MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kozák Jan** Personal ID number: **420381**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Branch of study: **Data Science**

II. Master's thesis details

Master's thesis title in English:

Efficient Algorithms for Relational Marginal Polytope Construction

Master's thesis title in Czech:

Efektivní algoritmy pro konstrukci relačních marginálních polytopů

Guidelines:

1. Get familiar with the framework of Markov logic networks and the probabilistic inference problems tackled within it.
2. Get familiar with the primal formulation of Markov logic network learning (so-called "relational marginal problems") and with the notion of "relational marginal polytope".
3. Design efficient heuristic algorithms for construction of relational marginal polytopes. Consider also approximation algorithms with rigorous guarantees.
4. Implement the designed algorithms and use them either inside the recent method for weight learning of Markov logic networks (using the algorithm described in Kuželka, O. and Kungurtsev, V., AISTATS 2019) or for removing redundant first-order logic formulas from Markov logic networks.
5. Compare your algorithms to the naive domain-lifted algorithms based on reductions from weighted-first-order model counting from (Kuzelka, O., and Wang, Y., AISTATS 2020)
6. Optional: Can you design a faster algorithm for detecting when the relational marginal polytope lives in a lower dimensional affine subspace (i.e. when some of the formulas are redundant) instead of constructing the polytope?

Bibliography / sources:

[1] Koller, D., Friedman, N., Džeroski, S., Sutton, C., McCallum, A., Pfeffer, A., ... & Neville, J. (2007). Introduction to statistical relational learning. MIT press.

[2] Richardson, Matthew, and Pedro Domingos. 'Markov logic networks.' Machine learning 62.1-2 (2006): 107-136.

[3] Kuželka, O., Wang, Y., Davis, J., & Schockaert, S. Relational marginal problems: Theory and estimation. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[4] Kuželka, O. and Kungurtsev, V., Lifted Weight Learning of Markov Logic Networks Revisited. AISTATS 2019: 22nd International Conference on Artificial Intelligence and Statistics, 2019.
[5] Kuželka, O., Wang Y., Domain-Liftability of Relational Marginal Polytopes, AISTATS 2019: 23rd International Conference on Artificial Intelligence and Statistics, 2020.

Name and workplace of master's thesis supervisor:

Ing. Ondřej Kuželka, Ph.D., Intelligent Data Analysis, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **11.02.2020** Deadline for master's thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

Ing. Ondřej Kuželka, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Computer Science

Master's Thesis

Efficient Algorithms for Relational Marginal Polytope Construction

Jan Kozák

August 2020
Supervisor: Ing. Ondřej Kuželka, Ph.D.

/ Declaration

I declare that this thesis has been composed solely by myself and except wherestates otherwise by reference or acknowledgment, the work presented is entirely my own.

In Prague, 14 August 2020

Abstrakt / Abstract

Cílem práce je navržení efektivních heuristik a/nebo aproximačních algoritmů pro konstrukci *relačních marginálních polytopů*. Ty jsou geometrickou reprezentací množiny přípustných řešení tzv. relačního marginálního problémů, což je konvexní optimalizační úloha hledající pravděpodobnostní rozdělení nad možnými světy v Markovských logických sítích mající maximální entropii. Heuristický algoritmus je porovnán s naivním exaktním doménově liftovatelným algoritmem popsaným Kuželkou a Yangem v jejich článku *Domain-Liftability of Relational Marginal Polytopes*, 2020 [1].

Klíčová slova: Markovské logické sítě, relační marginální polytopy

Překlad titulu: Efektivní algoritmy pro konstrukci relačních marginálních polytopů

The goal of the thesis is to design an efficient heuristic and/or approximation algorithms for construction of *relational marginal polytopes*, a geometrical representation of the set of feasible solutions of the relational marginal problem, which is a convex optimization task of finding the max-entropy distributions over possible worlds in Markov logic networks (MLN). The heuristic is compared to naive exact domain-liftable algorithm described by Kuželka and Yang in their paper *Domain-Liftability of Relational Marginal Polytopes*, 2020 [1].

Keywords: Markov Logic Networks, Relational Marginal Polytopes

/ Contents

Tables / Figures

Chapter 1
Introduction

The goal of the thesis is to design an efficient heuristic and approximation algorithms for calculation of *relational marginal polytopes*, a geometrical representation of the set of feasible solutions of the relational marginal problems which is a convex optimization task of finding the max-entropy distributions over possible worlds in Markov logic networks (MLN). MLNs are systems used in the statistical relational learning, a subfield of machine learning that is concerned with learning from relational data. MLNs are a generalization of the first-order probabilistic logic where each predicate is associated with a weight. The weight of the formula roughly specifies the level of our belief in it and importance of the formula — the higher the weight, the less probable is the possible world which violates it. The MLN may be also considered a template for creation of Markov random fields (or Markov nets), which are — together with Bayesian networks — one of the most commonly used probabilistic graphical models, which capture dependencies among random variables into a graph, allowing for more efficient evaluation of inference queries over (possibly) large field of random variables.

The thesis is structured into following chapters:

- Preliminaries — the chapter summarizes important basic concepts related to Markov logic networks. First the general approaches for handling uncertainty in logic are described followed by overview of probabilistic graphical models.
- Markov logic networks — the chapter describes properties and definitions related to Markov logic networks.
- Implementation — the chapter describes implementation of algorithms.

Chapter 2
Preliminaries

This chapter provides a basic background about mathemathical, logical and machine learning concepts that are related to the topic of the thesis. First the first-order logic (FOL) considered in the thesis is described, followed by description of probabilistic logics which incorporate uncertainty into the standard first-order or propositonal logics. Finally a notion of probabilistic graphical models is debated, focused on Bayesian networks and Markov random fields. The former are integral part of Markov logic networks, the key topic of the thesis.

2.1 First-Order Logic

The thesis considers a function-free first-order logic language $\mathcal{L}$ built from sets $Const$ (constants), $Vars$ (variables) and Rel (predicates). The set of predicates Rel is partitioned into subsets Rel_i each containing predicates of arity i, so $Rel = \bigcup_i Rel_i$. The constants represent the domain objects (e.g. `Alice`, `Bob`, `Prague`) and the variable symbols range over them. The predicates represent relations among objects (e.g. `Friends`) or their attributes (e.g. `Capital`). These three sets together constitute *non-logical symbols* and their actual meaning is specified by an *interpretation*. In addition to them the language $\mathcal{L}$ is also built from a standard set of *logical symbols*:

- *universal* ($\forall$) and *existential* ($\exists$) quantifiers,
- unary logical connective – *negation* ($\neg$),
- binary logical connectives – *and* ($\wedge$), *or* ($\vee$), *implication* ($\Rightarrow$) and *equivalence* ($\Leftrightarrow$).

First-order logic theories about domains being modelled are formulated by means of *formulas*. Following list summarizes terminology related to their creation.

- **Term** is a constant or a variable.
- **Atom** or **atomic formula** is a k-ary predicate $R(a_1, a_2, ..., a_k)$ with arguments $a_1, a_2, ...a_k \in Const \cup Vars$ (i.e. terms).
- **Literal** is an atom or its negation.
- **Formula** is a literal or a logical connection of two formulas (may be also applied recursively),
  - set of variables appearing in formula α is denoted as $Vars(\alpha)$,
  - formula α is called *ground formula* if its arguments are constants,
  - formula α_0 is called *grounding of formula* α if it can be obtained by substituting all variables in $Vars(\alpha)$ with constants from $Const$,
  - a variable in a formula is called *free* if it is not bound by any quantifier.
- **Sentence** is a formula with no free variables.

A special type of formula is a *clause* which is a disjunction of literals. Every formula in FOL can be mechanically transformed to conjunction of clauses, so called *clausal form* or *conjuctive normal form* (CNF). This form is convenient for automated processing

and due to beforementioned transformation we can consider all formulas to be in CNF without loss of generality.

A possible world ω is an assignment of truth values to every possible ground atom. A formula is *satisfiable* if there exists at least one possible world in which it holds true. All formulas together form a *knowledge base* (KB). The knowledge base might be considered a one big conjunction of all its formulas, as in basic setting it is expected that all formulas in the KB are simultaneously true. A typical inference problem involving usage of a knowledge base is to decide if the KB *entails* formula F (denoted as $KB \models F$), that is if F is true whenever KB holds. This is usually checked by *refutation* – $KB \models F$ holds iff $KB \cup \neg F$ is not satisfiable. Note however that this yields a positive answer also in cases when KB contains a contradiction.

First-order logic used in the thesis is further restricted by following assumptions:

- unique names assumption – different constants refer to different objects,
- injective substitution – different variables in a formula must be mapped to different terms,
- only domains of finite size are considered.

2.1.1 Probabilistic logic

Probabilistic logic is an extension of standard predicate (or propositional) logic which aims to handle uncertainty about actual truth values of formulas. Most common ways to achieve this goal are either specifying a probability that the formula is true or using multi-valued logic. An example of the former approach is the probabilistic logic defined in (Nilsson, 1986 [2]), which is the basis for formalism used in Markov Logic Networks, the main topic of the thesis. The latter approach is usually described in terms of *fuzzy logic* where the truth value of a formula may be any real number in interval [0,1].

The key difference between these two concepts is that in the (Nilsson's) probabilistic logic it is assumed the formula is true with some probability (let's say 0.5), but in the end the formula will eventually be evaluated as strictly true or false. The probability just captures our *belief* about the actual truth value — we are not sure what the value is at first, but once we are, there's no room for any value between true and false and the probabilistic logic becomes a standard 0–1 valued predicate logic. On the other hand it is perfectly valid to state that a truth value of a formula is 0.5 in fuzzy logic as it is built upon fuzzy set theory which extends the set membership function from bivalent to multi-valued, usually being defined as real number in the unit interval (but fuzzy theories with discrete values are also studied) [3].

With multi-valued logic it's possible to formally capture vague or imprecise definitions that naturally arise in everyday language, such as "*Tom is a little old.*" This may be represented as a predicate *old(Tom)*. In the standard predicate logic, we would have to decide if *a little old* is enough to declare this predicate true (maybe after asking for Tom's exact age and comparing it with some threshold), but in fuzzy logic the truth value of *old(Tom)* may be set to some appropriate value such as 0.3, indicating that Tom is not "fully" old yet but he's indeed a little old. With extending the range of possible truth values we also need to redefine behaviour of logic connectives (usually conjunction and implication) and it turns out there is not just one unique way how to do it, but there are actually many well-behaved definitions, each one creating a slightly different variant of fuzzy logic. Examples of some commonly used fuzzy conjunctions are shown in Figure 2.1.

The probabilistic logic as defined by Nilsson introduces a *probability of sentence* and *possible worlds* semantics to incorporate uncertainty about the truth values into the

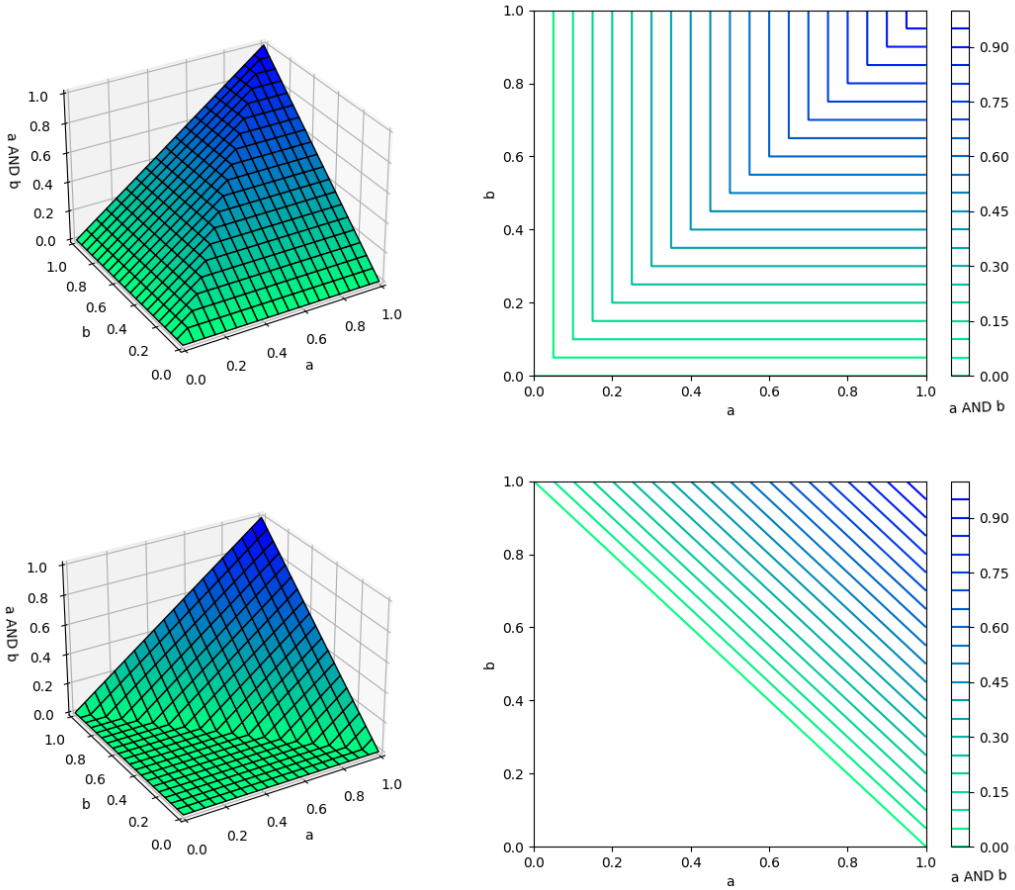

Figure 2.1. Surface and contour plots of two fuzzy conjunction examples which are also *triangular norms* (t-norms). **Upper**: Minimum t-norm $\top_{min} = \min\{a, b\}$. **Lower**: Łukasiewicz t-norm $\top_{Luk} = \max\{0, a + b - 1\}$.

first-order logic. If we consider only one sentence S, the sentence may be either *true* or *false*. This induces two sets of possible worlds — $\mathcal{W}_1$ containing possible worlds where S is true and $\mathcal{W}_2$ containing the worlds where S is false. Then we can reason about the truth value of sentence S in terms of probabilities by specifiying probability p_1 that the actual world is in $\mathcal{W}_1$ (and S is therefore true) and probability $p_2 = 1 - p_1$ that the actual world is in $\mathcal{W}_2$. We can then say that the *(probabilistic) truth value* of sentence S is p_1.

When we incorporate more sentences, the number of sets of possible worlds rises as every set of possible worlds $\mathcal{W}_i$ now represents a distinct combination of truth values assigned to each sentence. For N sentences this may result in up to 2^N sets of possible worlds, but usually their total count is lower as some combinations are logically inconsistent and therefore define an *impossible world* (e.g. S_1 true, S_2 true but $S_3 = S_1 \wedge S_2$ false). The set of consistent possible worlds is then considered a sample space over which a probability distribution is defined. For every set of possible worlds $\mathcal{W}_i$ a probability p_i specifies the probability that the actual world is in $\mathcal{W}_i$. As the sets of possible worlds are exclusive and exhaustive, all p_i sum to 1. The probabilistic truth value of a sentence S is then simply defined as a sum of probabilities of all sets of possible worlds where S is true. Analogically the logical entailment of sentence S from set of sentences $\mathcal{B}$ ($\mathcal{B} \vdash S$) is generalized as the *probabilistic entailment* which is the probability that S is true given the probabilities of sentences in $\mathcal{B}$ (set of beliefs).

Now suppose there are N sentences $S_1, S_2, ...S_N$ which together specify K sets of consistent possible worlds, denote the probabilistic truth values of sentences as a column vector $\Pi = [\pi_1, \pi_2, ..., \pi_N]$, denote the probability distribution over the possible worlds as $P = [p_1, p_2, ...p_K]$ and denote the actual truth values of sentences associated with each possible world as matrix V of dimensions $N \times K$, where element v_{ij} represents the truth value of sentence S_i in set of possible worlds $\mathcal{W}_j$. Note that each column of V there represents one set of possible worlds. Calculation of the probabilistic truth values of all sentences then may be concisely represented as a matrix equation

$$\Pi = VP \tag{2.1}$$

As a concrete example consider a theory with three sentences (taken from Nilsson's original article [2])

- $S_1 = \forall x : \ P(x)$,
- $S_2 = \forall x : \ P(x) \Rightarrow Q(x)$,
- $S_3 = \forall x : \ Q(x)$.

The sentences define 4 distinct sets of possible worlds with following combinations of consistent truth values:

	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_3$	$\mathcal{W}_4$
$S_1 = \forall x : \ P(x)$	*true*	*true*	*false*	*false*
$S_2 = \forall x : \ P(x) \Rightarrow Q(x)$	*true*	*false*	*true*	*true*
$S_3 = \forall x : \ Q(x)$	*true*	*false*	*true*	*false*

Table 2.1. Consistent combinations of truth values of sentences in possible worlds.

Translation of the table to matrix V is straightforward and omitted. Instead we'll focus on possible range of the truth values π_i. As we see from Equation (2.1), the value of Π depends on probabilities of possible worlds P. Now consider at first the extremal case where exactly one possible world achieves probability 1 and the probability of the rest is 0. This obviously results in Π being equal to the column of V corresponding with the currently selected set of posible worlds. We can then proceed with modifying probabilities p_i which in turn changes the outcome of all π_i. The probabilities p_i are however also constrained as their sum must be 1, so the actual attainable truth values pi_i are convex combinations of those achieved for extremal distributions of p_i. This is visualized in Figure 2.2. In this geometrical interpretation the extremal values are vertices of a polytope and all attainable truth values of the sentences lie inside or on boundaries of the polytope.

Figure 2.2 also shows that it is not straightforward to just arbitrarily set values of π_i independently on each other, as their consistent combinations are restricted by the polytope. This doesn't pose a problem in case when the calculation proceeds exactly in the direction of Equation (2.1) and the probability distribution of possible worlds is already specified, because the equation guarantees the result Π will be consistent. In practise however the reasoning often works the other way around — the probabilities of some sentences are assigned first (e.g. as an input from some expert), the sentences then form the knowledge base, and the goal is to find the probabilities of the other sentences, i.e. to evaluate a probabilistic entailment of the sentences with unspecified probabilities from those in the knowledge base. In this setting the actual probability

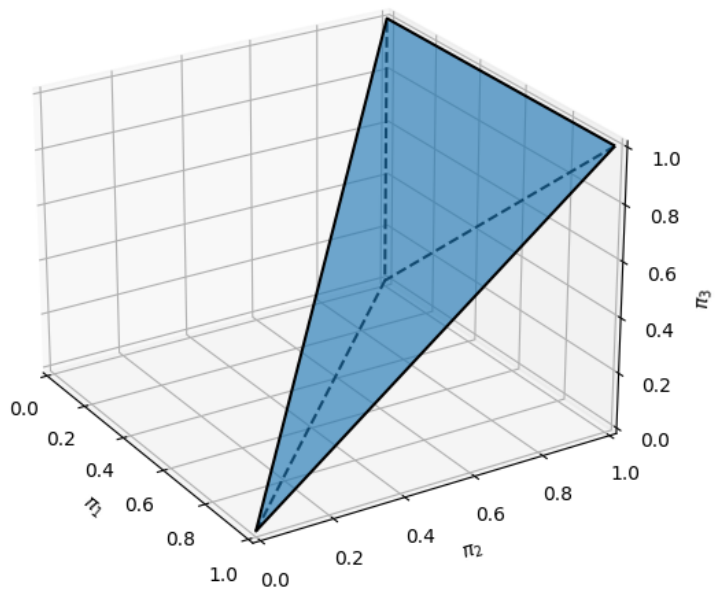

Figure 2.2. Polytope representing consistent truth values for a set of sentences $S_1 = \forall x : P(x)$, $S_2 = \forall x : P(x) \Rightarrow Q(x)$ and $S_3 = \forall x : Q(x)$ (the image is a rotated remake of Fig. 2 in (Nilsson, 1986 [2], p. 76))

values P of possible worlds may not be even specified in advance as we're just interested in the values of Π.

As an example we will now consider sentences S_1 and S_2 as the knowledge base and we will calculate the truth value of S_3, i.e. perform probabilistic entailment

$$\{\forall x : P(x),\ \forall x : P(x) \Rightarrow Q(x)\} \vdash \{\forall x : Q(x)\}.$$

In accordance with Figure 2.2 we'll assign some consistent truth values to the formulas in the knowledge base, for example $\pi_1 = \pi(S_1) = 0.6$ and $\pi_2 = \pi(S_2) = 0.7$. Then we can use Equation (2.1) to solve for π_3 as following:

1. Add vectors of 1 as the first row into V and Π. This may be interpreted as adding tautology to the knowledge base, but it is also a way to enforce the constraint $\sum p_i = 1$.

$$\begin{bmatrix} 1 \\ \Pi \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ V \end{bmatrix} \cdot P \Rightarrow \begin{bmatrix} 1 \\ 0.6 \\ 0.7 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

2. Eliminate the last rows of V and Π and calculate P from the modified matrices V', Π'. Generally the equation is under-determined (and this holds in our example) as the number of possible worlds is usually higher than the number of sentences present in the probabilistic entailment, therefore we should expect the solution for P will not be unique.

$$\Pi' = V'P \Rightarrow \begin{bmatrix} 1 \\ 0.6 \\ 0.7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

Formally we could proceed with multiplying the equation with left pseudo-inverse of V' but in this trivial case we can caluclate P by solving the system of linear equations:

$$\begin{aligned}
0.6 &= p_1 + p_2 & 0.7 &= p_1 + p_3 + p_4 & 1 &= p_1 + p_2 + p_3 + p_4 \\
p_1 &= 0.6 - p_2 & 0.7 &= 0.6 - p_2 + p_3 + p_4 & 1 &= 0.7 - p_2 \\
& & p_2 &= -0.1 + p_3 + p_4 & 0.3 &= -0.1 + p_3 + p_4 \\
p_1 &= 0.3 & p_2 &= 0.3 & p_3 + p_4 &= 0.4
\end{aligned}$$

3. Enforce non-negativity constraint $p_i \geq 0$ on possible values of P and check that P may actually represent a probability distribution — this may not hold if the initial truth values for sentences in knowledge base were assigned inconsistently. In our example the check passes and we find boundaries for p_3 and p_4 as:

$$p_3 \in [0.0, 0.4],\ p_4 \in [0.0, 0.4]\ , p_3 + p_4 = 0.4$$

4. Denote the last row of V (the one eliminated in step 2) as S. Target probability π_3 then may be calculated as:

$$\begin{aligned}
\pi_3 &= SP \\
\pi_3 &= [\,1 \quad 0 \quad 1 \quad 0\,] \cdot [\,0.3 \quad 0.3 \quad p_3 \quad p_4\,]^T \\
\pi_3 &= 0.3 + p_3 \\
\pi_3 &\in [0.3, 0.7]
\end{aligned}$$

As we can see, the result of the probabilistic entailment is not unique, but gives us only possible bounds on the values of π_3. More intuitive picture of the situation is shown in Figure 2.3, where the calculation is described in a geometric way as finding intersection of the polytope of consistent values with planes $\pi_1 = 0.6$ and $\pi_2 = 0.7$.

If we need to select only one solution, we may calculate π_3 from the probability distribution over the possible worlds with the largest entropy, as this is the one about which we know least prior information [4]. Entropy H of probability distribution $\mathbf{p}$ is defined as [5]:

$$H = -\sum p_i \log p_i$$

Maximization of H could be solved using the method of Lagrange multipliers, however in our example where p_1 and p_2 are already set and the only constraint on p_3 and p_4 is $p_3 + p_4 = 0.4$ we may conclude that the maximal entropy will be reached when $p_3 = p_4$, i.e. $p_3 = 0.2$ and $p_4 = 0.2$. The probabilistic truth value of sentence S_3 for this solution is $\pi_3 = 0.3 + 0.2 = 0.5$.

Following list summarizes the facts about Nilsson's probabilistic logic that were described in this section:

- Calculation of probabilistic truth values may be performed in a form of matrix equations, however as the first step all consistent truth values assignments in the possible worlds must be enumareted, and the complexity of the enumeration grows exponentialy in the nubmer of sentences N.
- Assignment of initial probabilistic truth values π_i to the sentences in the knowledge base must be performed carefully because a random assignment may also be inconsistent.
- Even if the initial assignment of π_i is consistent, the probabilistic entailment usually doesn't provide a unique solution to probability of entailed sentences. In this case we may choose the solution associated with the distribution over possible worlds P having the largest entropy.

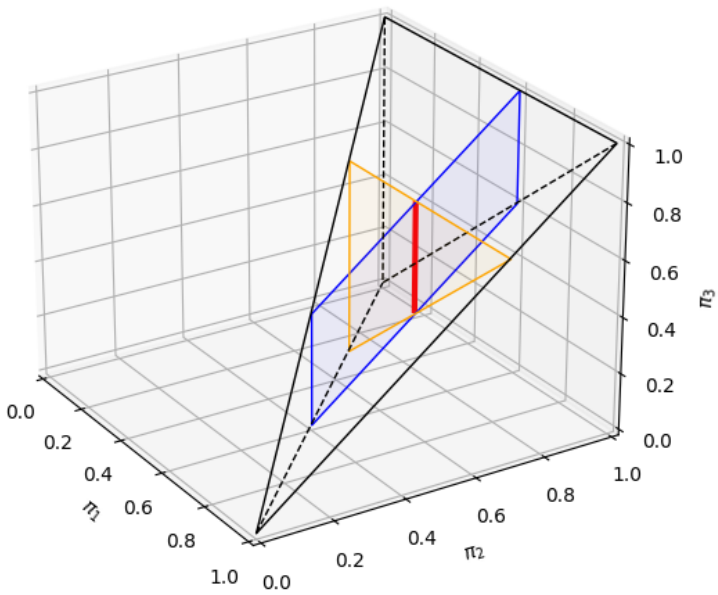

Figure 2.3. Intersection of the polytope from Figure 2.2 with planes $\pi_1 = 0.6$ (blue) and $\pi_2 = 0.7$ (orange). The red segment is the intersection of the planes and the polytope and represents admissible values for π_3 (interval [0.3, 0.7]).

2.2 Probabilistic Graphical Models

This section describes two most commonly employed probabilistic statistical models — the first is a *Bayesian network* and the other one is a *Markov random field* (MRF), sometimes called analogically with the first model a *Markov network*. The models were devised as an approach to encode dependency relations between random variables as a graph and then exploiting this knowledge for an efficient evaluation of random fields and their underlying joint probability distributions, also utilizing methods of the graph theory.

The models are based on the *chain rule* for calculation of joint probability distributions of multiple random variables. The chain rule is a generalization of an observation that the joint probability distribution of two random variables X, Y may be expressed as a product of the marginal probability of one variable and the conditional probability of the other given the first one:

$$\mathrm{P}(X, Y) = \mathrm{P}(X \mid Y) \cdot \mathrm{P}(Y)$$

In order to generalize this observation for multiple random variables we only need to apply the rule for one variable at time, always conditioning on the rest of not-yet entered variables, until the last one is reached:

$$\begin{aligned}
\mathrm{P}(X_1, X_2, ..., X_n) &= \mathrm{P}(X_1 \mid X_2, ..., X_n) \cdot \mathrm{P}(X_2, ..., X_n) && (2.2)\\
&= \mathrm{P}(X_1 \mid X_2, ..., X_n) \cdot \mathrm{P}(X_2 \mid X_3, ..., X_n) \cdot \mathrm{P}(X_3, ..., X_n) && (2.3)\\
&= ...\\
&= \mathrm{P}(X_1 \mid X_2, ..., X_n) \cdot \mathrm{P}(X_2 \mid X_3, ..., X_n) \cdot ... \cdot \mathrm{P}(X_n) && (2.4)
\end{aligned}$$

Actual order of the variables may be of course different as long as the intention of the chain rule is followed. In the thesis we will also use a shorthand notation $p(x_1, x_2, ...x_n)$

for probability of actual assignment of values to random varibles (analogically also for conditional probabilities):

$$p(x_1, x_2, ...x_n) = \mathrm{P}(X_1 = x_1, X_2 = x_2, ..., X_n = x_n)$$

Equation (2.4) is a good insight into splitting the calculation of the full joint probability distribution into the number of more tractable factors which could be represented by smaller probability tables or functions with less variables than the ones for the full joint probability. However applying the chain rull exactly in the form of Equation (2.4) doesn't actually considerably reduce the complexity. If we consider discrete random variables and denote the size of the largest domain of values for any X_i as K, evaluation of the left hand side requires construction of a probability table with $\mathcal{O}(K^N)$ elements, while evaluating first expression on the right hand side requires construction of a conditional probability table for (up to) K possible values of X_1 conditioned on $\mathcal{O}(K^{N-1})$ values for the rest of variables, i.e. the time complexity generally remains the same $\mathcal{O}(K^N)$.

The key problem in evaluating the Equation (2.4) is that each variable is conditioned on all remaining variables, while in practice most of the remaining variables influence the value of the conditional probability only negligible or not at all. This is captured in the concept of *conditional independence* [6].

Definition 2.1. (Conditional independence) *Two random variables A, B are conditionally independent given a random variable C (denoted $A \perp\!\!\!\perp B \mid C$) if and only if they are independent in their conditional probability distribution given C for all possible values of A,B,C:*

$$\mathrm{P}(A, B \mid C) = \mathrm{P}(A \mid C) \cdot \mathrm{P}(B \mid C) \tag{2.5}$$

The defition of conditional independence may be equivalently rephrased as follows — if we're given conditional probability $\mathrm{P}(A \mid C)$ and know $A \perp\!\!\!\perp B$, observing B has no effect on the value of the conditional probability, that is:

$$A \perp\!\!\!\perp B \mid C \;\Leftrightarrow\; \mathrm{P}(A \mid B, C) = \mathrm{P}(A \mid C) \tag{2.6}$$

Conditional independence may be also generalized for sets of random variables — actually it is more or less sufficient just to interpret random variables A, B, C in Definition 2.1 as sets of random variables. Equation (2.6) then may be used to simplify factors of the joint probability distribution if we can efficiently represent conditional (in)dependencies between variables, because as the equation suggests, all conditionally independent variables then may be ignored and the conditional probability tables may be calculated only w.r.t. conditioning variables. As an example, we may simplify calculation of the probability of X_1 in Equation (2.4) if we know that X_1 is conditionally independent on all other variables given X_2, X_5 as

$$\begin{aligned}\mathrm{P}(X_1, X_2, ..., X_n) &= \mathrm{P}(X_1 \mid X_2, ..., X_n) \cdot \mathrm{P}(X_2 \mid X_3, ..., X_n) \cdot ... \cdot \mathrm{P}(X_n) \\ &= \mathrm{P}(X_1 \mid X_2, X_5) \cdot \mathrm{P}(X_2 \mid X_3, ..., X_n) \cdot ... \cdot \mathrm{P}(X_n)\end{aligned}$$

The process then may be similarly repeated for conditional probability of X_2 and another random variables present in the equation.

As a last point before proceeding to the description of two most common probabilistic graphical models — Bayesian networks and Markov networks — we should note that conditional independence of random variables is not related to their standard independence. Two random variables may be independent on each other but conditionally dependent given another variable, and vice versa.

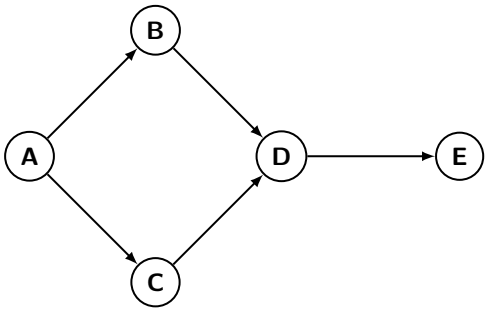

Figure 2.4. Graph of Bayesian network of 5 variables.

For example of two independent variables that become conditionally dependent let's consider rolling two fair six-sided dice, denote the result of the first die A and the result of the other B. As usually in such a case we expect that results of each roll are independent so $\mathrm{P}(A, B) = \mathrm{P}(A) \cdot \mathrm{P}(B)$. However, when we also observe variable C which checks if sum of rolls is even or odd, A and B become conditionally dependent given C — knowing that the sum of rolls is even doesn't provide any additional information without also knowing the result of the other die, so the conditional probability is equal to the marginal (same applies to $\mathrm{P}(B \mid C)$):

$$\mathrm{P}(A = a \mid C = c) = \mathrm{P}(A = a) = \frac{1}{6}.$$

However if we know that $C = even$ and $A = 3$, then we see that B must be also odd, so if we take even value $B = 2$, Equation (2.5) doesn't hold and therefore $A \not\perp B \mid C$:

$$\mathrm{P}(A = 3 \mid C = even) \cdot \mathrm{P}(B = 2 \mid C = even) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36} \neq$$
$$\neq \mathrm{P}(A = 3, B = 2, C = even) = 0$$

2.2.1 Bayesian Networks

Bayesian network is a directed acyclic graph (DAG) where vertices represent variables of interest (random variables, parameter models, hypotheses) and oriented edges represent conditional dependencies between the variables; oriented edge $X_u \to X_v$ specifies that X_v is conditionally dependent on X_u. Edge direction however primarily captures the real causal connections and not the actual direction used for computations, because the information necessary for reasoning can still be propagated in both ways [7].

The most important property of Bayesian networks is that every vertex X is independent from its non-descendants given set of its parent vertices Pa_X. Computation of the marginal probability of variable X is then conditioned on the parent nodes and only requires knowledge of their probabilities:

$$\mathrm{P}(X) = \mathrm{P}(X \mid Pa_X) \tag{2.7}$$

Probabilities of parent nodes are usually stored in the child node in a form of conditional probability table. Provided the number of parents for each node is bounded, the number of required conditional distributions for each node grows only linearly in the size of the Bayesian net, which is a considerable improvement over exponential growth for Equation (2.2).

Computation of full joint probability distribution in the Bayesian net is factorized into product of conditional distributions conditioned on parent nodes:

$$\mathrm{P}(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} \mathrm{P}(X_i \mid Pa_{X_i})$$

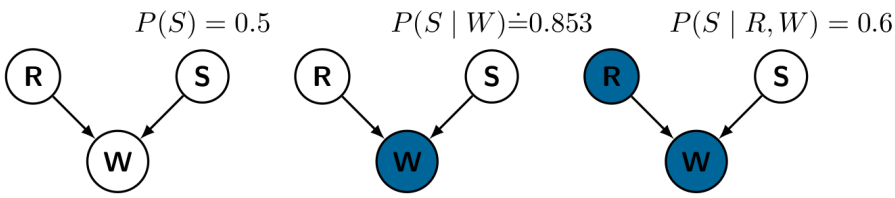

Figure 2.5. Illustration of Bayesian net described in the calcualtion example. **R** represents raining, **S** sprinkler and **W** a wet pavement. Initial situation is captured in the left graph, in the central graph we observe the pavement is wet which influences marginal probabilities of both **R** and **S**. In the right graph we find out that it was actually raining, but this information also affects our knowledge about **S**, because they become dependent after observing **W**.

Let's take as an example the Bayesian network presented in Figure 2.4. The joint probability distribution of the network may be expressed as:

$$\mathrm{P}(A, B, C, D, E) = \mathrm{P}(A) \cdot \mathrm{P}(B \mid A) \cdot \mathrm{P}(C \mid A) \cdot \mathrm{P}(D \mid B, C) \cdot \mathrm{P}(E \mid D)$$

More illustrative example which will also point to a not so obvious property of Bayesian networks is illustrated in Figure 2.5. In the morning, we may observe that the pavement in front of the house is wet. There are two possible causes for this — it may have been raining during the night or early in the morning the sprinkler on the grass was on. The sprinkler should be watering the grass every morning, but it is faulty and works more or less randomly. It also doesn't have any detector to check whether the grass is already wet, so it may also turn on even if it was raining. We have these prior probabilities for the sprinkler (S) and the raining (R):

$$\mathrm{P}(S = on) = 0.5$$
$$\mathrm{P}(R = true) = 0.2$$

The conditional probabilities for observing wet pavement (W) given the other two events are stated as follows:

$$\mathrm{P}(W = wet \mid S = on, R = true) = 0.9$$
$$\mathrm{P}(W = wet \mid S = on, R = false) = 0.7$$
$$\mathrm{P}(W = wet \mid S = off, R = true) = 0.6$$
$$\mathrm{P}(W = wet \mid S = off, R = false) = 0.01$$

Now in the morning we actually observe the pavement is wet and we may want to evaluate the posterior probability that the sprinkler was on. This may be done using Bayes' theorem:

$$\mathrm{P}(S \mid W) = \frac{\mathrm{P}(W \mid S) \cdot \mathrm{P}(S)}{\mathrm{P}(W)} \tag{2.8}$$

The denominator is evaluated by marginalizing over R, S:

$$\begin{aligned}\mathrm{P}(W = wet) &= \sum_{s \in \{on, off\}} \sum_{r \in \{true, false\}} \mathrm{P}(W = wet \mid S = s, R = r) \cdot \mathrm{P}(S = s) \cdot \mathrm{P}(R = r) \\ &= 0.434\end{aligned}$$

Similarly for conditional probability $\mathrm{P}(W \mid S)$:

$$\mathrm{P}(W = wet \mid S = on) = \sum_{r \in \{true, false\}} \mathrm{P}(W = wet \mid S = on, R = r) \cdot \mathrm{P}(R = r) = 0.74$$

So plugging all the numbers into Equation (2.8) we get:

$$\mathrm{P}(S = on \mid W = wet) = \frac{0.74 \cdot 0.5}{0.434} \doteq 0.853$$

We see that $\mathrm{P}(S = on \mid W = wet) > \mathrm{P}(S = on)$ so observing that the pavement is wet makes it more likely that it the sprinkler was on, which is something we would intuitively expect. Now let's see if something changes when we find out that it was raining in the night (e.g. from a weather report). The posterior probability for the sprinkler changes to:

$$\mathrm{P}(S \mid W, R) = \frac{\mathrm{P}(W \mid S, R) \cdot \mathrm{P}(S) \cdot \mathrm{P}(R)}{\sum_{s \in S} \mathrm{P}(W \mid S, R) \cdot \mathrm{P}(S) \cdot \mathrm{P}(R)}$$

$$\mathrm{P}(S = on \mid W = wet, R = true) = \frac{\mathrm{P}(W = wet \mid S = on, R = true) \cdot \mathrm{P}(S = on)}{\sum_{s} \mathrm{P}(W = wet \mid S = s, R = true) \cdot \mathrm{P}(S = s)} = 0.6$$

After observing that it was raining the probibility that sprinkler was on drops, even though initially these two variables were independent. They however became coupled when we observed the actual value of their common child.

As we can see from the previous example, even though the Bayesian network is a directed graphical model, the information may still flow in any direction when reasoning and evidence provided in the descendant node actually influenced the marginal probability of the parent node. Earlier in the beginning of the section it was declared that a node is conditionally independent from its non-descendants given its parents. This is indeed true, but we may be actually also interested in which nodes actually separate the node from the rest of the network, so we know which nodes may influence reasoning about the node and which are irrelevant.

Identification of separating set of nodes may be defined in terms of *d-separation*, which is based on a notion of *active paths*. First we should consider what configuration of nodes w.r.t. directed edges may be observed over triplets of nodes [8]:

1. *Cascade*: $A \rightarrow B \rightarrow C$ or $A \leftarrow B \leftarrow C$
   If B is observed, then $A \perp\!\!\!\perp C \mid B$, because we can determine output of C solely on B and A doesn't influence it. If B is unobserved, then $A \not\perp\!\!\!\perp C$, because observing A provides information about B and in turn we may also reason about C.
2. *Common parent*: $A \leftarrow B \rightarrow C$
   Reasoning is actually the same as above — if B is observed, $A \perp\!\!\!\perp C \mid B$, otherwise $A \not\perp\!\!\!\perp C$.
3. *V-structure*: $A \rightarrow B \leftarrow C$
   The results in this case are opposite to previous ones — if the common descendant B is *unobserved*, then parents are independent — $A \perp\!\!\!\perp C$. But when B is observed, then $A \not\perp\!\!\!\perp C \mid B$. This is also called *explaining away*.

These checks may be recursively applied on larger sets of variables in the graph, leading to a notion of *active paths* in Bayesian network. An undirected path in the Bayesian network is active given a set of observed variables O if for every consecutive triple of variables X, Y, Z one of the following holds:

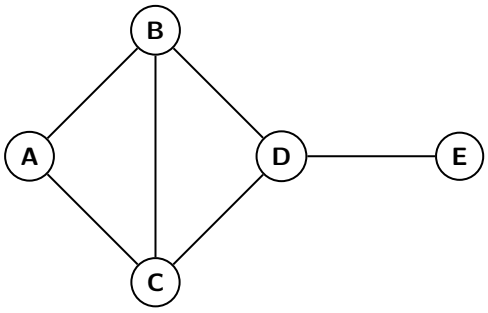

Figure 2.6. Graph of Markov random field of 5 variables with two 3-cliques $\{A, B, C\}$ and $\{B, C, D\}$ and one 2-clique $\{D, E\}$.

- $X \to Y \to Z$ and Y is unobserved ($Y \notin O$),
- $X \leftarrow Y \leftarrow Z$ and Y is unobserved,
- $X \leftarrow Y \to Z$ and Y is unobserved,
- $X \to Y \leftarrow Z$ and Y is or *any of its descendants* is observed.

The independence of sets in Bayesian networks is then specified using *d-separation*. Two sets of variables A, B are d-separated given set O if there is no active path connecting A and B given O. Then set O is also a separating set of sets A, B. Separating set is not actually unique — adding a variable which is not in A or B into the separating set still yields a separating set. The minimal separating set is a separating set from which no variable can be removed without violating d-separation property. In Bayesian networks, the minimal separating set for a variable from the rest of graph consists from variable's parents, its immediate children and all other parents of these immediate children.

2.2.2 Markov Random Fields

Markov random field (MRF) or *Markov network* is a graphical probabilistic model that represents dependencies between variables as an undirected graph. An MRF may be also cyclic, therefore it may, unlike Bayesian networks, conveniently represent cyclic dependencies. Also the notion of separating set for a node is simpler in MRFs as it consists only from all neighbours of the node in question [9].

If graph $G = (V, E)$ represents an MRF, it must satisfy following three Markov properties, ordered from the weakest to the strongest (variable represented by vertex v is denoted as X_v) [10]:

1. **Pairwise Markov property**:
   Any two non-adjacent variables are conditionally independent given all other variables:
   $$X_v \perp\!\!\!\perp X_u \mid X_{V \setminus \{u,v\}}$$
2. **Local Markov property**:
   A variable is conditionally independent of all other variables given its neighbors:
   $$X_v \perp\!\!\!\perp X_{V \setminus N[v]} \mid X_{V \setminus N(v)}$$
   where $N(v)$ is the set of neighbors of v and $N[v] = v \cup N(v)$ is the closed neighbourhood of v.
3. **Global Markov property**:
   Any two subsets of variables are conditionally independent given a separating subset:
   $$X_A \perp\!\!\!\perp X_B \mid X_S$$

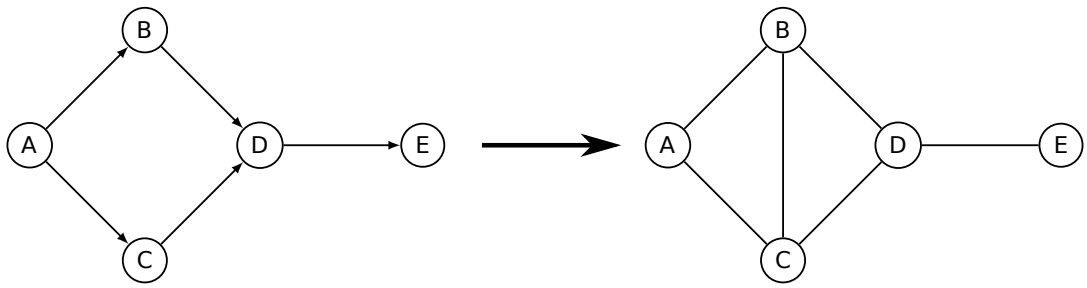

Figure 2.7. Moralization of a Bayesian network (left) into a Markov random field (right).

where X_A, X_B are sets of vertices and X_S is their separating subset (i.e. all paths between a node from X_A to a node in X_B pass through a node in X_S).

All three Markov properties are actually equivalent if the underlying probability distribution induced by variables in the graph is strictly positive.

Computation of the full joint probability distribution in MRFs can be factorized similarly to Bayesian networks as a product of quantities over sets of variables. Unlike the Bayesian networks the quantity is not represented in a form of probability tables, but as a *potential function*. The factorization is then performed over maximal cliques of a graph (graph clique is a fully-connected subgraph of the graph[1]):

$$p(x_1, x_2, ..., x_n) = \frac{1}{Z} \prod_{C \in cl(G)} \phi(C),$$

where $cl(G)$ is the set of maximal cliques of graph G, $\phi(C)$ is a potential function associated with assignments to all variables (vertices) in clique C, and Z is the partition function. This function ensures that the result is actually a probability distribution by summing potential functions for all possible configurations of MRF:

$$Z = \sum_{x_1, x_2 ... x_n} \prod_{C \in cl(G)} \phi(C)$$

As an actual example we show factorization of MRF presented in Figure 2.6, the set of maximal cliques is $cl(G) = \{\{A, B, C\}, \{B, C, D\}, \{D, E\}\}$ (note that if there was an edge connecting A, D the 3-cliques would be replaced with a 4-clique $\{A, B, C, D\}$) and the probability of a configuration factorizes into:

$$p(a, b, c, d, e) = \frac{1}{Z} \cdot \phi(a, b, c) \cdot \phi(b, c, d) \cdot \phi(d, e)$$

Two problems however arise when we try to perform exact inference in MRFs. The first one is that listing all maximal cliques in the graph is NP-complete problem (it is also listed in Karp's 21 NP-complete problems in formulation where we try do detect any clique of size k [11]). This may be overcome by the fact that the structure of MRFs is usually not random, but it is crafted intentionally, so the structure of maximal cliques is usually known beforehand and it is not needed to detect them. The other problem is that evaluating the partition function requires summing over all possible assignments, which is in general NP-hard. This problem may not be overcome so easily and so the exact inference in MRFs is generally intractable, even though there are classes of MRFs that may be computed efficiently.

[1] We may prepend a clique with a number of vertices present in it, i.e. 3-clique, 4-clique etc. 2-clique is an edge and 1-clique is just a vertex

There are also procedures to transform Bayesian networks into MRFs and vice versa. As a first step in transforming Bayesian network into MRF we only need to trivially substitute every directed edge with an undirected one. As a second step we need to add an edge between all vertices, which share a direct descendant and are disconnected in the Bayesian network. This is called *moralization* as it enforces a relation between parent nodes (a "marriage", though it may easily result in a polygamy if the node has more than 2 parents). If the second step is omitted, we lose information that the value of the child node is actually dependent on values of all its parents simultaneously. The procedure is illustrated in Figure 2.7. Potential functions for each clique then correspond to joint probability of all variables in the clique, which may be in turn calculated from the conditional probability table associated with the leaf node of the clique by Equation (2.7). The partition function of such a transformed net is trivially 1 (as all probabilities in the Bayesian network must sum to 1). The converse process of transforming an MRF into a Bayesian net is called *triangulation*, but is seldom used, as it is usually intractable (it often results in an almost fully connected DAG).

Chapter 3
Markov Logic Networks

This chapter describes *Markov logic networks* (MLN), a probabilistic logic framework used in the statistical relational learning (SRL). Markov logic networks encode statistical regularities in a from of weighted logical formulas. The following section provides definitions of MLNs and related concepts, then basic properties, means of inference and standard learning tasks in MLNs are discussed. Finally we'll focus on the key concept of the thesis — *relational marginal polytope* which originates from relational marginal problem — a task concerned with finding the maximum-entropy probability distribution satisfying specified marginal probabilities.

3.1 Definition

The concept of Markov logic networks first appeared in the paper of Richardson and Domingos in 2006 [12]. The rationale behind their proposal is that when we model a problem using first-order logic formulas (these form a knowledge base), the formulas are actually hard-constraints and any potential world that violates just one of them is consequently impossible. This behaviour however may not be always desirable as often a formula that doesn't hold in all cases may still capture useful information about modelled relationships. In order to soften the constraint checking a weight is associated with each formula. The weight should represent how important the constraint is in the model — the higher the wieght, the higher the importance of the constraint. In this setting the world violating a constraint doesn't become instantly impossible, only less probable. If the world violates higher number of constraints or if it violates more important ones, the world's probability decreases proportionally.

Definition 3.1. (Markov logic network): *A Markov logic network (MLN) is a set of weighted first-order logic formulas* (α, w) *where* $w \in \mathbb{R}$ *and* α *is function-free and quantifier-free first-order logic formula.*

MLN Φ induces a probability distribution over a set of possible worlds Ω:

$$\text{for } \omega \in \Omega: \; p_\Phi(\omega) = \frac{1}{Z} \exp\left(\sum_{(\alpha,w)\in\Phi} w \cdot N(\alpha,\omega) \right) \tag{3.1}$$

In this equation $p_\Phi(\omega)$ denotes probability of observing possible world ω, $N(\alpha,\omega)$ is total number of groundings of formula α that are satisfied in ω relative to a finite set of constants Δ (called the domain) and Z is the *partition function* that normalizes the result so it forms a probability distribution similarly as in MRFs. Presence of the normalizing term Z draws exact inference in MLNs generally intractable in the same way as in MRFs, as its evaluation requires summation over all possible worlds whose number is exponential in the size of domain.

An MLN can be created from a first-order logic knowledge base just by assigning arbitrary weights to each formula in the KB. The first-order logic is actually a special case of MLN where all weights are infinite, i.e. any violation of a formula renders

the associated world impossible. The probability distribution over satisfiable possible worlds in this case is uniform. The weight of the formula can be interpreted as a log-odd between observing a world where the formula holds and a world where it doesn't, assuming all remaining weights are equal.

3.1.1 Relation to MRF

Markov logic networks are closely related to Markov random fields — grounding an MLN with respect to a domain results in an instance of a MRF and in this sense MLNs may be considered templates for a variety of MRFs. The resulting MRFs may vary significantly in size but they will share common structures. The procedure for grounding MLN into MRF was described in the initial paper by Richardson and Domingos [12]). An instance of MRF $M_{\Phi,\Delta}$ may be grounded from MLN Φ with respect to the domain Δ this way:

1. $M_{\Phi,\Delta}$ contains one binary node for each possible grounding of each predicate appearing in Φ. The value of the node is 1 if the ground atom is true, and 0 otherwise.
2. $M_{\Phi,\Delta}$ contains one feature for each possible grounding of each formula α_i in MLN Φ. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_i associated with α_i in MLN Φ.

3.2 Inference

Exact inference in MLNs is in general intractable for similar reasons as in MRFs — the partition function Z is calculated as a sum of terms over all possible worlds, and the number of all possible worlds in general grows exponentially w.r.t the size of domain $|\Delta|$.

Calculation of the partition function may be converted to the *weighted first-order model count* problem (WFOMC)[13]:

Definition 3.2. (WFOMC): *Let $w(P)$ and $\overline{w}(P)$ be functions from predicates to real numbers (w and $\overline{w}$ are called weight functions) and let Φ be a first-order theory. Then*

$$\mathrm{WFOMC}(\Phi, w, \overline{w}) = \sum_{\omega \in \Omega : \omega \models \Phi} \prod_{a \in \mathcal{P}(\omega)} w(Pred(a)) \prod_{a \in \mathcal{N}(\omega)} \overline{w}(Pred(a))$$

where $\mathcal{P}(\omega)$ and $\mathcal{N}(\omega)$ denote the positive literals that are true and false in ω, respectively, and Pred(a) denotes the predicate of a (e.g. Pred(friends(Alice, Bob)) = friends).

The evaluation of WFOMC then proceeds with addition of a formula ξ_i for every weighted formula (α_i, w_i) in Φ whose free variables are exactly $x_1, x_2, ...x_k$:

$$\forall x_1, ..., x_k : \xi_i(x_1, ..., x_k) \Leftrightarrow \alpha_i(x_1, ..., x_k)$$

Then we set $w(\xi_i) = \exp(w_i)$, $\overline{w}(\xi_i) = 1$ for all new predicates and $w(\alpha_i) = 1$ and $\overline{w}(\alpha_i) = 1$ for the original predicates. If we denote the resulting set of predicates Γ, it will turn out that actually $\mathrm{WFOMC}(\Gamma, w, \overline{w}) = Z$. WFOMC may be also easily used for evaluation of the marginal probability of query q under Γ:

$$\mathrm{P}_{\Phi,\Omega}(q) = \frac{\mathrm{WFOMC}(\Gamma \cup \{q\}, w, \overline{w})}{\mathrm{WFOMC}(\Gamma, w, \overline{w})}$$

The WFOMC however doesn't change asymptotical complexity of computation of the partition function w.r.t. the domain (it remains exponential). However there are

classes of MLNs where inference may be performed more efficiently, in polynomial time w.r.t. to the size of the domain. These problems are called *domain liftable*.

Definition 3.3. (Domain liftability) *An algorithm for computing* WFOMC *is said to be domain-liftable if it runs in time polynomial in the size of the domain.*

Example of domain-liftable MLN instances are MLNs where each predicate contains at most two variables [14].

3.3 Relational marginal polytopes

This section introduces *relational marginal polytopes* (RMP) with which approximation the thesis is mainly concerned. RMPs emerge as a set of feasible solutions to the *relational marginal problems* which try to find weights for a maximum-entropy distribution over the possible worlds w.r.t. statistical marginal probabilities of formulas in the MLN.

3.3.1 Relational marginal problem

The total number of satisfiable formula groundings $N(\alpha, \omega)$ in Equation (3.1) presents the absolute number of admissible groundings. It may be however more convenient to express this quantity relative to the size of the number of possible groundings. This quantity is called *formula statistic* w.r.t. the possible world ω:

Definition 3.4. (Formula statistic) *Let α be a quantifier-free first-order logic formula with k variables $\{x_1, ...x_k\}$. Its formula statistic w.r.t. a possible world ω is defined as*:

$$Q_\omega(\alpha) = \left(\frac{|\Delta|}{k}\right)^{-1} \cdot (k!)^{-1} \cdot N(\alpha, \omega) \tag{3.2}$$

There $|\Delta|$ denotes size of the domain and k denotes arity of predicate α. Intuitively the formula statistic represents the probability that a random injective substitution of variables that ground formula α will be satisfied in the possible world ω if we draw the substitution randomly from the uniform distribution.

With notion of formula statistics, we may continue with a defintion of the *relational marginal problem*.

Definition 3.5. (Relational marginal problem): *The relational marginal problem is a convex optimization with the following formulation*:

$$\min \sum_{P_\omega : \omega \in \Omega} P_\omega \log P_\omega \quad \text{s.t.} \tag{3.3}$$

$$\forall i : 1, ..., l : \sum_{\omega \in \Omega} P_\omega \cdot Q_\omega(\alpha_i) = \theta_i \tag{3.4}$$

$$\forall \omega \in \Omega : P_\omega \geq 0, \sum_{\omega \in \Omega} P_\omega = 1 \tag{3.5}$$

where P_ω denotes the probability of possible world ω, $Q(\alpha_i)$ is formula statistic associated with formula α_i in the particular possible world, and $\theta_1, ...\theta_k$ are the target expected values for each formula statistics, also called the *relational marginals* (hence the name of the task).

To provide a more thorough analysis of the formulation — Equation (3.3) minimizes *negative* entropy of the probability distribution over the possible worlds, Equation (3.4) represents constraints specified by the relational marginals and the last Equation (3.5)

ensures the result of the task is a probability distribution. Assuming strictly positive solution, the optimal solution is:

$$P_\omega = p_\Phi(\omega) = \frac{1}{Z} \exp\left(\sum_{(\alpha_i,\lambda_i)\in\Phi} \lambda_i \cdot Q_\omega(\alpha) \right)$$

where λ_i are obtained by maximizing dual criterion which is incidentally MLN's log-likelihood w.r.t. some training example whose statistics are equal to expected ones:

$$L(\lambda) = \sum_{\alpha_i} \lambda_i \cdot \theta_i - \log \sum_{\omega in \Omega} e^{\sum_{\alpha_i} \lambda_i \cdot Q_\omega(\alpha_i)}$$

Due to the duality if we're able to efficiently solve relational marginal problems, we can also efficiently solve maximum likelihood estimation of MLN. However in order to compute values of λ_i we have to calculate the gradient of L which involves computation of the partition function. Solving the relational marginal problem is therefore as hard as evaluating the partition function, which is generally #P-hard.

3.3.2 RMP Definitions

When solving relational marginal problems it is possible to encounter a relational marginals that define expected values of formula statistics which are actually not realizable on the domain of the specified size (or on a domain of any size at all). Consider following example, which describes edges and triangles present in a graph in terms of propositional logic [14]:

Example 3.6. : Consider a MLN Φ consisting of following formulas (weight omitted):

- $\phi : edge(x_1, x_2)$,
- $\psi : edge(x_1, x_2) \wedge edge(x_2, x_3) \wedge (x_1, x_3)$.
- $\Delta = \{c_1, c_2 ... c_{100}\}$

Now when considering expected values of formula statistics $\mathbb{E}[(Q_\omega(\phi))] = 0$ and $\mathbb{E}[(Q_\omega(\psi))] = 0.5$, we can easily see that no possible world can conform to this distribution as there simply cannot be even one triangle in a graph without edges. Values of statistics corresponding to some actual probability distributions form so called *relational marginal polytope* [1]:

Definition 3.7. (Relational marginal polytope): *Let Ω be the set of possible worlds on domain Δ and $\Phi = (\alpha_1, ..., \alpha_m)$ be a list of formulas. The relational marginal polytope* **RMP**(Φ, Δ) *w.r.t. Φ is defined as*:

$$\textbf{RMP}(\Phi, \Delta) = \{\exists \text{ distribution on } \Omega \text{ s.t. } \mathbb{E}[Q(\alpha_1, \omega)] = x_1 \wedge ... \wedge \mathbb{E}[Q(\alpha_m, \omega)] = x_m\}.$$

Relational marginal polytopes form w.r.t. list of formulas $(\alpha_1, ..., \alpha_l)$ a convex hull of a set:

$$\{(Q_\omega(\alpha_1), ..., Q_\omega(\alpha_l)) \mid \omega \in \Omega\}.$$

Important property of RMPs is that RMPs associated with larger domains are *subsets* of RMPs associated with domains with less elements. Furthermore, using a notion of *η-interiority* a bound can be provided on the maximal difference between any points in these polytopes.

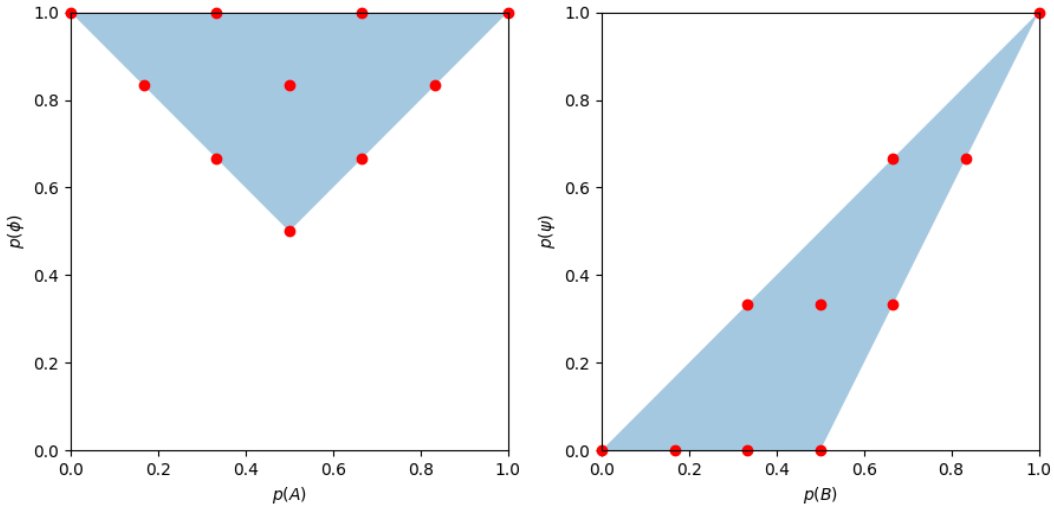

Figure 3.1. Examples of RMP w.r.t. domain of size 3 for two MLNs (both under unique names assumption). Blue area represents RMP, red points denote actual formula statistics Q that can be achieved in the MLN. **Left** $A = a(X,Y)$, $\phi = a(X,Y) \vee \neg a(Y,X)$. **Right** $B = b(X,Y)$, $\psi = b(X,Y) \wedge b(Y,X)$.

Definition 3.8. (η-interiority [14]): *Let $\eta > 0$, $\mathbf{P}$ be a polytope and $A^{=}\mathbf{x} = \mathbf{c}$ be the maximal linearly independent system of linear equations that hold for the vertices of $\mathbf{P}$. A point θ is said to be in the η-interior of $\mathbf{P}$ if $\{\theta' | A^{=}\theta' = \mathbf{c}, \| \theta' - \theta \| \leq \eta\} \subseteq \mathbf{P}$.*

Equivalently point y is in η-interior of polytope $\mathbf{P}$ if there a ball with radius η centered in y is subset of the polytope. Regardless on the definition we use, detecting if a point is in η-interior of RMP is NP problem.

Sometimes it is more convenient use an *integer relational marginal polytope*, which is a convex hull of all realizable groundings count:

$$\mathbf{IRMP} = \{N(\alpha_1, \omega), ..., N(\alpha_m, \omega) : \omega \in \Omega\} \tag{3.6}$$

Both types of relational polytopes are interchangeable as there is a straightforward relation between number of groundings and the formula statistics:

$$Q(\alpha_i, \omega) = |\Delta|^{-|vars(\alpha_i)} \cdot N(\alpha_i, \omega)$$

Chapter 4
Implementation

This sections describes programmatical implementation of the thesis.

4.1 Realizability of statistics

Realizability of expected formula statistics for a domain of specified size may be checked by integer linear program. As only the feasibility of constraints is checked, the program doesn't actually perform any optimization, so the constant is used as the objective function. Also total number of satisfied formula groundings $N(\alpha, \omega)$ with respect to domain size must be used instead of formula statistics $Q_\omega(\alpha)$. The program expects as input a size of domain $|\Delta|$, a list of function-free quantifier-free first-order formulas Φ in CNF, and an expected number of groundings N_i for each formula $\alpha_i \in \Phi$.

For the formulation of the ILP we also define sets:

- A – set of all grounded atoms
- $Lit^+_{\nu,\alpha}$, $Lit^-_{\nu,\alpha}$ – sets of all positive/negative ground literals created by a substitution ν from formula α
- $Cl_{\nu,\alpha}$ – set of all clauses created by a substitution ν from formula α.

The formulation of the ILP is as follows:

$$\max 0 \quad \text{s.t.}$$

$$\forall \text{ ground atoms } a_i \in A: \ a_i \in \{0,1\}, \ l_i^+ = a_i, \ l_i^- = 1 - a_i$$

$$\forall \text{ clauses, substitution } \nu \ c_{j,\nu,\alpha} \in Cl_{\nu,\alpha}: \ c_{j,\nu,\alpha} = \max\{l \in Lit^+_{\nu,\alpha} \cup Lit^-_{\nu,\alpha}\}$$

$$\forall \text{ formulas, substition } \nu \ f_{k,\nu} \in F_\nu : f_{k,\nu} = \min\{c_{j,\nu,\alpha} | \ c_{j,\nu,f} \in Cl_{\nu,f}\}$$

$$\forall \ F_i : \ N_i = \sum_\nu f_{i,\nu}$$

Definition might look a little bit complicated, but description of the steps actually performed should make it more clear:

1. Binary variable is created for every possible ground atom present in Φ and Φ_0.
2. Another binary variable is created for every positive and negative literals, for positive it is equal to underlying ground atom a, for negative it is $1 - a$.
3. Variables are created for all possible substitutions of clauses, taking maximum value from appropriate positive/negative literal variables (this represents disjunctions of literals in the CNF).
4. Analogically variables representing whole CNF formulas for all possible substitutions are created, but now taking the minimum value of the variables associated with the CNF (this represents conjunctions of clauses in the CNF)
5. Finally a sum of CNF formula variables is set to be equal to expected statistic.

ILP solver (specifically gurobi [15]) checks feasibility of generated constraints and if no violation is found, it also returns an assignment of all variables which in turn

represent one of vaild possible worlds. However it should be noted, that the program doesn't actually perform containment test for underlying IRMP. As is specified in the definition of IRMP (Equation (3.6)), the IRMP is a convex hull of feasible formula grounding counts, therefore the point may be still contained in the IRMP even if it represents infeasible grounding count as we cannot reject the possibility that it is indeed in the convex hull of feasible points. But we can at least conclude that after in case of failure, the point is not a vertex of IRMP.

This model is straightforward, but its performance is not overwhelming. Generally it creates $\mathcal{O}(n^k)$ ground atom variables (where n is domain size and k the highest number of variables in atoms) for every possible substitution and similarly for clauses and formulas. Even though number of variables and constraints remains polynomial in n, their number still grows steadily. We should also note that ILP is an NP-hard problem in general, so we cannot expect that this model will be efficient in general.

Chapter 5
Conclusion

The goals of the thesis were met only partially at most. An exact ILP program for testing feasibility of the marginal problem constraints was implemented in Python using Gurobi solver. This may be a part of actually implemented heuristical algorithm, when an exact solution for a small subset of vertices will be needed.

References

[1] KUŽELKA, Ondřej and Yuyi WANG, 2020. Domain-Liftability of Relational Marginal Polytopes. In: *AISTATS 2020: 23rd International Conference on Artificial Intelligence and Statistics*. Accessible from: `https://arxiv.org/pdf/2001.05198.pdf`

[2] NILSSON, Nils J., 1986. Probabilistic logic. *Artificial Intelligence*. **28**(1), 71-87. DOI: 10.1016/0004-3702(86)90031-7. ISSN 0004-3702.

[3] ZADEH, L.A. Fuzzy sets. *Information and Control*. 1965, **8**(3), 338-353. DOI: 10.1016/S0019-9958(65)90241-X. ISSN 00199958. Accessible from: `http://linkinghub.elsevier.com/retrieve/pii/S001999586590241X`

[4] JAYNES, E. T., 1957. Information Theory and Statistical Mechanics. *Physical Review*. **106**(4), 620-630. DOI: 10.1103/PhysRev.106.620. ISSN 0031-899X. Accessible from: `https://link.aps.org/doi/10.1103/PhysRev.106.620`

[5] SHANNON, C. E., 1948. A Mathematical Theory of Communication. *Bell System Technical Journal*. **27**(4), 623-656. DOI: 10.1002/j.1538-7305.1948.tb00917.x. ISSN 0005-8580. Accessible from: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773067`

[6] DAWID, A. Philip, 1980. Conditional Independence for Statistical Operations. *The Annals of Statistics*. **8**(3), 598-617. DOI: 10.1214/aos/1176345011. ISSN 0090-5364. Accessible from: `http://projecteuclid.org/euclid.aos/1176345011`

[7] PEARL, Judea and Stuart RUSSELL, 2002. Bayesian networks. *The Handbook of Brain Theory and Neural Networks* 2nd Edition. Cambridge (MA): MIT Press. ISBN 9780262011976.

[8] KOLESHOV, Volodymyr and Stefano ERMON, 2020. Bayesian networks. *Stanford CS 228 - Notes* [online]. [cit. 2020-08-14]. Accessible from: `https://ermongroup.github.io/cs228-notes/representation/directed/`

[9] PEARL, Judea, 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* . 1. Amsterdam: Elsevier, 552 pp. DOI: https://doi.org/10.1016/C2009-0-27609-4. ISBN 978-0-08-051489-5.

[10] Markov random field, 2020. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2020-08-14]. Accessible from: `https://en.wikipedia.org/wiki/Markov_random_field#Definition`

[11] KARP, Richard M., 1972. Reducibility among Combinatorial Problems. In: *Complexity of Computer Computations*. Boston, MA: Springer US, 1972, p. 85-103. DOI: 10.1007/978-1-4684-2001-2_9. ISBN 978-1-4684-2003-6. Accessible from: `https://people.eecs.berkeley.edu/~luca/cs172/karp.pdf`

[12] RICHARDSON, Matthew and Pedro DOMINGOS, 2006. Markov logic networks. *Machine Learning*. **62**(1-2), 107-136. DOI: 10.1007/s10994-006-5833-1.

ISSN 0885-6125. Accessible from:
http://link.springer.com/10.1007/s10994-006-5833-1

[13] VAN DEN BROECK, Guy, 2011. On the completeness of first-order knowledge compilation for lifted probabilistic inference. *Advances in Neural Information Processing Systems*. **24**, 1386—1394. Accessible from:
https://dl.acm.org/doi/10.5555/2986459.2986614

[14] KUŽELKA, Ondřej and Vyacheslav KUNGURTSEV, 2019. Lifted Weight Learning of Markov Logic Networks Revisited, In: *Proceedings of Machine Learning Research*, 89, p. 1753–1761. Accessible from
http://proceedings.mlr.press/v89/kuzelka19a/kuzelka19a.pdf

[15] Gurobi Optimization, LLC, 2020. *gurobi*. Gurobi Optimizer Reference Manual. Accessible from:
http://www.gurobi.com

Appendix A
List of abbreviations

CNF	conjuctive normal form
FOL	first-order logic (also predicate logic)
ILP	integer linear programming
KB	knowledge base
MLN	Markov logic network
MRF	Markov random field (also Markov network)

Appendix B
Supplementary data and documentation

B.1 Source code

Source code of the thesis is publicly available at `https://github.com/kozakja4/m_thesis`

B.2 Content of CD

```
root
 |_ Code ..................................... source code folder
 |_ img ...................................... figures including their TikZ or Python definitions
 |_ text.pdf ................................ text of the thesis
```