor $\tilde{T}$ with $\chi\left(\tilde{T}\right) = T$ as a CNF of T. Finding the sparsest CNF thus amounts to finding the $\tilde{T}$ with minimal $\tilde{\ell}\left(\tilde{T}\right)$ such that $\chi\left(\tilde{T}\right) = T$.*

19.2 Representation involving selection architectures

The set of slice-sparse tensors coincides with the expressivity of specific selection architecture. We first define a slice selecting tensor and then show its decomposition into a formula selecting neural network.

Definition 76. *Given a set of atomic variables $X_{[d]}$, a slice selecting tensor of maximal cardinality n is the tensor*

$$\mathcal{H}\left[X_{[d]}, L_{0,0}, \ldots, L_{n-1,0}, L_{0,1}, \ldots, L_{n-1,1}\right]$$

with dimensions

$$p_{s,0} = 2, p_{s,1} = d$$

and coordinates

$$\begin{aligned}&\mathcal{H}\left[X_{[d]} = x_{[d]}, L_{0,0} = l_{0,0}, \ldots, L_{n-1,0} = l_{n-1,0}, L_{0,1} = l_{0,1}, \ldots, L_{n-1,1} = l_{n-1,1}\right]\\ &\quad = \begin{cases} 1 & \text{if} \quad \forall_{k,s} : \left(l_{s,1} = k \wedge l_{s,0} \neq 2\right) \Rightarrow \left(l_{s,0} = x_k\right) \\ 0 & \end{cases}.\end{aligned}$$

In the next two Lemmata we first show that the defined slice selecting tensors indeed selects slices and then provide a representation as a formula selecting network.

Lemma 30. *If all input neurons with same selection index are agreeing on the connective index, the selected formula does not vanish and coincides with a slice to the set*

$$A = \{k : \exists_{s \in [n]} : l_{s,1} = k \wedge l_{s,0} \neq 2\}$$

and for $k \in A$

$$x_k = l_{s,0} \quad \text{if} \quad l_{s,1} = k.$$

Proof. We need to show that

$$\mathcal{H}\left[X_{[d]}, L_{0,0} = l_{0,0}, \ldots, L_{n-1,0} = l_{n-1,0}, L_{0,1} = l_{0,1}, \ldots, L_{n-1,1} = l_{n-1,1}\right] = e_{x_A}\left[X_{[d]}\right]. \tag{77}$$

If and only if an index $\tilde{x}_{[d]}$ reduced on A does not coincide with x_A, we have $e_{x_A}\left[X_{[d]} = \tilde{x}_{[d]}\right] = 0$ and otherwise $e_{x_A}\left[X_{[d]} = \tilde{x}_{[d]}\right] = 1$. Let us notice, that this condition is equivalent to

$$\forall_{k,s} : \left(l_{s,1} = k \wedge l_{s,0} \neq 2\right) \Rightarrow \left(l_{s,0} = x_k\right)$$

and thus (77) holds. □

Lemma 31. *The slice selection tensor coincides with a formula selecting neural network with neurons (see Figure 19.2):*

- *unary input neuron enumerated by s, selecting one of the $X_{[d]}$ with the variable $L_{s,1}$ and selecting a connective in $\{\neg, \mathrm{Id}, \mathrm{True}\}$ by $L_{s,0}$*
- *n-ary output neuron fixed to the $\wedge$ connective.*

Proof. This can be easily checked on each coordinate. □

It follows, that the expressivity of the slice selecting neural network coincides with the set of tensors with a bound on their slice sparsity, when $n \geq d$. For arbitrary n, the following theorem holds.

Theorem 96. *Let $\mathcal{H}$ be a slice selecting tensor. For any parameter tensor θ we have*

$$\mathrm{rank}^{\mathrm{bas+}}\left(\langle \mathcal{H}, \theta \rangle \left[X_{[d]}\right]\right) \leq \mathrm{rank}^{\mathrm{bas}}\left(\theta\right) .$$

Proof. Each non-vanishing coordinate of θ represents by Lemma 31 a slice and their weighted sum is thus a monomial decomposition. □

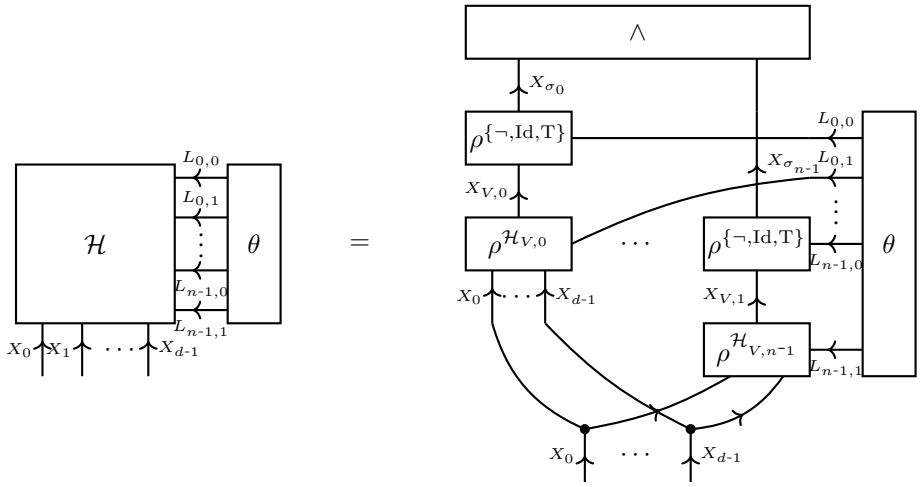


Figure 35: Representation of a basis+ Tensor by the contraction of a parameter tensor θ with a slice selecting architecture $\mathcal{H}$, which has a decomposition as a formula selecting neural network (see Lemma 31). The nonzero coordinates of θ represent the (see Lemma 30).

19.2.1 Applications

One application is as a parametrization scheme in the approximation of a tensor by a slice-sparse tensor, see Chapter 21.

The approximated parameter can then be used as a proxy energy to be maximized. When choosing $n = 2$, the approximating tensor contains only quadratic slices, which then poses a QUBO problem.

Remark 33 (Extension to arbitrary CP-formats)**.** *Select at each input neuron a specific leg. For finite number of legs, as it is the case in the binary, basis and basis+ formats, we can enumerate all possibilities by the selection variable. For the basis+ format, in case of binary leg dimensions, we here exemplified the approach, by enumerating the three possibilities $e_0, e_1, \mathbb{I}$ [1]. This approach, however, fails as a generic representation of the directed format, since the directed legs are continuous and there therefore are infinite choosable legs.*

19.3 Constructive Bounds on CP Ranks

After having defined three CP Decompositions, let us investigate bounds on their ranks which proofs come with constructions of the cores.

19.3.1 Format Transformations

Especially useful, when the leg dimensions are two, where the slice decomposition shows decomposition of the tensor into monomials.

Theorem 97. *For any tensor* $T\left[X_{[d]}\right] \in \bigotimes_{k\in[d]} \mathbb{R}^{m_k}$ *we have*

$$\operatorname{rank}(T) \leq \operatorname{rank}^{\text{bin}}(T) \leq \operatorname{rank}^{\text{bas+}}(T) \leq \operatorname{rank}^{\text{bas}}(T) \ .$$

Proof. Since any CP decomposition into binary leg cores can be normed to a CP decomposition with directed leg cores, the first bound holds. The second bound holds analogously, since any CP decomposition with basis leg cores is also a CP decomposition with binary leg cores. □

Consider for example the tensor $\mathbb{I}$ having maximal ℓ_0-norm being the dimension of the tensor space, but, since it is elementary, a CP decomposition with rank 1.

19.3.2 Summation of CP Decompositions

Theorem 98. *For any collections of tensors* $\{T^l[X_{\mathcal{V}}] : l \in [n]\}$ *with identical variables and scalars* $\lambda^l \in \mathbb{R}$ *for* $l \in [n]$ *we have*

$$\operatorname{rank}\left(\sum_{l\in[n]} \lambda^l \cdot T^l\right) \leq \sum_{l\in[n]} \operatorname{rank}\left(T^l\right) \ .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$*, by* $\operatorname{rank}^{\text{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas+}}(\cdot)$.

Proof. Products with scalars do not change the rank, since they just rescale the core σ. The sum of CP Decomposition is just the combination of all slices, thus the rank is at most additive. □

19.3.3 Contractions of CP Decompositions

More general, we can bound the sparsity of any contraction by the product of sparsities of affected tensors.

Theorem 99. *For any tensor network with variables* $\mathcal{V}$ *and edges* $\mathcal{E}$ *we have for any subset* $\tilde{\mathcal{V}} \subset \mathcal{V}$

$$\operatorname{rank}\left(\langle\{T^e : e \in \mathcal{E}\}\rangle\left[\tilde{\mathcal{V}}\right]\right) \leq \prod_{e\in\mathcal{E}\,:\,\tilde{\mathcal{V}}\cap e\neq\varnothing} \operatorname{rank}(T^e) \ .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$*, by* $\operatorname{rank}^{\text{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas+}}(\cdot)$.

Remarkably, in Theorem 99 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 32. *For any tensors* $T^1\left[X_{\mathcal{V}_1}\right]$ *and* $T^2\left[X_{\mathcal{V}_2}\right]$ *and any set of variables* $\tilde{\mathcal{V}} \subset \mathcal{V}_1 \cup \mathcal{V}_2$ *we have*

$$\operatorname{rank}\left(\langle\{T^1, T^2\}\rangle\left[\tilde{\mathcal{V}}\right]\right) \leq \operatorname{rank}\left(T^1\right) \cdot \operatorname{rank}\left(T^2\right) \ .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$*, by* $\operatorname{rank}^{\text{bas}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas+}}(\cdot)$.

Proof. By connecting the cores and restoring the binary or basis properties. □

When one core of the contracted tensor network does not contain variables which are left open, we can drastically sharpen the bound provided by Lemma 32 as we show next.

Lemma 33. *For any tensor network consistent of two tensors* $T^1\left[X_{\mathcal{V}_1}\right]$ *and* $T^2\left[X_{\mathcal{V}_2}\right]$ *and any set* $\tilde{\mathcal{V}}$ *with* $\tilde{\mathcal{V}} \cap \mathcal{V}_2 = \varnothing$ *we have*

$$\operatorname{rank}\left(\langle\{T^1, T^2\}\rangle\left[\tilde{\mathcal{V}}\right]\right) \leq \operatorname{rank}\left(T^1\right) \ .$$

The bound still holds, when we replace on both sides $\operatorname{rank}(\cdot)$ *by* $\operatorname{rank}^{\text{bin}}(\cdot)$ *or by* $\operatorname{rank}^{\text{bas}}(\cdot)$.

Proof. We show the lemma by constructing a CP decomposition of $\operatorname{rank}\left(\left\langle\{T^1, T^2\}\right\rangle\left[\tilde{\mathcal{V}}\right]\right)$ for any CP decomposition of T^1. Let therefore take any CP decomposition of T^1 consistent of the leg cores $\{V^v : v \in \mathcal{V}_1\}$ and a scalar core σ. Then we define a new σ by

$$\tilde{\sigma} = \left\langle\{\sigma\} \cup \{V^v : v \in \mathcal{V}_1, v \notin \tilde{\mathcal{V}}\} \cup \{T^2\}\right\rangle[J] .$$

Then, the leg cores $\{V^v : v \in \tilde{\mathcal{V}}\}$ build with the scalar core $\tilde{\sigma}$ a CP decomposition of $\left\langle\{T^1, T^2\}\right\rangle\left[\tilde{\mathcal{V}}\right]$. When the CP decomposition of T^1 was binary, basis or basis+, this property is also satisfied by the constructed CP decomposition. Thus the bound also holds for the ranks $\operatorname{rank}^{\text{bin}}(\cdot)$ or $\operatorname{rank}^{\text{bas}}(\cdot)$. □

Proof of Theorem 99. Use delta tensor representation to represent contractions by graphs. We then iterate through the cores and contract them to the previously contracted tensor, where we apply Lemma 32 when the tensor core has variables left open and Lemma 33 if not. □

Example 20 (Composition of formulas with connectives)**.** *For any formula f we have $1 - f = \neg f$. The CP rank bound brings an increase by at most factor 2 when taking the contraction with $\rho^{\neg}$ which has slice sparsity of 2. This is not optimal, since $\neg f$ has at most an absolute slice sparsity increase of 1.*

For any formulas f and h we have $f \cdot h = f \wedge h$. Here the CP rank bounds on contractions can also be further tightened.

Example 21 (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.*

19.3.4 Normations of CP Decompositions

As a theorem: If any of the above CP Decomposition is normable, the normation has the same CP ranks. Especially interesting when learning Bayesian Networks, where each core has a CP bound by the number of datapoints.

19.3.5 Sparse Encoding of Functions

We now state that the basis CP rank of relational encodings is equal to the cardinality of the domain. The basis CP format can therefore not provide a sparse representation when the factored system contains many categorical variables.

Theorem 100. *For any function*

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

between factored systems we have

$$\operatorname{rank}^{\text{bas}}\left(\rho^f\right) = \prod_{k \in [d]} m_k .$$

Proof. With Theorem 92, the basis CP rank coincides with the number of not vanishing coordinates, which is the cardinality of the domain of f. □

Allowing for trivial leg vectors can decrease the CP rank, as we show next.

Theorem 101. *We have*

$$\operatorname{rank}^{\text{bas+}}\left(\rho^f\right) \leq \sum_{y \in \operatorname{im}(f)} \operatorname{rank}^{\text{bas+}}\left(\mathbb{I}_{f==y}\right) ,$$

where by $\mathbb{I}_{f==y}$ we denote the indicator, whether the function f evaluates to y.

Proof. We have

$$\rho^f = \sum_{y \in \operatorname{im}(f)} \mathbb{I}_{f==y}[X] \otimes e_y\left[X_f\right] .$$

For each $y \in \operatorname{im}(f)$ we represent $e_y\left[X_f\right]$ in an basis+ CP format with $\operatorname{rank}^{\text{bas+}}\left(\mathbb{I}_{f==y}\right)$ summands and arrive at a basis+ CP decomposition of ρ^f with $\sum_{y \in \operatorname{im}(f)} \operatorname{rank}^{\text{bas+}}\left(\mathbb{I}_{f==y}\right)$ summands. □

The above claim still holds when replacing $\mathrm{rank}^{\mathrm{bas+}}(\cdot)$ with the ranks $\mathrm{rank}^{\mathrm{bas}}(\cdot)$ or $\mathrm{rank}^{\mathrm{bin}}(\cdot)$. For the rank $\mathrm{rank}^{\mathrm{bas}}(\cdot)$ it leads to the bound of Theorem 100, since summing the number of non zero coordinators of the indicators is the cardinality of the domain.

Example 22. *Conjunction of variables For the propositional formula $f = X_0 \wedge X_1$ we have*

$$\rho^f[X_0, X_1] = e_{1,1}[X_0, X_1] \otimes e_1[X_f] + (\mathbb{I}[X_0, X_1] - e_{1,1}[X_0, X_1]) \otimes e_0[X_f]$$

and thus

$$\mathrm{rank}^{\mathrm{bas+}}\left(\rho^f\right) \leq 3$$

while $\mathrm{rank}^{\mathrm{bas}}\left(\rho^f\right) = 4$. *Especially useful for d-ary conjunctions, see Remark 8!*

19.3.6 Construction by averaging the incoming legs

Basis CP Decompositions can be constructed by understanding the variable O_{in} of the relational encoding of a function $f : \mathcal{M}^{\mathrm{in}} \to \mathcal{M}^{\mathrm{out}}$ as the slice selection variable.

Example 23. *Empirical distributions, see Theorem 15 Let there be a data map*

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

We can use Theorem 85 to find a tensor network representation fo ρ^D as

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

This representation is in the CP format, when adding trivial scalar core and and delta tensor to the data index. It is furthermore in a basis CP format, since all ρ^{D^k} are directed and binary tensors. Normation to get the empirical distribution amounts to setting a slice core with coordinates $\frac{1}{m}$.

19.4 Subspaces of formulas

Formula Tensors have Tensor Network decomposition, which are best represented in a HT decomposition. We here describe this perspective and show applications of this formalism in the recovery/learning of formula tensors. The decomposition of formula tensors is basic, since atomic formula tensors being on the leafs consist of basic vectors in the respective legs.

19.4.1 Formula Subspaces

Each formula tensor defines the subspace of $\bigotimes_{k \in [d]} \mathbb{R}^2$

$$V^f = \mathrm{span}\{\neg f, f\} = \mathrm{im}\left(\rho^f\right) \tag{78}$$

Let us notice that the spanning vector of the subspace V^f are binary tensors summing up to the tensor of ones.

For each atom X_k we have

$$V^{X_k} = \mathbb{R}^2.$$

The tensor space carrying the factored representation of the worlds is thus

$$\bigotimes_{k \in [d]} V^{X_k}. \tag{79}$$

19.4.2 Formula Decomposition as a Subspace Choice

Given a formula $f \circ h$ composed of formulas f and h containing different atoms we have

$$V^{f \circ h} \subset V^f \otimes V^h \tag{80}$$

A connective $\circ$ thus determines the selection of a two-dimensional subspace in the four-dimensional tensor product of subspaces to both subformulas.

Reconstruction of a formula given its formula tensor amounts to finding the HT Decomposition under the constraints of subspace choices according to the allowed logical connectives.

Given a set of positive and negative examples of a formula poses further an approximation problem of the examples by a HT Decomposition.

Advantages of this perspective are

- Given a HT the best approximation always exists (Theorem 11.58 in Hackbusch (2012)), but need to further restrict to cores given by logical connectives
- Apply Approximation algorithms: ALS or HOSVD

20 Optimization of Sparse Tensors

Let us now study the problem of searching for the maximal coordinate in a tensor, when the tensor is represented in a basis+ CP Format.

The search for maximal coordinates appears in various reasoning tasks:

- MAP query: T is the contraction of evidence with the distribution, leaving the query variables open.
- Grafting: T is the contraction of the gradient of the likelihood with the relational encoding of the hypothesis.

Both tasks have been formulated as mode search problems in exponential families.

20.1 Higher-Order Unconstrained Binary Optimization (HUBO)

Here binary refers to the leg dimensions m_k being 2, not to binary coordinates as often refered to in this work.

Definition 77. *The binary optimization of a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^2$ is the problem*

$$\mathrm{argmax}_{x_0,\ldots,x_{d-1}\in[2]} T(x_0,\ldots,x_{d-1}) \tag{P_T}$$

We call Problem P_T a Higher Order Unconstrained Binary Optimization (HUBO) problem of order d and sparsity $\tilde{\ell}(T)$.

The leg dimension needs to be 2, in order to interpret indices as tuple of binaries. Legs with larger dimensions can be splitted into binary legs using a categorical constraint tensor.

The sparsity $\tilde{\ell}(T)$ is the minimal number of monomials, for which a weighted sum is equal to T. Thus we interpret Problem P_T as searching for the maximum in a polynomial consistent of $\tilde{\ell}(T)$ monomial terms.

Remark 34 (Sparsity)**.** *The number $\tilde{\ell}(T)$ of a tensor T defining a HUBO is of central importance to have an effective solution.*

We investigated the Sparsity as the slice sparsity not the vector sparsity in Chapter 19.

20.2 Quadratic Unconstrained Binary Optimization (QUBO)

We refine the monomial decomposition of tensors (see Definition 75) by demanding that monomials consist of at most two variables.

Definition 78. *We call a monomial decomposition $\mathcal{M}$ of a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^2$ a quadratic decomposition, if $|A| \leq 2$ for all $(\lambda, A, x_A) \in \mathcal{M}$. We denote the smallest cardinality $|\mathcal{M}|$ among quadratic decompositions of T by $\mathrm{rank}^{\mathrm{qua}}(T)$.*

If a tensor $T \in \bigotimes_{k\in[d]} \mathbb{R}^2$ has a quadratic decomposition, we call Problem P_T a Quadratic Unconstrained Binary Optimization (QUBO) problem of sparsity $\mathrm{rank}^{\mathrm{qua}}(T)$.

Analogously to monomial decompositions, quadratic decompositions have an equivalence in a CP decomposition of T. Beyond being binary tensors, the leg cores are further restricted that for each slice $l \in [p]$ at most two of them are basis vectors and the rest trivial vectors $\mathbb{I}$.

We notice, that there are tensors, for which no quadratic decomposition exists. This is already obvious from the fact, that the tensors with a quadratic decomposition build a $\binom{d}{2}$ dimensional submanifold in the 2^d dimensional tensor space. This is in contrast with monomial decompositions, where one can always construct a decomposition.

We can transform certain HUBO problems in QUBO problems with the usage of auxiliary variables, as we show in the next lemma.

Lemma 34. *For any $x_0, \dots, x_{d-1} \in [2]$ and $A \subset [d]$ we have*

$$\left(\prod_{k\in A} x_k\right)\left(\prod_{k\notin A}(1-x_k)\right) = \max_{z\in[2]} z\cdot 2\cdot\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right).$$

Proof. Only if $x_k = 1$ for $k \in A$ and $x_k = 0$ else we have

$$\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right) \geq 0\,.$$

In this case the maximum is taken for $z = 1$ and we have

$$\max_{z\in[2]} z\cdot 2\cdot\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right) = 1 = \left(\prod_{k\in A} x_k\right)\left(\prod_{k\notin A}(1-x_k)\right).$$

In all other cases, the maximum is taken for $z = 0$ and thus vanishes, that is

$$\max_{z\in[2]} z\cdot 2\cdot\left(\sum_{k\in A} x_k - |A| - \sum_{k\notin A} x_k + \frac{1}{2}\right) = 0 = \left(\prod_{k\in A} x_k\right)\left(\prod_{k\notin A}(1-x_k)\right).$$

Thus, the claim holds in all cases. □

20.3 Integer Linear Programming

Let us now show how optimization problems can be represented as linear programming problems.

Definition 79. *A Binary Integer Linear Program (ILP) is a problem of the form*

$$\max_{x\in\{0,1\}^n} c^T x \quad \textit{subject to} \quad A^{upper}x \leq b^{upper}, A^{lower}x \geq b^{lower}$$

where $A^{upper} \in \mathbb{R}^{n^{upper}\times n}$, $b^{upper} \in \mathbb{R}^{n^{upper}}$, $A^{lower} \in \mathbb{R}^{n^{lower}\times n}$, $b^{lower} \in \mathbb{R}^{n^{lower}}$.

Theorem 102. *Given a monomial decomposition $\mathcal{M} = \{(\lambda^l, A^l, x^l_{A^l}) : l \in [p]\}$ of a tensor T we define an Binary ILP as the maximation of*

$$\sum_{l\in[p]} \lambda^l z^l$$

with the constraints for any l

- $$z^l \leq X_k \quad \textit{for} \quad k \in A^j, x_k = 1$$
- $$z^l \leq (1 - X_k) \quad \textit{for} \quad k \in A^j, x_k = 0$$
- $$z^l \geq 1 + \sum_{k\in A^l : x_k = 1} (X_k - 1) - \sum_{k\in A^l : x_k = 0} X_k$$

The solution $x^{ILP,\mathcal{M}}$ of this ILP and the solution $x^{HUBO,\mathcal{M}}$ of the HUBO coincide on the variables of hypercore, i.e.

$$x^{ILP,\mathcal{M}}|_{[d]} = x^{HUBO,\mathcal{M}}\,.$$

Proof. We have to show that the constraints are satisfied if and only if $z^l = e_{X^l_{A^l}}[X_{A^l} = x_{A^l}]$. □

21 Reasoning by Tensor Approximation

Often reasoning requires the execution of demanding contractions of tensors networks, or combinatorical search of maximum coordinates. We in this chapter investigate methods, to replace hard to be sampled tensor networks by approximating tensor networks, which then serve as a proxy in inference tasks.

21.1 Approximation of Energy tensors

21.1.1 Direct Approximation

Direct approximation would be

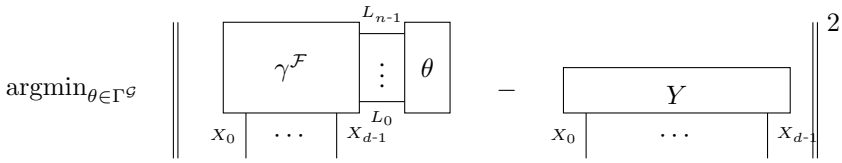
$$\mathrm{argmin}_{\theta \in \Gamma^{\mathcal{G}}} \| E\left[X_{[d]}\right] - \theta\left[X_{[d]}\right] \|^2$$

21.1.2 Approximation involving Selection Architectures

Direct approximation would be

$$\mathrm{argmin}_{\theta \in \Gamma^{\mathcal{G}}} \| E - \left\langle \left\{ \gamma^{\mathcal{H}}, \theta \right\} \right\rangle \left[X_{[d]}\right] \|^2$$

In a tensor network diagram we depict this as



Example 24 (Approximate based on a slice sparsity selecting architecture)**.** *Use a term selecting neural network (conjunction neuron on d unary neurons selecting a variable and* $\mathrm{Id}, \neg, \mathrm{True}$ *as connective selector. Demand the parameter tensor θ to be in a basis CP format, then each slice of the parameter tensor corresponds with the slice of the energy. The use the approximation for MAP search. Same construction possible for probability tensors, but often more involved to instantiate them as tensor network.*

21.2 Transformation of Maximum Search to Risk Minimization

By the squares risk trick, maximum coordinate searches involving contractions with boolean tensors can be turned into squares risk minimization problems. One can use this for MAP inference of MLN and the proposal distribution.

21.2.1 Squares Risk Trick

Lemma 35. *Let T be a Boolean tensor, that is* $\mathrm{im}\,(T) \subset \{0, 1\}$. *Then*

$$T = \mathbb{I} - (T - \mathbb{I})^2$$

where $\mathbb{I}$ is a tensor with same shape as T and all coordinates being 1.

Proof. Since for each $x_{[d]} \in \bigtimes_{k \in [d]} [m_k]$ we have $T[X_{[d]} = x_{[d]}] \in \{0, 1\}$, it holds that

$$T[X_{[d]} = x_{[d]}] = 1 - (T[X_{[d]} = x_{[d]}] - 1)^2$$

and thus

$$T[X_{[d]}] = \mathbb{I}\left[X_{[d]}\right] - (T - \mathbb{I})^2$$

Applying this to all coordinates shows the claim. □

Theorem 103. *Let Θ be a set of binary tensors in $\bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ and $I \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ arbitrary. Then we have*

$$\mathrm{argmax}_{T \in \Theta} \langle I, T \rangle [\varnothing] = \mathrm{argmin}_{T \in \Theta} \left\langle I, (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2 \right\rangle [\varnothing] \tag{81}$$

Proof. Using the Lemma above, T is identical to $\mathbb{I}\left[X_{[d]}\right] - (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2$ and we get

$$\langle I, T \rangle [\varnothing] = \left\langle I, \mathbb{I}\left[X_{[d]}\right] \right\rangle [\varnothing] - \left\langle I, (T\left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right])^2 \right\rangle [\varnothing]$$

Since the first term does not depend on T, it can be dropped in the maximization problem. The (-1) factor then turns the maximization into a minimization problem. □

Corollary **??** 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 Θ to differentiably parametrizable sets, since then common squares risk solvers can be applied. We will call T in the Corollary **??** importance tensor, since it manipulates the relevance of each coordinate in the squares loss.

Example 25 (Proposal distribution maxima)**.** *The Problem* $\mathrm{P}_{\mathcal{F},\mathbb{P}^D,\tilde{\mathbb{P}}}$ *of finding the maximal coordinate can thus be turned into*

$$\mathrm{argmax}_{l_{[n]}} \left\langle (\mathbb{P}^D - \tilde{\mathbb{P}}), \mathcal{H} \right\rangle \left[L_{[n]} = l_{[n]}\right] = \mathrm{argmin}_{l_{[n]}} \left\langle \left\{ (\mathbb{P}^D - \tilde{\mathbb{P}}), \left(\langle \mathcal{H}, e_{l_{[n]}} \left[L_{[n]}\right]\rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right]\right)^2 \right\} \right\rangle [\varnothing] \ .$$

21.3 Parametercores

Parameter cores as tradeoff between expressivity and complexity in sampling. When not hidden variables, their maximum can be found by

Parametercores need also an efficient representation.

In case of skeleton expressions with many placeholders further decomposition for algorithmic efficiency are required.

- Elementary Format (EL-Format):
- CP-Format: Closest to sum of formula tensors (when all vectors are basis, then have a sum).
- TT-Format: Showed better heuristic performance in optimization

For any tensor network decomposition into cores θ^s have the derivative $\frac{\partial}{\partial \theta^s}\theta$ as the tensor network with out the core θ^s.

Remark 35 (Parametercores being basis tensors)**.** *When the parameter core is a basis tensor, the contraction with the parametercore coincides with the respective formula tensor. Thus, we will search for basis tensors optimizing in contractions objectives to specific reasoning tasks, and add them iteratively to the network at hand.*

When the parametercores are a tensor network representing a probability tensor by there contraction, we can interpret them as a probability distribution over formulas.

21.4 Fitting a tensor by a formula tensor

Task: Given a tensor T, find a formula $f \in \mathcal{F}$ such that it coincides with T.

If T is a binary tensor, we understand it as a formula and want to find an f such that its number of worlds is maximal, that is solve the problem

$$\mathrm{argmax}_{f\in\mathcal{F}} \langle \{f \Leftrightarrow T\}\rangle [\varnothing] \ .$$

We can use the squares risk trick and get an equivalent problem

$$\mathrm{argmin}_{f\in\mathcal{F}} \| \ \langle \{f \Leftrightarrow T\}\rangle \left[X_{[d]}\right] - \mathbb{I}\left[X_{[d]}\right] \|^2 \ .$$

21.5 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 9).

Algorithm 9 Alternating Least Squares (ALS)

```
for k ∈ [d] do
    Set U^k to a random element in ℝ^{p_k}
end for
while Stopping criterion is not met do
    for k ∈ [d] do
        Set U^k to a to a solution of argmin_{U^k} ‖I ∘ (ρ(U^1, ..., U^d)|_W − Y)‖^2
    end for
end while
```

22 Uniform Concentration of Random Contractions

We here derive bounds on the uniform concentration of contractions with random tensors.

The width of a vector η is the supremum of contractions with respect to a set Γ is

$$\omega_\Gamma(\eta) = \sup_{\theta \in \Gamma} \langle \{\theta, \eta\} \rangle [\varnothing] \ . \tag{82}$$

We are interested in the random vector

$$\eta^{\mathcal{F},\mathbb{P}^*,D} = \left\langle \left\{ \mathbb{P}^D, \mathcal{F} \right\} \right\rangle [L] - \langle \{\mathbb{P}^*, \mathcal{F}\} \rangle [L]$$

which is the difference between the mean parameters given the empirical distribution and the underlying generating distribution.

We derive bounds for the hypothesis Γ being the set of basis vectors (i.e. for feature search) and being the set of normed vectors (i.e. for feature calibration).

22.1 Naive bounds given binomial coordinates

We here investigate width bounds on random tensors η, which coordinates have marginal distributions by Binomials.

This is the case for $\eta^{\mathcal{H},\mathbb{P}^*,m}$.

Naive means, that we do not exploit the dependencies of the coordinates on each other, as would be the case in more sophisticated chaining approaches.

22.1.1 Basis Vectors

We exploit the sub-Gaussian Norm (Def 2.5.6 in CITE Vershynin Book) to state concentration inequalities.

Definition 80 (Sub-Gaussian Norm)**.** *The sub-Gaussian norm of a random variable X is defined as*

$$\|X\|_{\psi_2} = \inf \left\{ C > \ 0 \ : \mathbb{E} \left[\frac{X^2}{C^2} \right] \leq 2 \right\} \ .$$

Theorem 104. *Any coordinate of η is sub-Gaussian with*

$$\|\eta_i\|_{\psi_2} \leq \frac{C}{\sqrt{\ln 2}} \frac{1}{\sqrt{m}}$$

Proof. Centered Bernoulli is bounded and therefore sub-Gaussian. Binomial is a sum and we apply Proposition 2.6.1 in CITE Vershynin Book. □

Theorem 105. *Let*

$$\Gamma = \{e_i : i \in [p]\}$$

and η be a random vector in $\mathbb{R}^p$, which coordinates have a marginal distribution being centered Binomials with a fixed $m \in \mathbb{N}$. Then

$$\|\omega_\Gamma(\eta)\|_{\psi_2} \leq C \sqrt{\frac{\ln p}{m}} \ .$$

where $C > 0$ is a universal constant.

Proof. Supremum of Sub-Gaussian variables. □

22.1.2 Sphere

We first provide a Chebyshev bound on the width of the sphere.

Theorem 106. *Let η be a random tensor with marginal coordinate distributions by binomials with parameters $(p^{(x)}, m)$.*

For any $\epsilon > 0$, $\tau > 0$ and $m \in \mathbb{N}$ with probability at least $1 - \epsilon$ we have

$$\left\| \frac{\eta - \mathbb{E}[\eta]}{m} \right\|_2 \leq \tau$$

provided that

$$m \geq \frac{\sum_{x \in \times_{k \in [d]}[m_k]} p^{(x)}(1-p^{(x)})}{\tau^2 \epsilon} .$$

Proof. Since the squared norm of the noise is the sum of squared centered and averaged Binomials, we have

$$\mathbb{E}\left[\|\eta - \mathbb{E}[\eta]\|_2^2\right] = \sum_{x \in \times_{k \in [d]}[m_k]} \frac{p^{(x)}(1-p^{(x)})}{m}$$

Here we used that the variance of Binomials with parameters $(p^{(x)}, m)$ is $p^{(x)}(1-p^{(x)})m$.
If follows, that

$$\mathbb{E}\left[\left(\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2\right)^2\right] = \frac{\sum_{x \in \times_{k \in [d]}[m_k]} p^{(x)}(1-p^{(x)})}{m} .$$

Then we apply a Chebyshev Bound to get for any $\tau > 0$

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 > \tau\right] = \mathbb{P}\left[\left(\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2\right)^2 > \tau^2\right] \leq \frac{\sum_{x \in \times_{k \in [d]}[m_k]} p^{(x)}(1-p^{(x)})}{m \cdot \tau^2} \tag{83}$$

For a $\epsilon > 0$ we choose any m with

$$m \geq \frac{\sum_{x \in \times_{k \in [d]}[m_k]} p^{(x)}(1-p^{(x)})}{\tau^2 \epsilon}$$

and get

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 > \tau\right] \leq \epsilon . \tag{84}$$

Thus, we have

$$\mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 \leq \tau\right] = 1 - \mathbb{P}\left[\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 > \tau\right] \geq 1 - \epsilon . \tag{85}$$

□

For $\phi = \delta$ the noise tensor is a rescaled and centered multinomial.

Corollary 13. *Let there be multinomial variable with parameters (p, m) where $p \in \bigotimes_{k \in [d]} \mathbb{R}^{m_k}$ a positive and normed tensor. Let D be a set of independent samples For any $\epsilon > 0$, $\tau > 0$ and $m \in \mathbb{N}$ with probability at least $1 - \epsilon$ we have*

$$\left\|\frac{\eta - \mathbb{E}[\eta]}{m}\right\|_2 \leq \tau$$

provided that

$$m \geq \frac{(1 - \langle\{(p)^2\}\rangle[\varnothing])}{\tau^2 \epsilon} .$$

Proof. Theorem 106 with

$$\sum_{x \in \times_{k \in [d]}[m_k]} p^{(x)}(1-p^{(x)}) = \sum_{x \in \times_{k \in [d]}[m_k]} p^{(x)} - \sum_{x \in \times_{k \in [d]}[m_k]} (p^{(x)})^2 = 1 - \langle\{(p)^2\}\rangle[\varnothing] .$$

□

22.1.3 Bounds based on the sub-gaussian norm

Unclear whether this is needed.

A faster tail decay can be achieved, when bounding sub-gaussian norms.

Theorem 107. *Let η be a random vector in $\mathbb{R}^p$, which coordinates have a marginal distribution being centered Binomials with a fixed $m \in \mathbb{N}$. Then*

$$\|\omega_{\mathcal{S}}(\eta)\|_{\psi_2} \leq C\sqrt{\frac{p}{m}}\,.$$

where $C > 0$ is a universal constant.

Proof. Using that each coordinate has Sub-gaussian norm of at most 1. Asymptotically, the binomial tends to a gaussian, which has a smaller sg norm. But the binomial has a sub-exponential regime preventing tighter sg bounds.

Norm of a Sub-gaussian vector, another application of Proposition 2.6.1 in CITE Vershynin Book. □

22.2 Chaining bounds given binomial coordinates

To proceed with the uniform concentration investigation, we need a concentration bound on Binomials.

Theorem 108. *For any $p \in [0,1]$ and $m \in \mathbb{N}$ any $X \sim B(p,m)$ satisfies for any $t > 0$*

$$\mathbb{P}\left[X - \mathbb{E}\left[X\right] > t\right] \leq \exp\left[-\frac{t^2}{2mp + \frac{2t}{3}}\right]$$

and

$$\mathbb{P}\left[\mathbb{E}\left[X\right] - \ X > t\right] \leq \exp\left[-\frac{t^2}{2mp}\right]$$

Thus

$$\mathbb{P}\left[|\mathbb{E}\left[X\right] - \ X| > t\right] \leq 2\exp\left[-\frac{t^2}{2mp + \frac{2t}{3}}\right]$$

Proof. See e.g. https://mathweb.ucsd.edu/ fan/wp/concen.pdf □

The binomial thus has a sub-gaussian and a sub-exponential regime.

Theorem 109. *For any $p \in [0,1]$ and $m \in \mathbb{N}$ any $X \sim B(p,m)$ satisfies for any $t > 0$*

$$\mathbb{P}\left[|\mathbb{E}\left[X\right] - \ X| > \sqrt{4mpt} + 2t\right] \leq 2\exp\left[-t\right]$$

Proof. For any s>0 we choose t>0 such that

$$s = -\frac{t^2}{2mp + \frac{t}{3}}$$

and observe

$$\min\left(\frac{t^2}{4mp}, \frac{t^2}{2t}\right)$$

and

$$t \leq \max(\sqrt{4mps}, 2s) \leq \sqrt{4mps} + 2s\,.$$

With the above bound it holds, that

$$\mathbb{P}\left[|\mathbb{E}\left[X\right] - \ X| > \sqrt{4mps} + 2s\right] \leq \mathbb{P}\left[|\mathbb{E}\left[X\right] - \ X| > t\right] \leq 2\exp\left[-\frac{t^2}{2mp + \frac{t}{3}}\right] \leq 2\exp\left[-s\right]\,.$$

□

We apply this on the variable $\langle\theta, \eta\rangle$.

Theorem 110. *Let $\theta = \sum_{f\in\mathcal{F}} \theta_f f$, then*

$$\mathbb{P}\left[m|\langle\theta,\eta\rangle| \geq \sqrt{4m}\cdot\left(\sum_{f\in\mathcal{F}}|\theta_f|\sqrt{p^{(f)}}\right)\sqrt{t} + 2\cdot\left(\sum_{f\in\mathcal{F}}|\theta_f|\right)t\right] \leq 2|\mathcal{F}|\cdot\exp[-t]$$

Proof. We can not assume independence of the $\langle f,\eta\rangle$ (in that case we could use a Bernstein inequality) and instead take the naive bound over all formulas in $\mathcal{F}$

$$\begin{aligned}
&\mathbb{P}\left[|\langle\theta,\eta\rangle| \geq \sqrt{4m}\cdot\sum_{f\in\mathcal{F}}|\theta_f|\sqrt{p^{(f)}}\sqrt{t} + 2\cdot\sum_{f\in\mathcal{F}}|\theta_f|t\right] && (86)\\
&\leq \mathbb{P}\left[\exists f\in\mathcal{F} : |\langle f,\eta\rangle| \geq \sqrt{4m}|\theta_f|\sqrt{p^{(f)}}\sqrt{t} + 2|\theta_f|t\right] && (87)\\
&\leq \sum_{f\in\mathcal{F}}\mathbb{P}\left[|\langle f,\eta\rangle| \geq \sqrt{4m}|\theta_f|\sqrt{p^{(f)}}\sqrt{t} + 2|\theta_f|t\right] && (88)\\
&\leq 2|\mathcal{F}|\cdot\exp[-t]\ . && (89)
\end{aligned}$$

□

We can thus proof uniform concentration bounds given covering bounds of a hypothesis set in ℓ_1 norm (and the reweighted one).

Remark 36 (Small Formula Probabilities)**.** *Directions with small $p^{(f)}$ will require larger covering sets and thus have large contributions to the bounds. Intuitively, they correspond with exceptional special cases, which need many samples to be observed. Strange: Should also $1-p^{(f)}$ small intuitely be an issue?*

22.2.1 Generic Width Bounds

We define the maps

$$\nu_2^p(\theta) = \inf_{(\mathcal{F},\theta):\sum_{f\in\mathcal{F}}\theta_f f=\theta}\left(\sum_{f\in\mathcal{F}}|\theta_f|\sqrt{p^{(f)}}\right)$$

and

$$\nu_1(\theta) = \inf_{(\mathcal{F},\theta):\sum_{f\in\mathcal{F}}\theta_f f=\theta}\left(\sum_{f\in\mathcal{F}}|\theta_f|\sqrt{p^{(f)}}\right)$$

corresponding to the sub-gaussian and sub-exponential regimes.

Theorem 111. *Let $\mathcal{F}$ be a set of formulas and $W\subset\mathbb{R}^{|\mathcal{F}|}$ a set of weight vectors. Then with probability at least $1-|\mathcal{F}|C_1\exp\left[-\frac{u^2}{2}\right]$ we have for the set*

$$\Gamma = \left\{\sum_{f\in\mathcal{F}}\theta_f f : (\theta_f)_{f\in\mathcal{F}}\in W\right\}$$

the bound

$$\omega_\Gamma(\eta) \leq \frac{C_2\gamma_2(\Gamma,\nu_2^p)}{\sqrt{m}} + \frac{C_3\gamma_1(\Gamma,\nu_1)}{m}\ .$$

A Implementation in the tnreason package

We here document the implementation of the discussed concepts in the python package tnreason .

tnreason is an abbreviation of **t**ensor **n**etwork **reason**ing, by which we emphasize the capabilities of this package to represent and answer reasoning tasks by tensor network contractions.

The package can be installed either by cloning https://github.com/EnexaProject/enexa-tensor-reasoning or by

```
!pip install tnreason
```

A.1 Script Language

To specify propositional sentences, neuro-symbolic architectures and Markov Logic Networks, we developed a script language.

A.1.1 Propositional Sentences by Nested Lists

Are those of Propositional Logics, but instead of brackets we nest the symbols into lists.

Connectives are represented by strings, where the following are supported (see Definition **??**):

Unary connective $\circ$	$S(\circ)$
$\neg$	"not"
$()$	"id"

Binary connective $\circ$	$S(\circ)$
$\wedge$	"and"
$\vee$	"or"
$\Rightarrow$	"imp"
$\oplus$	"xor"
$\Leftrightarrow$	"eq"

Atomic Formulas are represented by arbitrary strings, which are not used for the representation of connectives. We further avoid the symbols {"(", ")", "_"} in the names of atoms, to not confuse them with colors of categorical variables.

Composed Formulas $f_1 \circ, f_2$ are represented by

$$[S(\circ), S(f_1), S(f_2)]$$

where we apply the conventions

- Connectives are at the 0th position in each list
- Further entries are either atoms as strings or encoded formulas itself

The applied grammar in Backus-Naur form is

Unary Connective	"not" | "id"
Binary Connective	"and" | "or" | "imp" | "xor" | "eq"
Atomic Formula	Set of strings not in Connectives
Complex Formula	Atomic Formula | [Unary Connective, Complex Formula] | [Binary Connective, Complex Formula, Complex Formula]

Example 26 (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"]]*

A.1.2 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

$$\{\text{key}(f) : S(f) \text{ for } f \in \mathcal{KB}\}$$

A.1.3 Markov Logic Networks

The formulas with soft interpretation are called weighted formulas and encoded by $\exp[\theta_f \cdot f]$. We thus require a specification of the weights, which we do by adding θ_f as a float or an int to the list $S(f)$. We then store Markov Logic Networks by dictionaries

$$\{\text{key}(f) : S(f) + [\theta_f] \text{ for } f \in \mathcal{F}\}$$

A.1.4 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 Definition 43 and **??**).

A connective selector (see Definition 43) is encoded by the list

$$S(\circ) = [S(\circ_0), \ldots, S(\circ_{p\text{-}1})]$$

and a formula selector (see Definition **??**) by

$$S(\mathcal{H}) = [S(\circ_0), \ldots, S(\circ_{p\text{-}1})]$$

A logical neuron of order n (see Definition 47), defined by a connective selector $\circ$, and a formula selector $\mathcal{H}_k$ on each argument $k \in [n]$, is encoded by

$$S(\sigma) = [S(\circ), S(\mathcal{H}_0), \ldots, S(\mathcal{H}_{n-1})]$$

Only the unary $n = 1$ and the $n = 2$ cases are supported.

The resulting nested lists indices have an alternating interpretation at each level compared with the elements of each list. That is, when $S(\sigma)$ is the encoding of a neuron, then any element $x \in S(\sigma)$ represents a list of choices. When x is not the first element, then each choice is either the encoding $S(X)$ of an atomic formula, or another neuron.

A neural architecture $\mathcal{A}$ is then represented in the dictionary

$$S(\mathcal{A}) = \{\text{key}(\sigma) : S(\sigma) \text{ for } f \in \mathcal{A}\}$$

where key(σ) is a string, which can be used in the formula selections of other neurons.

It is important that the directed graph of neurons induced by the choice possibilities is acyclic, to ensure well-definedness of the architecture.

In order to represent neuro-symbolic architectures, the grammar of $S(\cdot)$ in Backus-Naur Form is extended by the production rules

Unary Connectives	[Unary Connective] | [Unary Connective] + Unary Connectives
Binary Connectives	[Unary Connective] | [Binary Connective] + Binary Connectives
Dependency Choice	Atomic Formula | Neuron
Dependency Choices	[Dependency Choice] | [Dependency Choice] + Dependency Choices
Neuron	[Unary Connectives, Dependency Choices] | [Binary Connectives, Dependency Choices, Dependency Choices]

Example 27 (Neuro-Symbolic Architecture for the Wet Street)**.** *Following the wet street example, we can define a neuron by*

$$S(\sigma) = \textit{[["imp","eq"],["Wet","Sprinkler"],["Street"]]}$$

from which the formulas

["imp", "Wet", "Street"]
["eq", "Wet", "Street"]
["imp", "Sprinkler", "Street"]
["eq", "Sprinkler", "Street"]

can be chosen. Combining this neuron with further neurons, e.g. by the architecture

$$S(\mathcal{A}) = \textit{\{ "neur1": [["imp","eq"],["neur2"],["Street"]] ,}$$
$$\textit{"neur2": [["lnot","id"],["Wet","Sprinkler"],["Street"]] \}}$$

the expressitivity increases. In this case, the further neuron provides the flexibility of the first atoms to be replaced by its negation.

A.2 Tensor Networks

A.2.1 Core Nomenclature

Appearing cores:

- "_conCore": logical connectives (relational encoding of the connective map)
- "_headCore": Binary vectors representing of the exponential (directly a tensor)
- "_dataCore": Representing (relational encoding of the data map)

A.2.2 Color Nomenclature

Represent the three appearing types of variables:

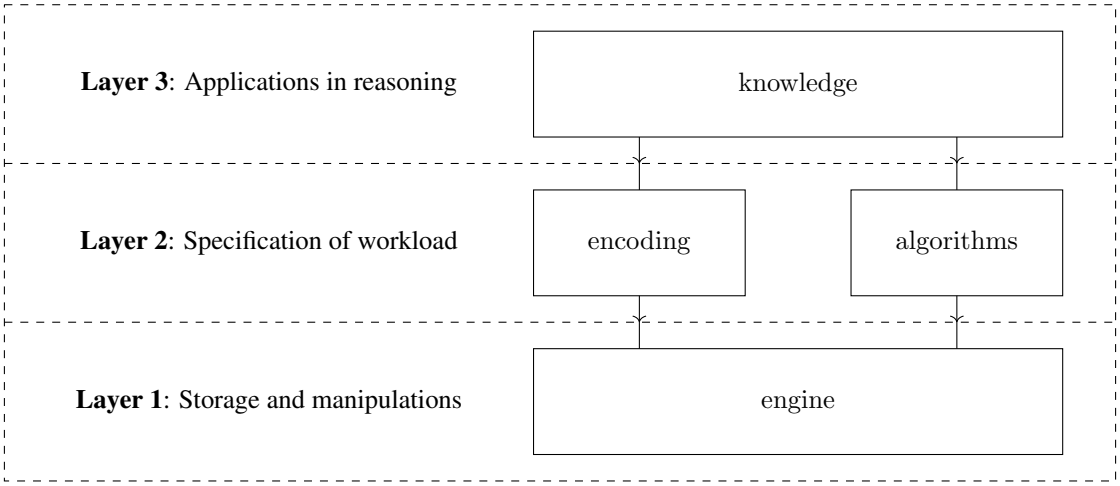
- Categorical variables $X_\cdot$: cVar / aVar
- Selection variables $L_\cdot$: sVar
- Term variables $O_\cdot$: tVar

A.3 Architecture

tnreason is structured in four subpackages and three layers

- Layer 1: Storage and numerical manipulations, by subpackage engine , "Tensor Networks" -> building "tn" of tnreason
- Layer 2: Specification of workload, subpackage encoding specific for storage, subpackage algorithms specific for manipulations
- Layer 3: Applications in reasoning, by subpackage knowledge , "Reasoning" -> building "reason" of tnreason

We sketch this structure by



A.4 Subpackage engine

The engine subpackage is for the storage and numerical manipulation of tensors and tensor networks.

We think of it as the lowest layer, specializing in storage of Tensor Networks and performing the contractions.

Cores

Each Tensor core has attributes

- values (array-like): storing the value of the coordinates
- colors (list of str): specifying the name of the variables represented by its axes
- name (str): to distinguish from other cores

The implemented core types differ in the values argument. Cores are instantiated by

```
engine.getCore(coreType)(coreValues, coreColors, coreName)
```

Polynomial Cores Polynomial Cores are implementations of the monomial decomposition or basis+ (see Definition 75). Here the each tuple (λ, A, X_A) is stored as a tuple of the scalar λ and a dictionary with A as keys and X_A as values.

The supported cores are

coreType	**Package**	Explanation
"NumpyTensorCore"	numpy	Numpy array storing the values
"PolynomialCore"	numpy	Storeing the values in a binary CP Decomposition

Binary CP Decomposition

Based on the monomial decomposition $\tilde{\ell}(\cdot)$ as specified in Definition 75. To store the values of a tensor we store the slices of tensors by the indices x_A.

Contractions can be performed by partially contracting the cores of the decomposition. In this way, one can avoids coordinatewise storages of high-order tensors, which can be intractable.

Tensor Networks

Tensor networks $\mathcal{T}^{\mathcal{G}}$ are defined by hypergraphs with hyperedges decorated by tensor cores. We store them by dictionaries with values being tensor cores and keys coinciding with the name of each tensor core.

Contractions

Reflected in the notation

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

a contraction is defined by

- Tensor Network $\mathcal{T}^{\mathcal{G}}$, i.e. a dictionary of tensor cores
- Open Variables $\mathcal{V}$

Contraction calls are done by

engine.contract(contractionMethod, coreDict, openColors, evidenceColorDict)

Where

- contractionMethod: str, chooses one of the contractionproviders
- coreDict: Dictionary of TensorCores (of the above formats), representing the Tensor Network $\mathcal{T}^{\mathcal{G}}$
- openColors: List of str, each str identifying a color, that is a variable to be left open in the contraction
- evidenceColorDict: Dict valued by int and keys by str, indicating sliced variables

The supported contraction methods are

contractionMethod (str)	**Package**	Explanation
"NumpyEinsum"	numpy	Einstein summation of numpy arrays
"TensorFlowEinsum"	tensorflow	Einstein summation of tensorflow tensors
"TorchEinsum"	torch	Einstein summation of torch tensors
"TentrisEinsum"	tentris	Einstein summation of tentris hypertries
"PgmpyVariableEliminator"	pgmpy	Variable Elimination of DiscreteFactors in pgmpy
"PolynomialContractor"	numpy	Contraction of CP Decompositions stored in numpy arrays

Contractions represented as Einstein summation, as implemented in:

- numpy
- tensorflow
- pytorch
- tentris

Contractions can be executed by variable elimination as implemented in:

- pgmpy

Manipulation of Binary CP Decomposition Contraction of tensors in Binary CP Decomposition as in Section **??**.

A.5 Subpackage encoding

In the encoding subpackage we encode maps Here the relational encodings ρ^f of various maps f are created. The maps are either specified by the script language (logical formulas or neuro-symbolic architectures), categorical constraints or data. Given a specification of a formula f in script language $S(\cdot)$, the task amounts to building a semantic representation based on the syntactic specification.

We arrange the encoding subpackage into the second layer of the tnreason architecture, since it specifies tensor cores which formats are specified in engine .

A.5.1 Relational encoding of formulas

Core names are

- Connective cores with suffix "_conCore" in name
- Head cores with suffix "_headCore" in name
- Data cores with suffix "_dataCore" in name
- Categorical constraint cores with suffix "_catCore" in name

Propositional formulas f are represented in three schemes:

- Script language $S(f)$ by nested lists (see Section A.1). Most practical to choose a formula from a neuro-symbolic architecture.
- Strings specifying the categorical variables X_f.
- Representation of formulas by tensor networks being contracted to ρ^f

Conversions of the formats:

- $S(f)$ to color by

  encoding.get_formula_color($S(f)$)

  Here the nested lists are turned in a string by concatenating all elements of a list with "_" and adding "[" and "]" at the beginning and end of each list.
- $S(f)$ to tensor network

  encoding.create_raw_cores($S(f)$)

  This creates the connective cores for the semantic representation of ρ^f. We encode them by

When encoding formulas with hard interpretation, we furthermore add a head core of type "truthEvaluation" since we have

$$f = \langle \{\rho^f, e_1\} \rangle [X_f] \ .$$

A.5.2 Representation of MLNs

Structure Cores are binary cores relating the variables in a predefined way, which is not changing during reasoning.

- Logical interpretation: Cores ρ° Structure Cores are those of the Bayesian Propositional Network
- Categorical constraints: Cores ρ^Z

Activation Cores encode the weights of the formulas in a Markov Logic Network. For proper MLN only have unary cores, which we call headCores. Head cores with suffix "headCore" in name.

They are modified during reasoning: Selection of activation cores in structure learning, assigning a weight in parameter estimation.

A.5.3 Formula Selecting Maps

Encoding of Neurons according to Definition 47:

- Activation selection core with suffix "actCore" in name. Selection by variable with suffix "actVar"

- Selection of neurons as arguments with suffix "selCore" in name. Each argument of each neuron comes with a control variable with suffix "selVar".

Encoding of Formula Selecting Neural Networks (Definition 47) by creating all formula selecting neurons.

Skeleton expression (Definition **??**) are stored with placeholderkeys and the candidatelists by dictionaries with the placeholderkeys and values being the possible symbols.

A.6 Subpackage algorithms

The algorithms subpackage implements basic tensor network algorithms with calls of specific execution in engine . As the encoding subpackage it is arranged in the second layer of the tnreason architecture, since it specifies the manipulation of tensor networks in the engine subpackage.

A.6.1 Alternating Least Squares

- Tensor Network of Structure Cores
- Tensor Network of Parameter Cores
- List of importance cores

  algorithms.ALS

A.6.2 Gibbs Sampling

- Tensor Network of Structure Cores
- Parameter cores: Variable tensor network cores representing basis vectors.
- List of importance cores

  algorithms.Gibbs

A.6.3 Knowledge Propagation

algorithms.ConstraintPropagator

A.7 Subpackage knowledge

With the knowledge subpackage we provide an interface for reasoning workload. It builds a third layer, since it used encoding to represent knowledge by tensor networks and algorithms in the execution of reasoning tasks.

A.7.1 Distributions

We encode Markov Networks by specifying a set of tensor cores. Each distribution needs to have a routine

.create_cores()

creating the factor cores and

.get_partition_function()

calculating the partition function. Although the partition function can be calculated by the contraction of all cores, we separate the method since there are situations where a faster calculation can be performed.

Empirical Distributions are distributions of sample data. We represent the values as a CP Format of data cores as specified in Section 3.8

knowledge.EmpiricalDistribution

Here the partition function is the number of samples used in the creation of the empirical distribution.

HybridKnowledgeBases are probability distributions, which are specified by propositional formulas in the script language.

knowledge.HybridKnowledgeBase

They are initialized with arguments

- facts: Dictionary of propositional formulas stored as $S(f)$ representing hard logical constraints
- weightedFormulas: Dictionary of propositional formulas stored as $S(f)$+$[\theta_f]$ representing soft logical constraints
- evidence: Dictionary of atomic formulas, where key are the formulas in string representation and values the certainty in $[0, 1]$ (float or int) of the atom being true
- categoricalConstraints: Dictionary of categorical constrained, which values are lists of atomic formulas stored as strings $S(X)$

A.7.2 Inference

By

knowledge.InferenceProvider

taking a distribution from the above as argument.

Probabilistic queries as specified Definition 29) by

.query(variableList, evidenceDict)

MAP queries by

.exact_map_query()

or by

.annealed_sample()

using Simulated Annealing (see Remark **??**) to find an approximate maximum. The second method circumvents the creation of the coordinatewise representation of the distribution and circumvents therefore, at the expense of potentially approximative solutions, a bottleneck in case of many query variables.

Entailment from the distribution (Definition **??**) 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.7.3 Parameter Estimation

EntropyMaximizer implements Algorithm 7, which is motivated by the maximum entropy principle (see Section 4.5.3) to optimize Markov Logic Networks. The class

knowledge.EntropyMaximizer

is initialized with the arguments

- expressionsDict: Dictionary of formulas in the format $S(f)$
- satisfactionDict: Dictionary of the satisfaction rates (mean parameters) to be matched by the optimal distribution

The optimization is then performed by

.alternating_optimization(sweepNum, updateKeys)

method, where the iteration in Algorithm 7 through the updateKeys is performed sweepNum times.

A.7.4 Structure Learning

Formula Booster chooses a formula given a formula selecting map.

knowledge.FormulaBooster

is initialized with the arguments

- knowledgeBase: Distribution representing a current model to be improved
- specDict: A neuro-symbolic architecture encoded in a dictionary of neurons

References

Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic Tensor Networks. *Artificial Intelligence*, 303:103649, February 2022. ISSN 0004-3702. doi: 10.1016/j.artint.2021.103649. URL https://www.sciencedirect.com/science/article/pii/S0004370221002009.

Alexander Bigerl, Felix Conrads, Charlotte Behning, Mohamed Sherif, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Tentris -A Tensor-Based Triple Store. September 2020.

William Cohen, Fan Yang, and Kathryn Rivard Mazaitis. TensorLog: A Probabilistic Database Implemented Using Deep-Learning Infrastructure. *Journal of Artificial Intelligence Research*, 67:285–325, February 2020. ISSN 1076-9757. doi: 10.1613/jair.1.11944. URL https://jair.org/index.php/jair/article/view/11944.

Caglar Demir and Axel-Cyrille Ngonga Ngomo. DRILL- Deep Reinforcement Learning for Refinement Operators in ALC. *CoRR*, abs/2106.15373, 2021. URL https://ris.uni-paderborn.de/record/25217.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422, Rio de Janeiro Brazil, May 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488425. URL https://dl.acm.org/doi/10.1145/2488388.2488425.

Patrick Gelß, Stefan Klus, Jens Eisert, and Christof Schütte. Multidimensional Approximation of Nonlinear Dynamical Systems. *Journal of Computational and Nonlinear Dynamics*, 14(6):061006–061006–12, April 2019. ISSN 1555-1415. doi: 10.1115/1.4043148.

Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press, September 2019. ISBN 978-0-262-53868-8.

Ivan Glasser, Ryan Sweke, Nicola Pancotti, Jens Eisert, and Ignacio Cirac. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

Alex Goeßmann, Ingo Roth, Gitta Kutyniok, Michael Götte, Ryan Sweke, and Jens Eisert. Tensor network approaches for data-driven identification of non-linear dynamical laws. In *Advances in Neural Information Processing Systems - First Workshop on Quantum Tensor Networks in Machine Learning*, page 21, 2020.

Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer Series in Computational Mathematics. Springer-Verlag, Berlin Heidelberg, 2012. ISBN 978-3-642-28026-9. doi: 10.1007/978-3-642-28027-6.

Sepp Hochreiter. Toward a broad AI. *Communications of the ACM*, 65(4):56–57, April 2022. ISSN 0001-0782, 1557-7317. doi: 10.1145/3512715. URL https://dl.acm.org/doi/10.1145/3512715.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, Jose Emilio Labra Gayo, Roberto Navigli, and Sebastian Neumaier. *Knowledge Graphs*. Springer, Cham, 1st edition edition, November 2021. ISBN 978-3-031-00790-3.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, Mass., 1. edition edition, July 2009. ISBN 978-0-262-01319-2.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Neural Class Expression Synthesis, December 2022. URL http://arxiv.org/abs/2111.08486. arXiv:2111.08486 [cs].

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Neural Class Expression Synthesis. In Catia Pesquita, Ernesto Jimenez-Ruiz, Jamie McCusker, Daniel Faria, Mauro Dragoni, Anastasia Dimou, Raphael Troncy, and Sven Hertling, editors, *The Semantic Web*, volume 13870, pages 209–226. Springer Nature Switzerland, Cham, 2023. ISBN 978-3-031-33454-2 978-3-031-33455-9. doi: 10.1007/978-3-031-33455-9_13. URL https://link.springer.com/10.1007/978-3-031-33455-9_13. Series Title: Lecture Notes in Computer Science.

Jens Lehmann, Sören Auer, Lorenz Bühmann, and Sebastian Tramp. Class expression learning for ontology engineering. *Journal of Web Semantics*, 9(1):71–81, March 2011. ISSN 1570-8268. doi: 10.1016/j.websem.2011.01.001. URL https://www.sciencedirect.com/science/article/pii/S1570826811000023.

Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, 328:104062, March 2024. ISSN 0004-3702. doi: 10.1016/j.artint.2023.104062. URL https://www.sciencedirect.com/science/article/pii/S0004370223002084.

Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994. URL https://www.sciencedirect.com/science/article/pii/0743106694900353. Publisher: Elsevier.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1):11–33, January 2016. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2015.2483592. URL https://ieeexplore.ieee.org/document/7358050/.

Elina Robeva and Anna Seigal. Duality of graphical models and tensor networks. *Information and Inference: A Journal of the IMA*, 8(2):273–288, June 2019. ISSN 2049-8772. doi: 10.1093/imaiai/iay009.

Chiaki Sakama, Katsumi Inoue, and Taisuke Sato. Linear Algebraic Characterization of Logic Programs. In Gang Li, Yong Ge, Zili Zhang, Zhi Jin, and Michael Blumenstein, editors, *Knowledge Science, Engineering and Management*, volume 10412, pages 520–533. Springer International Publishing, Cham, 2017. ISBN 978-3-319-63557-6 978-3-319-63558-3. doi: 10.1007/978-3-319-63558-3_44. URL http://link.springer.com/10.1007/978-3-319-63558-3_44. Series Title: Lecture Notes in Computer Science.

Taisuke Sato. A linear algebraic approach to datalog evaluation. *Theory and Practice of Logic Programming*, 17(3):244–265, May 2017. ISSN 1471-0684, 1475-3081. doi: 10.1017/S1471068417000023. URL https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/linear-algebraic-approach-to-datalog-evaluation/CED3EEB903D9D8A16843CFCA5AC4D577. Publisher: Cambridge University Press.

Theo Trouillon. Knowledge Graph Completion via Complex Tensor Factorization. 2017.

Théo Trouillon and Maximilian Nickel. Complex and Holographic Embeddings of Knowledge Graphs: A Comparison, July 2017. URL http://arxiv.org/abs/1707.01475. arXiv:1707.01475 [cs, stat].

Martin J. Wainwright and Michael Irwin Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc, 2008. ISBN 978-1-60198-184-4.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. arXiv, August 2015. URL http://arxiv.org/abs/1412.6575. arXiv:1412.6575 [cs].